

**UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA**

**FACULTAD DE INGENIERÍA**

**SECRETARÍA DE INVESTIGACIÓN Y POSGRADO**

---



**HACIA UN NUEVO INDICADOR DE CALIDAD PARA  
SOLUCIONES A PROBLEMAS MULTI-OBJETIVO POR  
MEDIO DE PROGRAMACIÓN GENÉTICA**

POR:

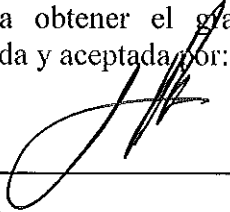
**CRISTIAN SANDOVAL REYES**

TESIS PRESENTADA PARA OBTENER EL GRADO DE

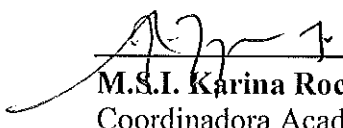
**MAESTRO EN INGENIERÍA EN COMPUTACIÓN**




Hacia un nuevo indicador de calidad para soluciones a problemas multi-objetivo por medio de programación genética. Tesis presentada por Cristian Sandoval Reyes como requisito parcial para obtener el grado de Maestro en Ingeniería en Computación, ha sido aprobada y aceptada por:

  
\_\_\_\_\_  
**M.I. Javier González Cantú**  
Director de la Facultad de Ingeniería

  
\_\_\_\_\_  
**Dr. Alejandro Villalobos Aragón**  
Secretario de Investigación y Posgrado

  
\_\_\_\_\_  
**M.S.I. Karina Rocío Requena Yáñez**  
Coordinadora Académico

  
\_\_\_\_\_  
**Dr. Luis Carlos González Gurrola**  
Director de Tesis

**Noviembre 2019**  
\_\_\_\_\_

Fecha

Comité:

**Dr. Luis Carlos González Gurrola**  
**Dr. Leonardo Trujillo Reyes**  
**Dra. Graciela Ramírez Alonso**  
**M.I. Jesús Roberto López Santillán**

© Derechos Reservados

**Cristian Sandoval Reyes**  
Circuito Universitario Campus II,  
Chihuahua, Chih., México.

Noviembre 2019



UNIVERSIDAD AUTÓNOMA DE  
CHIHUAHUA

11 de noviembre de 2018

**ING. CRISTIAN SANDOVAL REYES**

Presente

En atención a su solicitud relativa al trabajo de tesis para obtener el grado de Maestro en Ingeniería en Computación, nos es grato transcribirle el tema aprobado por esta Dirección, propuesto y dirigido por el director **Dr. Luis Carlos González Gurrola** para que lo desarrolle como tesis, con el título: **"HACIA UN NUEVO INDICADOR DE CALIDAD PARA SOLUCIONES A PROBLEMAS MULTI-OBJETIVO POR MEDIO DE PROGRAMACIÓN GENÉTICA"**.

## ÍNDICE

Índice de figuras

Índice de tablas

### 1. Introducción

- 1.1 Antecedentes
- 1.2 Problema de investigación
- 1.3 Preguntas de investigación
- 1.4 Justificación
- 1.5 Objetivos de la tesis

### 2. Optimización multi-objetivo

- 2.1 Cómputo evolutivo
- 2.2 Problemas de optimización multi-objetivo
- 2.3 Terminología de Pareto
- 2.4 Vectores de referencia
- 2.5 Algoritmos evolutivos multi-objetivo
- 2.6 Indicador hipervolumen



UNIVERSIDAD AUTÓNOMA DE  
CHIHUAHUA

### 3. Programación Genética

- 3.1 Codificación de los programas
- 3.2 Proceso evolutivo
- 3.3 Parámetros de control
- 3.4 Algoritmo GPTIPS

### 4. Funciones de prueba

- 4.1. Conjunto de prueba DTLZ

### 5. Experimentos y resultados obtenidos

- 5.1 Metodología
- 5.2 Resultados

### 6. Conclusiones y trabajo futuro

- 6.1 Conclusiones
- 6.2 Trabajo futuro

### Referencias

Solicitamos a Usted tomar nota de que el título del trabajo se imprima en lugar visible de los ejemplares de las tesis.

ATENTAMENTE  
"Naturam subiecit aliis"

EL DIRECTOR

M.I. JAVIER GONZÁLEZ CANTÚ

FACULTAD DE  
INGENIERÍA  
U.A.CH.



DIRECCIÓN

EL SECRETARIO DE INVESTIGACIÓN  
Y POSGRADO

DR. ALEJANDRO VILLALOBOS ARAGÓN

## **Abstract**

In the real world it is not uncommon to find an optimization problem with multiple objectives in conflict, and therefore there is not a single optimal solution but a set of optimal solutions. Multi-objective Genetic Algorithms have resulted in a promising tool for solving these types of problems. However, an important issue in multi-objective optimization is to evaluate and compare the quality of the solution sets generated by different multi-objective algorithms. A simple way to compare the quality of solution sets is through visualization, but visual comparison cannot quantify the difference between solution sets since this problem becomes harder as the number of objective functions increases. Quality indicators arise as a quantitative way of comparing solution sets. The most commonly used indicator, probably due to its mathematical properties, is the so-called hypervolume. Hypervolume is not only used to compare sets of solutions, but also as a selection criterion in the optimization process within these algorithms. It is important to note that coming with the exact of hypervolume is computationally expensive, meaning that the running time grows exponentially with respect to the number of objective functions. The latter being one of its main shortcomings, especially when the number of objective functions is high. In this thesis work a new approach is proposed to approximate the hypervolume value using models trained with Genetic Programming. The results are very promising, showing that some of the models can approximate the hypervolume value with an accuracy of up to 99% with a speed 10 to 80 times faster than other competing algorithms used for the same task.

## Resumen

En un mundo real no es raro encontrar un problema de optimización con múltiples objetivos donde estos objetivos a menudo están en conflicto entre si y no existe una solución óptima única sino un conjunto de soluciones óptimas. Los algoritmos genéticos multi-objetivo han resultado en una herramienta prometedora para la resolución de este tipo de problemas. Sin embargo, un tema importante en la optimización multi-objetivo es evaluar y comparar la calidad de los conjuntos de soluciones generados por los diferentes optimizadores multi-objetivo. Una forma sencilla de comparar la calidad de los conjuntos de soluciones es la visualización. No obstante, la comparación visual no puede cuantificar la diferencia entre los conjuntos de soluciones además de volverse más difícil conforme se incrementa el número de funciones objetivo. Los indicadores de calidad surgen como una forma cuantitativa de comparar los conjuntos de soluciones. El indicador más utilizado debido a sus propiedades matemáticas es el indicador hipervolumen. Este indicador no solo es utilizado para medir el desempeño de los algoritmos de búsqueda, sino también como criterio de selección de dichos algoritmos. Sin embargo, calcular el valor exacto de hipervolumen computacionalmente hablando es costoso. Los algoritmos conocidos para el cálculo del valor de hipervolumen crecen exponencialmente respecto al número de funciones objetivos circunstancia que limita su uso, especialmente cuando el número de funciones objetivo es elevado. En este trabajo de tesis se propone un nuevo enfoque para aproximar el valor de hipervolumen utilizando modelos entrenados con Programación Genética. Los resultados son muy prometedores, mostrando que algunos de los modelos son capaces de aproximar el valor de hipervolumen con una exactitud de 99 % con una velocidad 10 a 80 veces más rápida que otros algoritmos utilizados para el cálculo y aproximación de hipervolumen.

# Índice

<b>Índice de figuras</b>	<b>VIII</b>
<b>Índice de tablas</b>	<b>IX</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Antecedentes . . . . .	2
1.2. Problema de investigación . . . . .	4
1.3. Preguntas de investigación . . . . .	5
1.4. Justificación . . . . .	5
1.5. Objetivos de la tesis . . . . .	6
1.5.1. Objetivo general . . . . .	6
1.5.2. Objetivos específicos . . . . .	6
<b>2. Optimización multi-objetivo</b>	<b>7</b>
2.1. Cómputo evolutivo . . . . .	7
2.1.1. Principales paradigmas . . . . .	8
2.2. Problemas de optimización multi-objetivo . . . . .	12
2.3. Terminología de Pareto. . . . .	12
2.4. Vectores de referencia. . . . .	14
2.5. Algoritmos evolutivos multi-objetivo . . . . .	14
2.5.1. Métricas para evaluar la eficiencia de los algoritmos genéticos multi-objetivo . . . . .	16
2.5.2. Algoritmos evolutivos multi-objetivo basados en indicadores . . . . .	21
2.6. Indicador hipervolumen . . . . .	22
2.6.1. Contribución al hipervolumen . . . . .	23
2.6.2. Algoritmos para calcular el valor de hipervolumen . . . . .	24
2.6.3. Algoritmos para estimar el valor y contribución de hipervolumen . . . . .	28
2.6.4. Algoritmo SMS-EMOA basado en hipervolumen dominado . . . . .	30
<b>3. Programación Genética</b>	<b>32</b>
3.1. Codificación de los programas . . . . .	33
3.1.1. Elementos del árbol de sintaxis . . . . .	34
3.2. Proceso evolutivo . . . . .	36



3.2.1. Operadores primarios . . . . .	37
3.2.2. Selección . . . . .	37
3.2.3. Cruza . . . . .	37
3.2.4. Mutación . . . . .	40
3.3. Parámetros de control . . . . .	41
3.4. Algoritmo GPTIPS . . . . .	42
<b>4. Funciones de prueba</b>	<b>44</b>
4.1. Conjunto de prueba DTLZ . . . . .	44
4.1.1. Problema DTLZ1 . . . . .	45
4.1.2. Problema DTLZ2 . . . . .	46
4.1.3. Problema DTLZ3 . . . . .	47
4.1.4. Problema DTLZ4 . . . . .	49
4.1.5. Problema DTLZ5 . . . . .	50
4.1.6. Problema DTLZ6 . . . . .	51
4.1.7. Problema DTLZ7 . . . . .	53
<b>5. Experimentos y resultados obtenidos</b>	<b>56</b>
5.1. Metodología . . . . .	57
5.1.1. Bases de datos. . . . .	58
5.1.2. Entrenamiento y evaluación de desempeño . . . . .	60
5.1.3. Selección de modelos 3 funciones objetivo. . . . .	61
5.1.4. Selección de modelos 4 funciones objetivo. . . . .	64
5.2. Resultados . . . . .	68
5.2.1. Modelo como indicador para evaluar la calidad de conjuntos de soluciones no dominadas en problemas con 3 funciones objetivo . . . . .	68
5.2.2. Modelo como método de selección del evolutivo SMS-EMOA para problemas con 3 funciones objetivo . . . . .	72
5.2.3. Modelo como indicador para evaluar la calidad de conjuntos de soluciones no dominadas en problemas con 4 funciones objetivo . . . . .	86
5.2.4. Modelo como método de selección del evolutivo SMS-EMOA para problemas con 4 funciones objetivo . . . . .	88
<b>6. Conclusiones y trabajo futuro</b>	<b>100</b>
6.1. Conclusiones . . . . .	100
6.2. Trabajo futuro . . . . .	102
<b>Referencias</b>	<b>103</b>

# Índice de figuras

2.1. Frente de Pareto. . . . .	13
2.2. Aspectos a considerar para evaluar la calidad de una aproximación al frente de Pareto. . . . .	19
2.3. Conjuntos de soluciones. . . . .	20
2.4. El indicador hipervolumen en el caso de dos objetivos. . . . .	23
2.5. Contribucion al valor de hipervolumen en el caso de dos objetivos. . . . .	24
3.1. Representación de un árbol de sintaxis operadores aritméticos. . . . .	33
3.2. Representación de un árbol de sintaxis operadores lógicos. . . . .	34
3.3. Nodos de un árbol de sintaxis. . . . .	35
3.4. Ejemplo de población inicial. . . . .	36
3.5. Selecccion de los individuos más aptos. . . . .	38
3.6. Individuos seleccionados para cruza. . . . .	39
3.7. Hijos generados por el operador de cruza. . . . .	40
3.8. Ejemplo de mutacion. . . . .	41
4.1. Frente de Pareto problema DTLZ1. . . . .	46
4.2. Frente de Pareto problema DTLZ2. . . . .	47
4.3. Frente de Pareto problema DTLZ3. . . . .	48
4.4. Frente de Pareto problema DTLZ4. . . . .	50
4.5. Frente de Pareto problema DTLZ5. . . . .	51
4.6. Frente de Pareto problema DTLZ6. . . . .	53
4.7. Frente de Pareto problema DTLZ7. . . . .	54
5.1. Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ1 con 3 funciones objetivo. . . . .	74
5.2. Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ2 con 3 funciones objetivo. . . . .	75



5.3.	Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ3 con 3 funciones objetivo. . . . .	76
5.4.	Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ4 con 3 funciones objetivo. . . . .	77
5.5.	Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ5 con 3 funciones objetivo. . . . .	78
5.6.	Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ6 con 3 funciones objetivo. . . . .	79
5.7.	Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ7 con 3 funciones objetivo. . . . .	80
5.8.	Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ1 con 3 funciones objetivo. . . . .	82
5.9.	Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ2 con 3 funciones objetivo. . . . .	82
5.10.	Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ3 con 3 funciones objetivo. . . . .	83
5.11.	Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ4 con 3 funciones objetivo. . . . .	83
5.12.	Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ5 con 3 funciones objetivo. . . . .	84
5.13.	Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ6 con 3 funciones objetivo. . . . .	84



5.14. Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ7 con 3 funciones objetivo. . . . .	85
5.15. Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ1 con 4 funciones objetivo. . . . .	89
5.16. Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ2 con 4 funciones objetivo. . . . .	90
5.17. Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ3 con 4 funciones objetivo. . . . .	91
5.18. Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ4 con 4 funciones objetivo. . . . .	92
5.19. Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ4 con 3 funciones objetivo. . . . .	93
5.20. Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ6 con 4 funciones objetivo. . . . .	94
5.21. Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ7 con 4 funciones objetivo. . . . .	95
5.22. Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ1 con 4 funciones objetivo. . . . .	96
5.23. Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ2 con 4 funciones objetivo. . . . .	97
5.24. Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ3 con 4 funciones objetivo. . . . .	97



5.25. Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ4 con 4 funciones objetivo. . . . .	98
5.26. Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ5 con 4 funciones objetivo. . . . .	98
5.27. Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ6 con 4 funciones objetivo. . . . .	99
5.28. Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ7 con 4 funciones objetivo. . . . .	99

# Índice de tablas

4.1. Características problemas de prueba DTLZ. . . . .	55
5.1. Parámetros utilizados en el algoritmo SMSEMOA. . . . .	58
5.2. Tamaño de las bases de datos de problemas DTLZ. . . . .	59
5.3. Rasgos estadísticos que describen un conjunto de puntos no dominados de un problemas de 3 funciones objetivo. . . . .	59
5.4. Parámetros de entrenamiento GPTIPS. . . . .	60
5.5. Desempeño de los modelos entrenados con bases de datos normalizadas problemas con 3 funciones objetivo. . . . .	62
5.6. Desempeño de los modelos entrenados con bases de datos no normalizadas problemas con 3 funciones objetivo. . . . .	63
5.7. Rasgos estadísticos que describen un conjunto de puntos no dominados de un problemas de 4 funciones objetivo. . . . .	65
5.8. Desempeño de los modelos entrenados con bases de datos normalizadas problemas con 4 funciones objetivo. . . . .	66
5.9. Desempeño de los modelos entrenados con bases de datos no normalizadas problemas con 4 funciones objetivo. . . . .	67
5.10. Resultados de tiempo para los problemas DTLZ con 3 funciones objetivo. . . . .	69
5.11. Resultado de errores del <i>Modelo</i> <sub>11</sub> para los problemas DTLZ con 3 funciones objetivo. . . . .	71
5.12. Resultado de errores del <i>Modelo</i> <sub>12</sub> para los problemas DTLZ con 3 funciones objetivo. . . . .	72
5.13. Resultados de tiempo para los problemas DTLZ con 4 funciones objetivo. . . . .	86
5.14. Resultado de errores del <i>Modelo</i> <sub>15</sub> para los problemas DTLZ con 4 funciones objetivo. . . . .	87
5.15. Resultado de errores del <i>Modelo</i> <sub>2</sub> para los problemas DTLZ con 4 funciones objetivo. . . . .	87

# Capítulo 1

## Introducción

En las últimas décadas los algoritmos evolutivos multi-objetivo (por sus siglas en inglés, *Multi-Objective Evolutionary Algorithms*: MOEAs) se han convertido en una herramienta muy popular para resolver problemas de optimización numérica y combinatoria en distintas áreas de la ingeniería y ciencias computacionales. Su efectividad en la solución de problemas complejos de optimización, así como su flexibilidad y fácil uso los ha convertido en uno de los métodos más utilizados en la literatura especializada.

Generalmente, los problemas de optimización que se presentan en el mundo real no son lineales, sino que suelen requerir de la optimización de dos o más objetivos que se contraponen entre sí, en donde no existe una única solución sino un conjunto de soluciones denominadas frente de Pareto [1], a este tipo de problemas se les denomina problemas de optimización multi-objetivo [2].

Una de las principales causas que ha motivado el uso de estos evolutivos para resolver problemas de optimización es que a diferencia de otros algoritmos trabajan con un grupo de soluciones a la vez, lo cual permite aproximar el conjunto de óptimo de Pareto en una sola ejecución [3]. Esta característica disminuye la limitante que tienen algunos otros evolutivos y algunas técnicas de programación matemática-



ca las cuales necesitan ser ejecutadas múltiples veces para obtener varios elementos del óptimo de Pareto.

Cuando se trabaja con problemas de optimización multi-objetivo es necesario comprobar el desempeño que tienen los algoritmos en la búsqueda del conjunto de óptimos de Pareto. Para realizar una comparación objetiva ha sido necesario desarrollar indicadores de desempeño que puedan cuantificar las diferencias de calidad entre las aproximaciones obtenidas por los MOEAs. Uno de los indicadores más populares y utilizados debido a sus propiedades matemáticas es el indicador hipervolumen, se ha demostrado que maximizar este indicador conduce a lograr la convergencia hacia el óptimo de Pareto [4]. Sin embargo, calcular el valor exacto de hipervolumen es costoso computacionalmente hablando. Por esta razón, en este trabajo de investigación se propone un enfoque de *Machine Learning* donde se utiliza Programación Genética para entrenar modelos que puedan aproximar el valor de hipervolumen de una aproximación realizada por un algoritmo de búsqueda al óptimo de Pareto. Los modelos más eficientes se utilizarán para aproximar la contribución de hipervolumen en el criterio de selección en un algoritmo genético multi-objetivo basado en indicadores.

## 1.1. Antecedentes

El indicador hipervolumen es el indicador más utilizado para medir la calidad de las soluciones finales producidas por un algoritmo de búsqueda. Los importantes beneficios de utilizar esta métrica han llevado a los investigadores a buscar técnicas más eficientes para calcularlo.

Existen diversas propuestas para realizar el cálculo de hipervolumen, por ejemplo el algoritmo de inclusión-exclusión [5], este método suma todos los volúmenes obtenidos al intersecar un conjunto de volúmenes n-dimensionales, cuya compleji-



dad es  $\mathcal{O}(n2^m)$ . Beaume [6] propone considerar la métrica de hipervolumen como un caso especial de un problema más general llamado métrica de *klee* [7] cuya complejidad para  $n$  puntos con  $d \geq 3$  objetivos es  $\mathcal{O}(n \log(n) + n^{\frac{m}{2}} \log(n))$ . La medida de *Lebesgue* fue propuesta por Fleischer [8] este indicador se enfoca en remover un polítopo rectangular disjunto del espacio formado por un conjunto de puntos no dominados, el valor de hipervolumen se incrementa en base al polotipo eliminado, la complejidad de este algoritmo se demostró empíricamente por While [9] donde para el peor de los casos se tiene un tiempo computacional de  $\mathcal{O}(2^{m-1})$ . El algoritmo *Hypervolume by Slicing Objectives* (HSO) [10–12] procesa objetivos en lugar de puntos este método trabaja con un conjunto de puntos no dominados, previamente ordenados con respecto al primer objetivo, posteriormente traza secciones transversales a lo largo de este objetivo. Al igual que el algoritmo de *Lebesgue* su complejidad fue demostrada por While [12] la cual es  $\mathcal{O}(m^{n-1})$  y aunque logra una mejora respecto a *Lebesgue* no logra resolver el problema de alta dimensionalidad. Fonseca, Paquete y López Ibáñez propusieron el algoritmo FLP [13] basado en el algoritmo HSO, con las mejoras realizadas se logró reducir el tiempo de ejecución a  $\mathcal{O}(n^{m-2} \log(n))$ . El algoritmo WFG [14] es una variación del algoritmo FLP que fue propuesta por el *Walking Fish Group*. Estos últimos algoritmos están considerados dentro de los mejores para realizar el cálculo de hipervolumen. Sin embargo, aún sigue siendo complicado utilizarlos en problemas de alta dimensionalidad.

Por otro lado, están los algoritmos diseñados para aproximar el valor de hipervolumen para un conjunto de soluciones no dominadas. El más representativo es el algoritmo denominado HypE [15] que está basado en muestreo de Monte Carlo, este método ha demostrado ser un buen estimador capaz de aumentar la precisión conforme se aumenta el número de puntos de muestra. No obstante, también es muy sensible al incremento del número de dimensiones.

El indicador hipervolumen también es utilizado dentro del esquema de selec-



ción de algoritmos basados en indicadores, esto debido principalmente a que es compatible con la dominancia de Pareto. Algunos de los MOEAs basados en indicadores que se encuentran entre los más representativos del área son el *Indicador Based Evolutionary Algorithm* (IBEA) [16] y *S-Metric Selection Evolutionary Multi-Objective Algorithm* (SMS-EMOA) [17]. El algoritmo IBEA permite el uso de distintos indicadores de desempeño como criterio de selección y generalmente la aptitud de una solución está basada en la suma de los valores de los indicadores resultantes por pares de las comparaciones a todas las soluciones. El algoritmo SMS-EMOA es un algoritmo muy potente para problemas difíciles, que utiliza el valor de hipervolumen para mejorar la distribución de las soluciones en el frente de Pareto. El principal inconveniente de estos algoritmos es su alta complejidad computacional relacionada al cálculo de contribución de hipervolumen.

## 1.2. Problema de investigación

Aunque el uso del indicador hipervolumen engloba de una manera positiva varias ventajas importantes [4] que permiten evaluar el desempeño de un MOEA, este indicador presenta diversas limitantes las cuales radican en el costo computacional que representa realizar el cálculo del valor exacto de hipervolumen. No existe ningún algoritmo que pueda calcular en tiempo polinomial el valor de hipervolumen, desventaja que limita su uso en la práctica. Por tal razón, persiste la motivación de buscar otras alternativas para realizar el cálculo del valor de hipervolumen en tiempo polinomial. En este trabajo se plantea incorporar modelos entrenados con GP a un algoritmo basado en indicadores. Estos modelos permitirán aproximar el valor de hipervolumen sin incrementar su costo computacional al abordar problemas de alta dimensionalidad.



### 1.3. Preguntas de investigación

- ¿Puede un modelo entrenado con Programación Genética aproximar el valor del indicador hipervolumen?
- ¿Puede un modelo entrenado con Programación Genética guiar la búsqueda de un algoritmo genético multi-objetivo basado en indicadores?
- ¿Un indicador sintetizado mediante Programación Genética será de menor costo computacional que las propuestas ya existentes?

### 1.4. Justificación

Generalmente los problemas de optimización del mundo real son problemas que tienen un gran número de funciones objetivo los cuales se encuentran en conflicto entre sí, estos problemas son muy costosos de evaluar. Sin embargo, en la actualidad disponemos de herramientas de optimización multi-objetivo que nos permiten encontrar vectores de solución con valores aceptables para todas las funciones objetivo. No obstante, entre mayor es el número de funciones objetivo es más difícil discernir la calidad de las soluciones que estos algoritmos arrojan. Por tal motivo es necesario contar con métricas de calidad que midan la convergencia, la uniformidad y la dispersión de las soluciones obtenidas por un algoritmo de búsqueda. El indicador hipervolumen es la métrica más ampliamente utilizada gracias a sus propiedades matemáticas pero, el costo de realizar el cálculo de hipervolumen se incrementa exponencialmente conforme se aumenta el número de funciones objetivo [18, 19] lo que limita su uso en problemas de muchos objetivos. Estas limitantes del indicador hipervolumen han motivado a este trabajo de investigación para buscar formas más eficientes de realizar el cálculo de hipervolumen. Se busca entrenar modelos utilizando Programación Genética que sean capaces de obtener una bue-



na aproximación al valor real de hipervolumen con un costo computacional bajo. Los modelos serán utilizados para el cálculo de contribución de hipervolumen en el criterio de selección del algoritmo basado en indicadores SMS-EMOA [17]. La idea principal radica en minimizar el tiempo de ejecución al reducir el costo computacional que requiere realizar el cálculo del valor de hipervolumen.

## 1.5. Objetivos de la tesis

### 1.5.1. Objetivo general

Desarrollar un indicador de calidad que guíe la búsqueda de un algoritmo genético multi-objetivo, utilizando Programación Genética.

### 1.5.2. Objetivos específicos

- Incorporar el modelo al algoritmo genético multi-objetivo basado en indicadores *S-Metric Selection Evolutionary Multi-Objective Algorithm* (SMS-EMOA).
- Evaluar el desempeño del algoritmo SMS-EMOA basado en modelos entrenados con Programación Genética.
- Comparar el desempeño de las soluciones obtenidas del modelo como indicador con respecto al valor real de hipervolumen calculado con el indicador WFG de *Walking Fish Group*.

## Capítulo 2

# Optimización multi-objetivo

Muchos problemas de la vida cotidiana requieren de la optimización de dos o más objetivos que se contraponen entre sí. Por lo general este tipo de problemas no tienen una solución única sino un conjunto de soluciones que se distribuyen en un frente de Pareto [20]. Hoy en día existen diversas técnicas para resolver problemas de optimización multi-objetivo, los algoritmos evolutivos resultan particularmente adecuados para resolver problemas de optimización. En este capítulo se discuten los conceptos básicos de los algoritmos evolutivos y la optimización multi-objetivo.

### 2.1. Cómputo evolutivo

La computación evolutiva es una rama de la inteligencia artificial que constituye un conjunto de técnicas estocásticas que simulan la evolución natural. A estas técnicas se les conoce generalmente como algoritmos evolutivos. Las técnicas de computación evolutiva trabajan sobre una población de posibles soluciones la cual es evolucionada utilizando operadores inspirados en la variación genética y en la selección natural con la idea de producir iterativamente mejores soluciones. Dichas técnicas son procedimientos inspirados en el principio natural de evolución de las



especies conocido como *neodarwinismo*.

En el cómputo evolutivo existen individuos los cuales representan una solución potencial al problema que se está tratando, a un conjunto de individuos se le denomina población. Los individuos tienen la probabilidad de reproducirse en base a su *aptitud*, la cual indica que tan bueno es el individuo con respecto a los demás; con dicha aptitud se evalúa la supervivencia de un individuo en cada generación bajo el principio de supervivencia del más apto.

Para el paso de una generación a otra se aplica una serie de operadores genéticos los cuales se en listan a continuación.

- **Selección:** El operador de selección determina que individuos generaran descendencia, dicha descendencia formara parte de la siguiente generación. Los individuos son elegidos en base a su valor de aptitud, este procedimiento está basado en el principio de supervivencia del más apto.
- **Cruza:** El operador de cruza o recombinación tiene como finalidad heredar la información genética de dos o más padres a los hijos que formaran parte de la siguiente generación.
- **Mutación:** La mutación provoca una variación del cromosoma de uno o más individuos. La mutación opera solo sobre un individuo a la vez.

### 2.1.1. Principales paradigmas

La computación evolutiva está dividida en tres principales paradigmas. Estos paradigmas se dividen en Algoritmos Genéticos, Programación Evolutiva y Estrategias Evolutivas.

Las Estrategias Evolutivas (por sus siglas en inglés, *Evolutionary Estrategics*: EE) fueron propuestas por I. Rechenberg, P. Bienert y H.P. Schwefel [21] en 1964. La



primera versión de una estrategia evolutiva fue nombrada (1+1)-EE, la cual estaba inspirada en el mecanismo de mutación que ocurre en la naturaleza. La representación de cada individuo fue un vector de valores reales, el cual era manipulado por operadores de mutación. La (1+1)-EE crea un solo hijo usando mutación de un solo padre, ambos individuos compiten por sobrevivir; el mejor individuo pasaba a la siguiente generación mientras que el peor era eliminado. Rechenberg [22] introdujo el concepto de población al proponer una EE llamada  $(\mu + 1) - EE$ , en la cual hay  $\mu$  padres y se genera un solo hijo, el cual puede reemplazar al peor padre de la población. Las EE simulan el proceso evolutivo al nivel de los individuos, por lo que la recombinación es posible. Normalmente la selección de los individuos es determinista.

---

### Algoritmo 2.1 Estructura básica de una Estrategia Evolutiva

---

1. Establecer criterio de paro
2. Generar población inicial
3. **while** *criterio de paro no se satisfaga* **do**
4.   Para cada individuo de la población hacer
5.     Generar descendencia
6.     Seleccionar mejor individuo entre padre e hijo
7.     Agregar mejor individuo a la nueva generación
8. **end while**

---

Los Algoritmos Genéticos (por sus siglas en inglés, *Genetic Algorithms: GAs*) denominados originalmente como *planes reproductivos*, fueron propuestos por Holland [23, 24] en 1962. El desarrollo de los GAs estuvo influenciado por los procesos de adaptación natural en términos en que los individuos de una población interactua-



ran y mejoran en base a cierto ambiente que determina lo apropiado de su comportamiento. Los GAs toman para su definición una terminología utilizada en Genética Natural, conceptos como *genotipo*, *fenotipo*, *cromosoma*, *gene* y *alelo*. El *genotipo* describe la información genética que ha sido heredada por un individuo y puede transmitir a sus descendientes. El *fenotipo* representa las características visibles que manifiesta un individuo por la expresión de su *genotipo*. En el algoritmo el *fenotipo* es la representación codificada de una posible solución y el *genotipo* la cadena binaria que codifica dicha solución. A la cadena binaria se le llama *cromosoma*. A cada posición del *cromosoma* se le denomina *gene* y al valor dentro de esta posición se le denomina *alelo*. Los GAs utilizan los tres operadores genéticos: selección, cruce y mutación. Sin embargo, la cruce es el operador principal de este paradigma, en el GA se enfatiza la importancia de la cruce sexual sobre la mutación. La selección de los individuos es probabilística con base a aptitud.

---

#### Algoritmo 2.2 Estructura básica de un Algoritmo Genético

---

1. Establecer criterio de paro
  2. Generar población inicial
  3. **while** *criterio de paro no se satisfaga* **do**
  4.    Seleccionar padres
  5.    Recombinar los individuos utilizando cruce para generar descendencia
  6.    Mutar los individuos
  7.    Evaluar la aptitud de los nuevos individuos
  8.    Determinar los individuos que pasan a la nueva generación
  9. **end while**
- 

La Programación Evolutiva (por sus siglas en inglés, *Evolutionary Programming*:



EP) fue propuesta por Lawrence J. Fogel [25, 26], en 1964. Este paradigma es muy similar a las Estrategias Evolutivas con la diferencia de que no emplea el operador de cruza u algún otro tipo de recombinación. EP es una abstracción de la evolución al nivel de las especies puesto que en la naturaleza una especie no se mezcla con otra y este algoritmo evolutivo simula este principio. La EP enfatiza los nexos de comportamiento entre padres e hijos en vez de buscar operadores genéticos específicos, como lo hacen los GA. El único indicador de variación que incorpora es la mutación. La selección es realizada mediante torneo estocástico, el cual consiste en comparar un número de individuos entre sí y seleccionar el de mayor aptitud. Fogel consideraba que uno de los principales rasgos que caracterizaban el comportamiento inteligente era la capacidad de predecir el ambiente y la capacidad de traducir estas predicciones de manera adecuada, para modelar este comportamiento Fogel utilizó autómatas de estado finito.

---

### Algoritmo 2.3 Estructura básica de un algoritmo de Programación Evolutiva

---

1. Establecer criterio de paro
  2. Generar población inicial
  3. Evaluar población inicial
  4. **while** *criterio de paro no se satisfaga* **do**
  5.   Para cada individuo de la población hacer
  6.     Generar descendencia utilizando mutación
  7.     Evaluar descendencia
  8.     Realizar selección de individuos por torneo
  9.     Reemplazar la población actual con los individuos seleccionados
  10. **end while**
-

## 2.2. Problemas de optimización multi-objetivo

Se define el problema de optimización multi-objetivo como el proceso de encontrar un vector de variables de decisión que satisfaga todas las restricciones en caso de tenerlas y optimice un vector de funciones objetivo [27].

La mayor parte de los problemas de optimización del mundo real involucran más de un objetivo que necesita ser minimizado o maximizado. Esto es, suelen tener dos o más funciones objetivo que deben satisfacerse simultáneamente y que presentan un grado de conflicto entre sí. Un conflicto que ocasiona que la mejora de un objetivo provoque el deterioro de otros. Este tipo de problemas son considerados como problemas de optimización multi-objetivo [2]. Debido a la naturaleza conflictiva de los objetivos no existe una única solución capaz de optimizar todos los objetivos, sino que existen una serie de soluciones con valores aceptables para todas las funciones objetivo [2]. Un problema de optimización multi-objetivo consiste en buscar el mejor compromiso posible entre todos los objetivos.

*Definición 2.1* Un problema de optimización multi-objetivo se puede definir de la siguiente manera:  $F(\vec{x}) = (f_1(\vec{x}), \dots, f_k(\vec{x}))$ , donde  $k$  es el número de funciones a optimizar, sujetas a  $g_i(\vec{x}) \leq 0$  restricciones de desigualdad y  $h_j(\vec{x}) = 0$  restricciones de igualdad con  $i = 1, 2, \dots, m$ ,  $j = 1, 2, \dots, n$ , donde  $\vec{x} = [x_1, x_2, \dots, x_n]^T$  es el vector de variables de decisión que definen el problema. Las restricciones  $g_i(\vec{x}) \leq 0$  y  $h_j(\vec{x}) = 0$  deben de ser de ser satisfechas al mismo tiempo que se generan valores óptimos para todas las funciones objetivo  $F(\vec{x})$ .

## 2.3. Terminología de Pareto.

La noción de óptimo que es utilizada en optimización multi-objetivo fue propuesta originalmente por Francis Ysidro Edgeworth [28] y más tarde en 1896 fue

generalizada por Vilfredo Pareto [20], la cual se conoce como óptimo de Pareto.

**Definición 2.2 Optimalidad de Pareto:** Se dice que una solución  $\vec{x}^* \in \Omega$  es un óptimo de Pareto si no existe  $\vec{x} \in \Omega$  tal que  $f_i(\vec{x}) \leq f_i(\vec{x}^*)$  para toda  $i = 1, \dots, k$  y  $f_j(\vec{x}) < f_j(\vec{x}^*)$  para al menos una  $j$ .

**Definición 2.3 Dominancia de Pareto:** Se dice que un vector  $\vec{u} = [u_1, \dots, u_k]^T$  domina a otro vector  $\vec{v} = [v_1, \dots, v_k]^T$  si y solo si  $\vec{u}$  es parcialmente menor que  $\vec{v}$  es decir:  $\forall i \in \{1, \dots, k\} : u_i \leq v_i \wedge \exists i \in \{1, \dots, k\} : u_i < v_i$ .

**Definición 2.4 Conjunto de óptimos de Pareto:** Para un problema multi-objetivo dado  $f_i(\vec{x})$ , el conjunto de óptimo de Pareto, denotado por  $P^*$  se define como:  $P := \{\vec{x} \in \Omega \mid \nexists \vec{x}' \in \Omega, f_i(\vec{x}') \leq f_i(\vec{x})\}$ .

**Definición 2.5 Frente de Pareto:** Para un problema multi-objetivo dado  $f(\vec{x})$  y su conjunto de óptimos de Pareto ( $P^*$ ), el frente de Pareto  $PF^*$  se define como  $PF^* := \{f(\vec{x}), \vec{x} \in P^*\}$ .

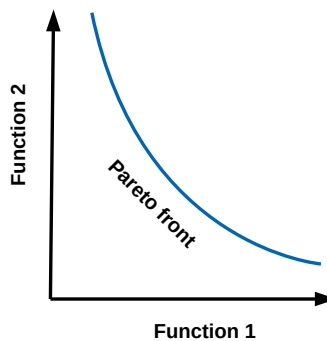


Figura 2.1: Frente de Pareto.

## 2.4. Vectores de referencia.

Los algoritmos de optimización multi-objetivo utilizan algunos vectores especiales como puntos de referencia, dichos vectores se definen de la siguiente manera.

**Definición 2.6 Vector objetivo ideal:** El vector ideal minimiza cada función objetivo de manera independiente y se define como:  $f_{ideal} = (\min_{\vec{x} \in P^*} f_1(\vec{x}), \dots, \min_{\vec{x} \in P^*} f_k(\vec{x}))$ .

**Definición 2.7 Vector objetivo de nadir:** El vector objetivo de nadir contiene en cada uno de sus componentes al máximo de cada objetivo y se define como:  $f_{nadir} = (\max_{\vec{x} \in P^*} f_1(\vec{x}), \dots, \max_{\vec{x} \in P^*} f_k(\vec{x}))$ .

El vector ideal y el vector de nadir suelen utilizarse para acotar el frente de Pareto, ya que normalizan los valores objetivo en una escala común.

## 2.5. Algoritmos evolutivos multi-objetivo

La capacidad de los algoritmos evolutivos de resolver problemas con múltiples objetivos fue sugerida por Rosenberg [2] desde 1967. Sin embargo, no llegó a desarrollar un algoritmo. La primera implementación de lo que hoy se conoce como un algoritmo genético multi-objetivo fue propuesta por Schaffer [29] en 1984. Desde entonces los algoritmos genéticos multi-objetivo se han popularizado como métodos robustos y efectivos para resolver problemas de optimización basados en dominancia de Pareto [30]. Los MOEAs presentan varias ventajas al momento de resolver problemas de optimización multi-objetivo, una de ellas es que trabajan simultáneamente con un conjunto de soluciones potenciales o poblaciones lo que permite encontrar varias soluciones del óptimo de Pareto en una sola ejecución [29] además de que tienden a ser menos susceptibles a la discontinuidad y la forma del frente de Pareto.



Los MOEAs surgen como una extensión de los algoritmos evolutivos, al igual que los AGs su funcionamiento se basa en la simulación del proceso de evolución natural de los seres vivos. Sin embargo, los MOEAs utilizan un esquema de selección que debe considerar  $k \geq 2$  funciones objetivo. Los principales métodos utilizados para manejar varias funciones objetivo en el proceso de selección se clasifican en: métodos de agregación, métodos basados en población y métodos basados en jerarquización de Pareto.

Los métodos de agregación [31] utilizan una función escalar para calcular la aptitud de los individuos, este método solo produce una solución al transformar un problema de optimización multi-objetivo en un problema mono-objetivo.

El método basado en población [29] cambia el criterio de selección durante la reproducción, lo que permite guiar la búsqueda en varias direcciones, a diferencia del método anterior este método es capaz de evolucionar múltiples soluciones no dominadas en una sola ejecución.

El método basado en jerarquización de Pareto [30] es el más utilizado en la actualidad, este método asigna la aptitud de los individuos en base a la dominancia de Pareto, su objetivo es seleccionar soluciones no dominadas y considerar todas las soluciones no dominadas en una población como igualmente buenas, esto significa que a todos los individuos no dominados se les asigna un mismo rango de aptitud.

Otras características importantes de los algoritmos genéticos multi-objetivo son la diversidad y el elitismo, mantener la diversidad en la población permite encontrar un conjunto de soluciones bien distribuidas sobre el frente de Pareto, para mantener la diversidad de las soluciones existen varios enfoques tales como *niching* [32], *clustering* [33–35], *fitness sharing* [36], *entropy* [37, 38] y *crowding* [39] entre otros.

El elitismo [39] normalmente consiste en mantener a los individuos más ap-



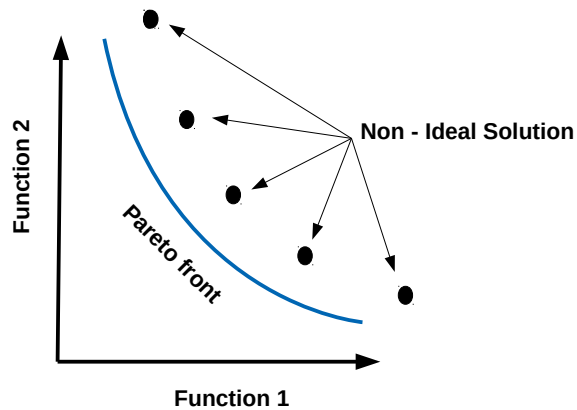
tos de una generación a otra y de esa manera garantizar que un algoritmo tenga un desempeño monótono creciente; una forma de implementación de elitismo es que en lugar de remplazar la generación de padres con la nueva generación ambas son combinadas para posteriormente aplicar una selección determinista. Otra forma de elitismo es utilizar archivos históricos también conocidos como población secundaria la cual interactúa de diferentes maneras con la población principal, estos archivos retienen las soluciones no dominadas generadas a lo largo del proceso de búsqueda. La aproximación al óptimo de Pareto realizada por un algoritmo de búsqueda generalmente es el contenido final de este archivo.

### **2.5.1. Métricas para evaluar la eficiencia de los algoritmos genéticos multi-objetivo**

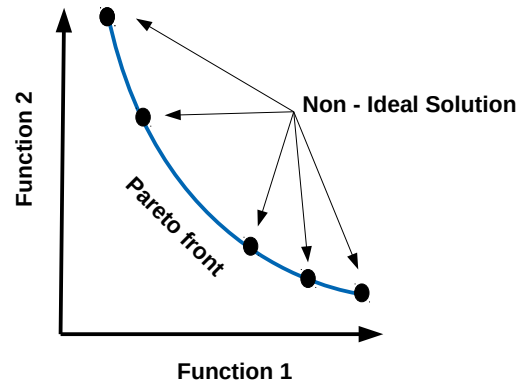
Las métricas de desempeño o indicadores de calidad permiten comparar la calidad de las soluciones obtenidas por un algoritmo de búsqueda mapeando la aproximación que este realiza al frente de Pareto [27, 40]. Para evaluar la calidad de estas aproximaciones se deben tomar en cuenta tres aspectos importantes: la convergencia al frente de Pareto, la diversidad de las soluciones en el espacio objetivo y la dispersión de las soluciones en el frente de Pareto [41, 42].

- La convergencia al frente de Pareto indica cuan distante es la aproximación realizada por un algoritmo de búsqueda al frente óptimo de Pareto.
- La diversidad de las soluciones en el espacio objetivo indica la distancia relativa entre las soluciones encontradas.
- Mientras que la dispersión se refiere al número de soluciones del conjunto de óptimos de Pareto encontrados, las cuales deben cubrir la mayor región posible.

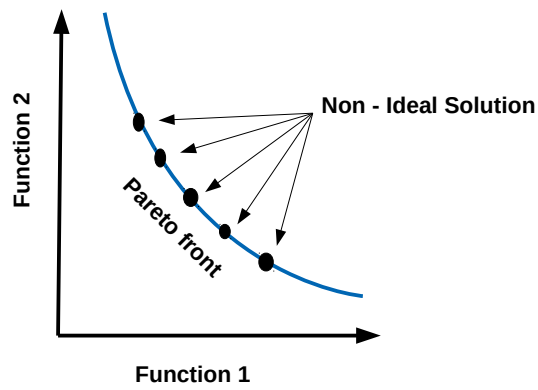
Los aspectos considerados para evaluar la calidad de una aproximación al frente de Pareto se ejemplifican en la figura 2.1. En (a) se muestra un conjunto de soluciones que no satisfacen el criterio de convergencia. En (b) se ejemplifica un conjunto de soluciones que poseen poca diversidad. En (c) se puede observar un conjunto de soluciones que no poseen una buena dispersión de las soluciones. El conjunto de características deseadas de una aproximación al frente de Pareto se muestran en (d), en donde se puede observar un conjunto de soluciones que cumplen con los criterios de convergencia, diversidad y dispersión.



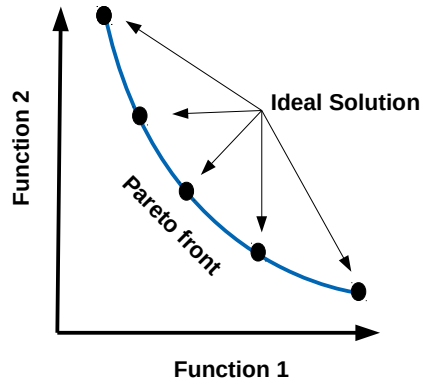
(a)



(b)



(c)



(d)

Figura 2.2: Aspectos a considerar para evaluar la calidad de una aproximación al frente de Pareto.

Las métricas de calidad se clasifican en unarias y binarias [43]. Las métricas unarias devuelven un único valor real que representa la calidad de un conjunto de soluciones según un criterio específico que engloba uno o más de los tres aspectos mencionados anteriormente, mientras que las métricas binarias o  $M$ -arias comparan la calidad relativa de  $n$  conjuntos de soluciones no dominadas.

### Indicadores como medida de calidad

Un indicador de calidad es una medida cuantitativa de la calidad de las soluciones producidas por un algoritmo de búsqueda, esta medida permite realizar una comparación objetiva de las soluciones finales producidas por varios MOEAs. La idea de utilizar un indicador como medida es cuantificar las diferencias de calidad

entre las aproximaciones obtenidas por distintos MOEAs.

**Definición 2.8 Indicador de calidad:** Un indicador de calidad es una función  $I : \Omega \rightarrow \mathbb{R}$  que asigna un valor real a  $k$  conjuntos de aproximaciones al frente de Pareto  $A_1, \dots, A_K \in \Omega$ .

El orden que establece  $I$  en  $\Omega$  equivale a la calidad de la aproximación al frente de Pareto. Si tenemos dos conjuntos de aproximación  $A_1$  y  $A_2$  tales que  $I(A_1) > I(A_2)$  entonces el conjunto  $A_1$  es preferible al conjunto  $A_2$ .

En la figura 2.3 se muestra un ejemplo para el caso que un conjunto de soluciones sea mejor que otro. En la imagen el conjunto que consta de los puntos del círculo rojo es mejor que el conjunto que consta de los cuadros azules.

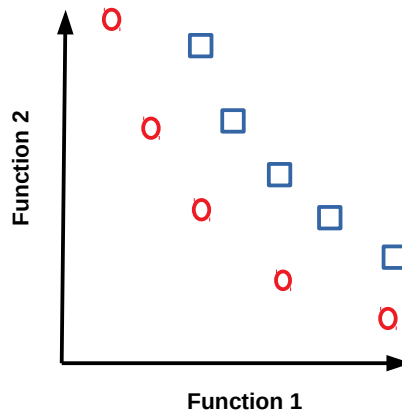


Figura 2.3: Conjuntos de soluciones.



### Indicadores como método de selección

En la optimización multi-objetivo los indicadores para la evaluación del desempeño pueden ser adaptados para que realicen el proceso de selección en presencia de muchas funciones objetivo. Generalmente dichas métricas proporcionan un mejor método de selección el cual permite mejorar el rendimiento de los evolutivos multi-objetivo. Esto bajo el supuesto de que, si un conjunto de soluciones es mejor que otro bajo una métrica de calidad, ese conjunto debería ser más deseable durante el transcurso de la ejecución del algoritmo.

El indicador hipervolumen es utilizado en varios algoritmos basados en indicadores como criterio de selección [16, 17] ya que la clasificación de las soluciones que produce es invariante al escalamiento lineal de las funciones objetivo.

#### 2.5.2. Algoritmos evolutivos multi-objetivo basados en indicadores

Un indicador de calidad puede integrarse como método de selección de un algoritmo genético multi-objetivo. Esta técnica permite transformar un problema de varios objetivos en un problema de un solo objetivo, donde ahora el objetivo consiste en maximizar el valor del indicador.

Utilizar indicadores de calidad para guiar la búsqueda de un MOEA ha permitido a los algoritmos adaptarse de forma positiva a espacios objetivos de alta dimensión evitando que la calidad de sus soluciones se degrade conforme aumenta el número de dimensiones. Los algoritmos IBEA (*Indicator-Based Evolutionary Algorithm*) [16] y SMS-EMOA (*S-Metric Selection Evolutionary Multi-Objective Algorithm*) [17] son algoritmos basados en indicadores que se adaptan muy bien a espacios objetivo de alta dimensión. Aunque existen distintas variantes, el uso del indicador hipervolumen como guía es el más utilizado.

## 2.6. Indicador hipervolumen

La métrica hipervolumen fue propuesta por Zitzler y Thiele [44] en 1998, quienes lo definen como el tamaño del espacio cubierto o dominado. Este indicador de calidad es uno de los más recurridos para evaluar la calidad de las soluciones finales producidas por un algoritmo genético multi-objetivo, también es el más utilizado como criterio de selección de los optimizadores multi-objetivo. Este indicador es el único de los reportados en la literatura que cumple con la propiedad de estricta monotonía con respecto a la relación de la dominancia de Pareto.

**Definición 2.9 Indicador hipervolumen:** Dado  $\Lambda$  que denota la métrica de Lebesgue o métrica  $\mathcal{S}$ , el hipervolumen es definido como  $I_h(\mathcal{P}) = \mathcal{S}(\mathcal{P}, y_{ref}) = \Lambda \left( \bigcup_{y \in \mathcal{P}} \{y' | y \prec y' \prec y_{ref}\} \right)$ .

El hipervolumen encapsula en un solo valor unario una medida de la extensión de las soluciones a lo largo del frente de Pareto, así como la distancia del conjunto desde el frente óptimo de Pareto. Las propiedades matemáticas favorables del hipervolumen permiten detectar cuando un conjunto de soluciones no es peor que otro conjunto para todos los pares de soluciones [4], además de que si se maximiza el valor de hipervolumen equivale a lograr la convergencia hacia el verdadero óptimo de Pareto [4].

El valor de hipervolumen es calculado utilizando un punto de referencia, regularmente el peor punto en espacio de soluciones. Sin embargo, la elección del punto de referencia puede afectar el resultado e inferir en las conclusiones resultantes de las comparaciones de hipervolumen. Por lo que el punto de referencia debe de ser elegido bajo ciertas reglas. Primero el punto de referencia debe ser menor o igual que el vector objetivo Ideal para la minimización de hipervolumen. Segundo el punto de referencia debe ser mayor o igual al vector de Nair para la maximización del hipervolumen. La figura 2.4 muestra el indicador hipervolumen para el caso de dos funciones objetivo.

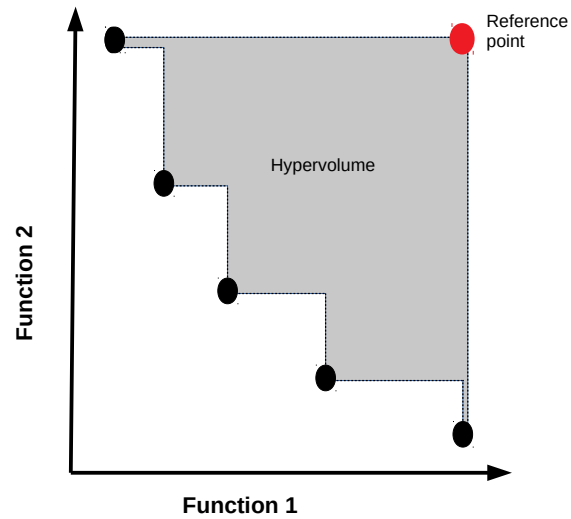


Figura 2.4: El indicador hipervolumen en el caso de dos objetivos.

### 2.6.1. Contribución al hipervolumen

La contribución de hipervolumen de un individuo  $x \in M$  se define como:  $Con_M(x) := Hyp(M) - Hyp(M \setminus \{x\})$ . La cual refleja la influencia de un solo punto del conjunto de soluciones que aproximan un frente de Pareto. La figura 2.5 ejemplifica la contribución al hipervolumen de cada solución de un conjunto de soluciones.

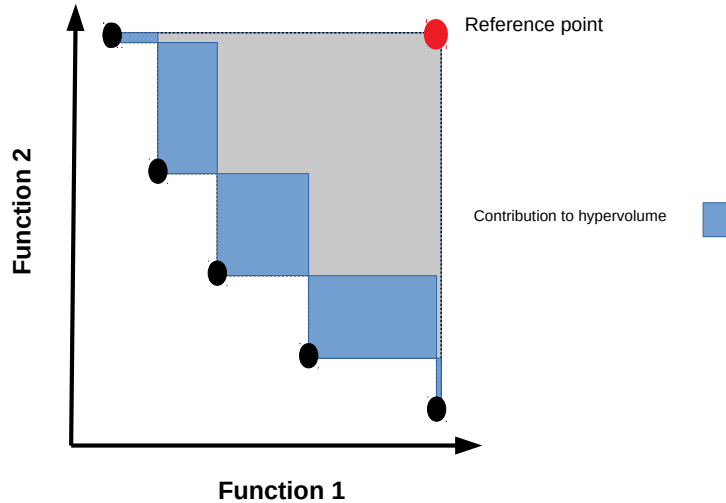


Figura 2.5: Contribucion al valor de hipervolumen en el caso de dos objetivos.

El algoritmo SMS-EMOA [17] en su proceso de selección utiliza las características positivas de la medida de hipervolumen para realizar la selección de los candidatos a la solución de acuerdo con su contribución al hipervolumen dominado; esto con la finalidad de que la población del algoritmo evolucione a un conjunto de soluciones bien distribuido al centrarse en las regiones interesantes del frente de Pareto. El algoritmo SMS-EMOA se describe con mayor detalle en el apartado 2.6.4.

### 2.6.2. Algoritmos para calcular el valor de hipervolumen

Existen en la literatura muchos algoritmos capaces de calcular el valor exacto de hipervolumen. No obstante, no hay ninguno que pueda calcular dicho valor en tiempo polinomial, está circunstancia a motivado a buscar formas menos costosas de realizar el cálculo de este popular indicador. Los algoritmos que aproximan el valor de hipervolumen a pesar de la pérdida de precisión, siguen siendo capaces de clasificar y comparar de forma fiable los conjuntos de soluciones, lo que los vuelve una opción viable. A continuación, se mencionan algunos de los algoritmos para el



cálculo del valor exacto de hipervolumen, así como el algoritmo más representativo utilizado para la aproximación del valor de hipervolumen.

El algoritmo de *inclusión-exclusión* [5] para el cálculo de hipervolumen está basado en el principio de inclusión-exclusión de matemática combinatoria. Este método es uno de los más simples de aplicar y entender, el cual trabaja de la siguiente manera: El algoritmo agrega volúmenes de polítopos rectangulares dominados individualmente, enseguida resta volúmenes dominados por intersecciones de pares de puntos, luego vuelve a sumar volúmenes de polotipos dominados por intersecciones de tres puntos, y así de forma iterativa. Aunque sencillo este algoritmo tiene una complejidad de  $\mathcal{O}(n2^m)$  que lo hace exponencial respecto al número de puntos.

El algoritmo *Lebesgue Measure* [8] está basado en la eliminación de polotipos rectangulares que no se cruzan con ninguna región de un espacio cubierto por un conjunto de puntos no dominados. La contribución de hipervolumen de las regiones eliminadas es calculada y acumulada para incrementar el valor de hipervolumen, además se generan nuevos puntos para reflejar la región eliminada. Este proceso se repite hasta que los polotipos restantes no dominen ninguna región del espacio objetivo. La complejidad de este algoritmo fue demostrada empíricamente por While [9], quien determinó que este algoritmo tiene una complejidad de tiempo exponencial. La medida de *Lebesgue* [8] para el peor de los casos tiene una complejidad de  $\mathcal{O}(2^{m-1})$ , exponencial en el número de objetivos.

El algoritmo *Hypervolume by Slicing Objectives* (HSO) [10–12] es un algoritmo basado en barrido de dimensiones, el cual realiza el cálculo de hipervolumen procesando un objetivo a la vez, trazando secciones transversales a lo largo del objetivo elegido. El algoritmo HSO en primera instancia ordena los valores en el primer objetivo, los puntos ordenados de este objetivo se utilizan para crear cortes transversales. Al barrer los objetivos cada punto del objetivo se visita una vez. Se genera una



lista de los puntos procesados que se clasifica en  $n-1$  objetivos, cada lista contiene la profundidad de la sección correspondiente al primer objetivo. El hipervolumen se calcula de forma recursiva multiplicando por la profundidad del segmento. Una vez que se elimine cualquier punto que domine el punto procesado, este es agregado a una nueva lista  $n-1$ -objetivo. El proceso se repite para cada punto de la lista, sumando la contribución de hipervolumen de cada lista. El valor de hipervolumen total se obtiene sumando la contribución de hipervolumen de cada lista. Este algoritmo muestra una mejora respecto al algoritmo inclusión-exclusión [5] y *Lebesgue* [9] con un costo computacional  $\mathcal{O}(m^{n-1})$ . Sin embargo, su complejidad sigue siendo exponencial al número de objetivos.

El *3D Hypervolume Algorithm* [45] es un algoritmo propuesto por Paquete, para calcular de manera óptima el valor de hipervolumen en problemas de tres dimensiones, este algoritmo al igual que el anterior está basado en barrido de dimensiones, el algoritmo recorre una lista ordenada en un objetivo manteniendo un área 2D para los puntos considerados. Para cada punto se consulta un árbol binario para determinar la posición del punto en los demás objetivos. Si el punto es dominado es descartado, si el punto domina otros puntos el árbol es eliminado. La contribución de hipervolumen es obtenida de la multiplicación de la altura o profundidad del punto en el árbol determinada por el siguiente punto. El valor total de hipervolumen se obtiene sumando las contribuciones de hipervolumen de todos los puntos. Este algoritmo se limita solo al cálculo de hipervolumen de problemas de 3 funciones objetivo.

*The kles's measure problem (kmp)* [7] es un método utilizado en geometría computacional para calcular el tamaño de la unión de un conjunto de hipercubos  $n$ -dimensionales. Existe una gran relación entre este método y el cálculo de hipervolumen, por lo que se han propuesto varios algoritmos para calcular el valor de hipervo-



lumen basados en *kpm*. El algoritmo *Overmars y Yap* [46] es un método utilizado para calcular *kpm* utilizando hipercubos anclados a un punto de referencia. Beame y Rudolph [6] adaptaron el algoritmo *Overmars y Yap* [46] para dividir el espacio cubierto por un polotipo formado por cada punto no dominado. Este algoritmo lleva por nombre *Hypervolume Overmars y Yap (HOY)* [6]. El algoritmo *HOY* comienza a trabajar con la región más pequeña del polotipo rectangular anclada a un punto de referencia que contiene todos los puntos del frente. esta región es dividida en dos más pequeñas de forma recursiva. Se considera que un punto interseca una región si dicho punto se encuentra dentro de la región para la división actual y para al menos un objetivo procesado. La región es dividida en el objetivo actual utilizando los puntos de intersección, si no hay puntos de intersección la región es dividida por puntos que no se cruzan, si no hubiese puntos se aumenta el objetivo de división. Cada región es dividida hasta que se forma un enrejado, con los enrejados se logra que el hipervolumen cubra la región en todos los objetivos. Finalmente se obtiene el valor total de hipervolumen de una forma rápida utilizando el algoritmo de *inclusión-exclusión*. La complejidad de este algoritmo para  $d \geq 3$  objetivos es  $\mathcal{O}(n \log(n) + n^{\frac{m}{2}} \log(n))$ .

El algoritmo *FLP* [13] fue propuesto Fonseca, Paquete y López-Ibáñez el cual es una mejora al algoritmo *HSO* [10–12]. El algoritmo *FLP* incorpora el algoritmo *3D Hypervolume Algorithm* [45] propuesto por Paquete como base para ser utilizado en los problemas de tres funciones objetivo, además mejora la estructura de datos vinculados para reducir el trabajo de mantener los frentes construidos por *HSO*. En lugar de eliminar los puntos dominados los marca y los guarda para reinsertarlos en orden inverso de su eliminación, lo que permite mejorar el rendimiento al minimizar el número de comparaciones necesarias para mantener la clasificación de los cortes. Cuando ya se han calculado valores de hipervolumen de cortes anteriores, estos son almacenados junto con la coordenada actual en el objetivo actual, estos



valores son reutilizados y eliminados una vez que se vuelven obsoletos. Con las mejoras realizadas el algoritmo *FLP* redujo la complejidad del algoritmo *HSO* [12] para el peor de los casos a  $\mathcal{O}(n^{m-2}\log(n))$  [13].

El algoritmo propuesto por *Walking Fish Group (WFG)* [47] está basado en el cálculo de contribuciones únicas de hipervolumen. Técnica que funciona limitando los valores de puntos para un frente en la región delimitada por un punto que contribuye al valor total de hipervolumen. El hipervolumen de un conjunto de soluciones se calcula limitando el conjunto subyacente con el punto contribuyente y restando el hipervolumen del conjunto modificado al hipervolumen del conjunto inclusivo. Este algoritmo puede verse como un tipo de método de inclusión-exclusión [5] que utiliza la dominancia para reducir el número y el tamaño de los conjuntos que deben examinarse. Este algoritmo tiene la misma complejidad que el algoritmo propuesto por Fonseca, Paquete y López [13]. Es importante resaltar que el algoritmo *FLP* tiene una mejor implementación que el algoritmo *HSO* [10–12]. *HSO* tiene una ventaja adicional sobre *Lebesgue* [8]. Y en base a Lucas Bradstreet [48] el algoritmo *WFG* [47] puede mejorar un poco más la viabilidad del cálculo exacto de hipervolumen. No obstante, aunque estén considerados como los mejores algoritmos, sigue siendo complicado utilizarlos en problemas de alta dimensionalidad.

### 2.6.3. Algoritmos para estimar el valor y contribución de hipervolumen

El cálculo exacto del valor de hipervolumen sigue siendo inviable para problemas que tienen un gran número de funciones objetivo. Una buena opción es aproximar el valor de hipervolumen de un conjunto de soluciones. Este método permite aumentar el rendimiento de los algoritmos significativamente, a expensas de una pérdida en la precisión de los resultados. Los algoritmos que aproximan el valor de hipervolumen deben considerar el compromiso entre la precisión y eficiencia [49, 50]. Este compromiso será aceptable siempre y cuando la precisión este dentro



de los límites tolerables. Las técnicas de aproximación de hipervolumen se han convertido en un área importante de investigación. Existen varios enfoques [19, 51, 52] para aproximar cálculos costosos de hipervolumen. Sin embargo, el más representativo es algoritmo *Hypervolume Estimation Algorithm for Multi-Objective Optimization* (*HypE*) propuesto por Bader y Zitzler [15].

El algoritmo *HypE* [15] utiliza un enfoque de división y agregación para evitar calcular la contribución de hipervolumen para todas las combinaciones de subconjuntos. El algoritmo corta el espacio dominado de forma recursiva y en cada nivel de recursión se hace una exploración a lo largo de un objetivo en específico que representa la posición actual de exploración, utiliza un vector para contener todas las posiciones de las dimensiones de exploración, para cada iteración los vectores objetivo y los puntos de referencia son filtrados de acuerdo con las posiciones del vector, en base a las posiciones también se pueden seleccionar las soluciones dominadas. La contribución de hipervolumen es actualizada en cada iteración en base a la distancia de la siguiente posición de exploración.

Para problemas de más de tres objetivos *Hype* [50] utiliza muestreo de Monte Carlo, método conocido por ser fácil de utilizar para resolver problemas numéricos utilizando un número determinado de muestras generadas de forma aleatoria. Primero se establece la cantidad de muestras a generar. Para cada muestra generada se verifica:

- Que el punto de muestreo este por debajo del conjunto de referencia.
- Que en el conjunto de soluciones exista algún miembro que domine el punto de muestreo.

Si las condiciones anteriores se cumplen se denota un acierto, en caso contrario se denota un fallo. La estimación del valor de hipervolumen se obtiene contando el número de aciertos y multiplicándolo por el total de muestras generadas. La pre-



cisión de las estimaciones realizadas por *HypE* se puede mejorar incrementando el número de muestras en el espacio objetivo. No obstante, entre mayor sea el número de muestras el esfuerzo computacional requerido para la estimación de hipervolumen será mayor. La complejidad de este algoritmo para el peor de los casos es  $\mathcal{O}(n^m + m \cdot n \log(n))$  [15].

#### 2.6.4. Algoritmo SMS-EMOA basado en hipervolumen dominado

El algoritmo SMS-EMOA fue propuesto por Beume y Emmerich [17] en 2007, este algoritmo utiliza el indicador hipervolumen como criterio de selección. Su proceso evolutivo inicia generando una población aleatoria. A continuación se genera un nuevo individuo utilizando operadores de variación, el nuevo individuo es agregado a la población actual, para posteriormente realizar un ordenamiento no dominado utilizando el algoritmo *fast nondominated sort* [39], método utilizado en el algoritmo NSGA-II [39]. El ordenamiento no dominado consiste en clasificar a la población en niveles o jerarquías de dominancia  $R_1, \dots, R_v$  en donde cada conjunto  $R_i$  contiene soluciones no dominadas entre sí. La contribución de hipervolumen es asignada como aptitud a cada individuo del último frente de soluciones no dominadas. El individuo que menos contribuye al valor de hipervolumen es eliminado de la población. El proceso se repite hasta que una condición de paro es satisfecha. El algoritmo SMS-EMOA es un algoritmo muy potente que se adapta de manera positiva a problemas difíciles. Las características del indicador hipervolumen permiten mantener la calidad de las soluciones al aumentar el número de objetivos. Sin embargo, persiste la desventaja de la complejidad de orden exponencial de los algoritmos existentes para calcular la contribución de hipervolumen. Causa que sigue limitando el uso del SMS-EMOA en problemas de muchos objetivos. El algoritmo 2.4 muestra la estructura básica del algoritmo SMS-EMOA.



---

**Algoritmo 2.4** Estructura básica del algoritmo SMS-EMOA

---

**Entrada:** Problema a optimizar, Tamaño de población, No. de generaciones

**Salida:** Aproximación al frente de Pareto

1.  $P_0 \leftarrow$  inicializar Población de forma aleatoria
  2.  $t \leftarrow 0$
  3. **while**  $t <$  No. de generaciones **do**
  4.  $q_{t+1} \leftarrow$  generar( $P_t$ ) Generar descendencia por variación
  5.  $P_{t+1} \cup \{q_{t+1}\}$  Incorporar nuevo individuo a población
  6.  $\{\mathcal{R}_1, \dots, \mathcal{R}_v\}$  Clasificar población con *fast-nondominated-sort* algorithm
  7.  $r \leftarrow \operatorname{argmin}_{s \in \mathcal{R}_v} [\Delta_{\mathcal{P}}(s, \mathcal{R}_v)]$  Calcular contribución de hipervolumen.
  8.  $r \leftarrow (Q \setminus \{r\})$  Eliminar individuo menor contribución
  9.  $t = t + 1$
  10. **end while**
-

## Capítulo 3

# Programación Genética

Programación Genética (por sus siglas en inglés, *Genetic Programming*: GP) es un método de programación automática que produce programas a través de cálculo genético y evolutivo. GP fue desarrollado y popularizado por John Koza [53] en 1992 en su libro *Genetic Programming* en el cual acuñó el término y sentó las bases formales de esta técnica. Existen trabajos previos que pueden ser considerados los precursores de la materia. Cramer [54] en 1985 ya había hecho alusión a otras ideas similares, sin usar explícitamente el término de Programación Genética. El planteaba un sistema adaptativo para la generación de pequeños programas secuenciales. Hicklin [55] en 1986 aplicó la reproducción y mutación para generar programas en el lenguaje LISP. Posteriormente Fujiki [56] en 1986 utilizó todos los operadores genéticos para generar programas LISP. Más tarde en 1987 Fujiki y Dickenson [57] utilizaron los operadores genéticos para manipular las cláusulas *if – then* en un programa que especifica la estrategia para la resolución de un juego iterativo clásico llamado el del dilema del prisionero.

GP surge como la evolución de los algoritmos genéticos tradicionales, manteniendo el mismo principio de selección natural. La diferencia entre los GAs y GP radica en la forma de codificación de los problemas.

### 3.1. Codificación de los programas

En GP los programas se pueden representar de varias maneras, pero los árboles de sintaxis son la representación tradicional utilizada en la literatura. En la figura 3.1 y 3.2 se muestran árboles de sintaxis que representan cromosomas utilizados en Programación Genética.

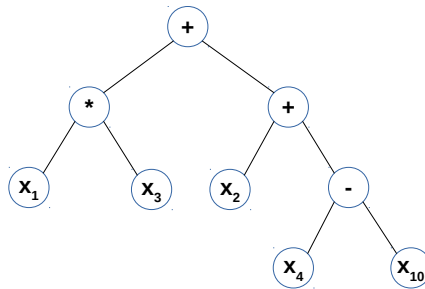


Figura 3.1: Representación de un árbol de sintaxis operadores aritméticos.

Existen diferentes operadores especializados para evolucionar programas generados aleatoriamente para resolver determinada tarea específica, aunque los principios básicos de este método pueden generalizarse a cualquier otro dominio. Las funciones utilizadas podrían ser aritméticas, trigonométricas, booleanas, condicionales, ciclos, funciones recursivas, o cualquier otra función definida en el dominio utilizado.

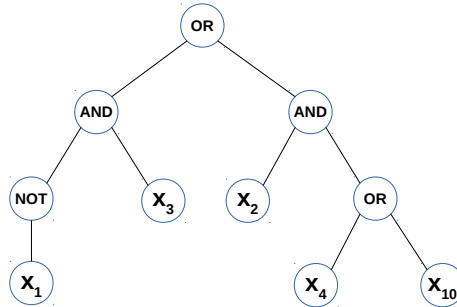


Figura 3.2: Representación de un árbol de sintaxis operadores lógicos.

### 3.1.1. Elementos del árbol de sintaxis

Para la representación de los árboles de sintaxis en GP existen dos tipos de nodos: Los nodos terminales u hojas del árbol. Son aquellos que no tienen hijos y que típicamente son variables o constantes que no toman argumentos. Por ejemplo nodo  $Terminal = \{X_1, X_2, \dots, X_n\}$ .

Los nodos función son aquellos que tienen uno o más hijos, generalmente se asocian con los operadores que se quieren desarrollar. Por ejemplo nodo  $Función = \{sin, cos, tan, and, or, not, \dots, etc\}$ . En la figura 3.3 se señalan los nodos que conforman un árbol de sintaxis.

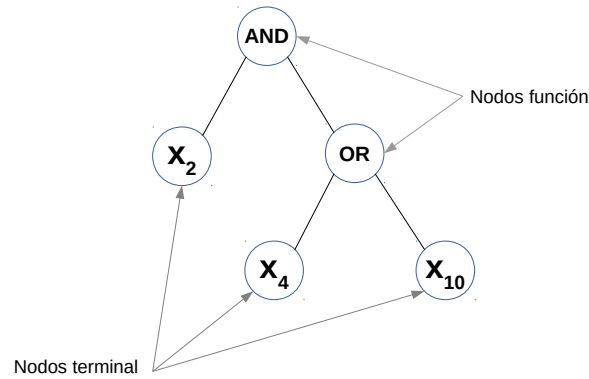


Figura 3.3: Nodos de un árbol de sintaxis.

Cuando se especifican los conjuntos de elementos y terminales, es necesario que se especifiquen dos requisitos los cuales fueron definidos por Koza [53]. El requisito de *suficiencia* el cual dice que la solución al problema debe poder ser especificada con el conjunto de operadores especificados y el conjunto de *cerradura* el cual implica que cualquier árbol que se construya con estos operadores debe ser correcto.

En la condición de *cerradura* se aplica algunas restricciones a los árboles generados, ya que cualquier árbol construido por los operadores debe de ser correcto se impone restricción al momento de la elección de terminales y funciones. Otra restricción habitual es restringir la altura máxima de los árboles, esta restricción ayuda a evitar que los árboles posean mucho código redundante provocado por el crecimiento desmesurado el cual provoca un fenómeno conocido como *bloat* [58]. Por tal motivo, es necesario poner un límite al tamaño que pueden alcanzar los árboles, esta restricción permitirá a los árboles protegerse ante posibles modificaciones negativas originadas por los operadores de cruce y mutación.

### 3.2. Proceso evolutivo

Al igual que en otros algoritmos evolutivos, en GP los individuos en la población inicial se generan de forma aleatoria. Koza [53] propuso una combinación llamada *ramped half-and-half*, en donde la mitad de la población se construye utilizando *full and half* y la otra mitad se construye usando *grown*. Esto se hace utilizando un rango de límites de profundidad que garantiza que se generen árboles que tengan una variedad de formas y tamaños. Si se sabe algo sobre las propiedades de la probable solución deseada la población inicial no necesariamente necesita ser completamente al azar. Determinados árboles con ciertas propiedades pueden utilizarse para generar la población inicial. En la figura 3.4 se muestra un conjunto de individuos que conforman una población inicial.

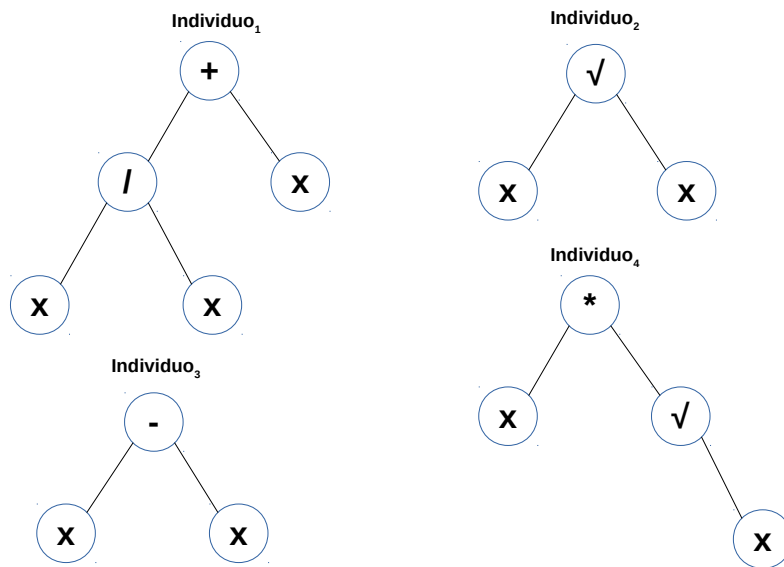


Figura 3.4: Ejemplo de población inicial.



### 3.2.1. Operadores primarios

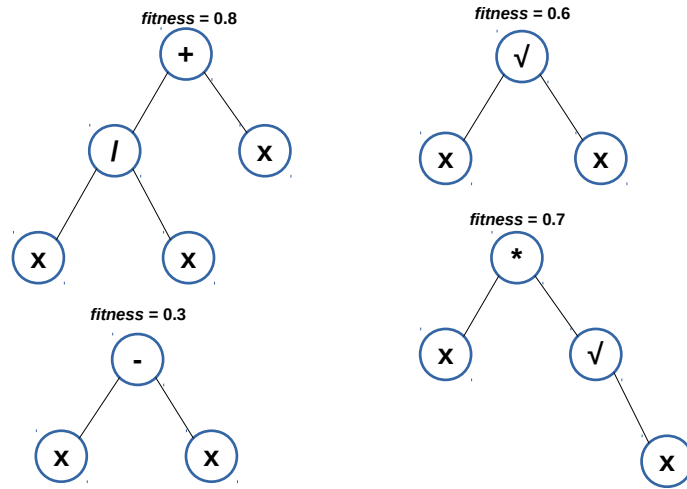
### 3.2.2. Selección

La selección de los individuos se realiza en base a su aptitud física (*fitness*). Los individuos que son más aptos físicamente tienen una mayor probabilidad de reproducirse y generar descendencia. Los métodos más comúnmente utilizados en GP para seleccionar individuos son la selección por torneo, seguida de la selección proporcional de aptitud. No obstante, se puede utilizar cualquier selección de algoritmo evolutivo estándar.

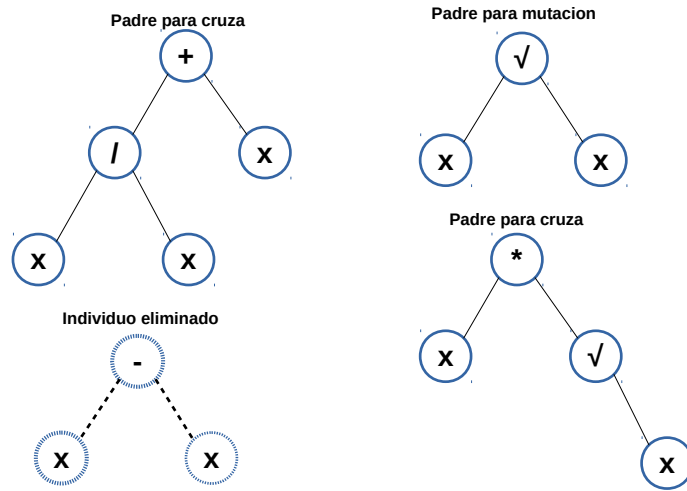
En la selección por torneo se elige una cantidad aleatoria de individuos de la población actual. Enseguida se evalúa la aptitud física de cada individuo y el más apto es elegido para ser padre. Para la recombinación son necesarios dos padres, por lo tanto, se realizan dos torneos de selección. Una de las características de la selección por torneo, es que garantiza que incluso los programas de calidad promedio tengan alguna posibilidad de tener hijos. Característica que permite mantener la diversidad de la población. En la figura 3.5 se ejemplifica la evaluación física(a) de los individuos, así como la selección (b) de los individuos que generara descendencia.

### 3.2.3. Cruza

El operador de cruce o recombinación sexual crea variación en la población al producir descendencia que consiste en partes tomadas de cada padre. El cruce entre los padres se efectúa mediante el intercambio de sub-árboles. La cruce puede efectuarse numerando los nodos de los árboles correspondientes entre los dos padres elegidos y seleccionando al azar un punto en cada uno de ellos de manera que los sub-árboles por debajo de dicho punto se intercambien. Los hijos se generan al intercambiar el sub-árbol enraizado en el punto de cruce en una copia del primer padre con una copia del sub-árbol enraizado en el punto de cruce en el segundo



(a) Evaluación



(b) Selección

Figura 3.5: Selección de los individuos más aptos.

padre. Si el punto cruza es la raíz de uno de los dos padres, entonces todo ese individuo se volverá un sub-árbol del otro padre, lo cual permite la incorporación de subrutinas en un programa. Si las raíces de ambos padres son elegidos como punto de cruce entonces no se efectúa la cruce y los hijos serán idénticos a los padres. Al realizar el cruce, los nodos escogidos en ambos padres no suelen tomarse al azar, generalmente se asigna una probabilidad. Koza [53] en su libro sugirió utilizar una probabilidad de cruce alta que corresponde a 0.9.

En la figura 3.6 y 3.7 se muestra la cruce de los individuos más aptos, así como la descendencia generada.

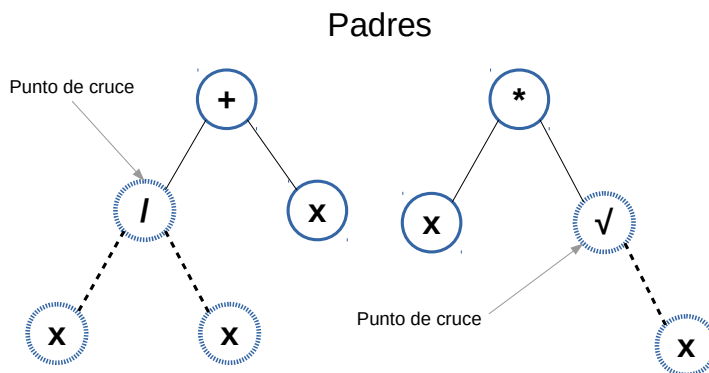


Figura 3.6: Individuos seleccionados para cruce.

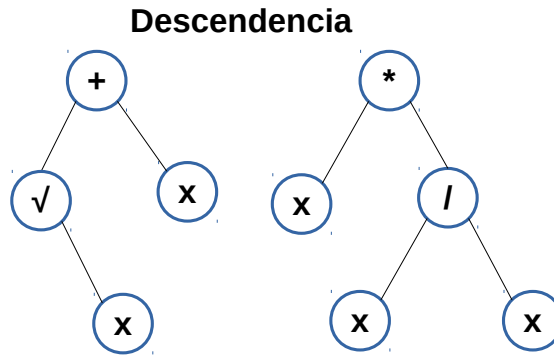


Figura 3.7: Hijos generados por el operador de cruce.

### 3.2.4. Mutación

El operador de mutación provoca la variación de un individuo de la población, este operador suele utilizarse con una probabilidad muy baja, Koza [53] sugiere una probabilidad de mutación menor a 0.1. La mutación más utilizada en GP es la mutación de sub-árbol, donde, se selecciona aleatoriamente un nodo de mutación en el árbol y se sustituye el árbol enraizado en ese nodo con un sub-árbol generado aleatoriamente. La mutación de sub-árbol a veces se implementa como un cruce entre un programa y un programa aleatorio recién generado. Otra forma común de notación es la notación puntual, en la notación puntual se selecciona un nodo aleatorio y la primitiva almacenada ahí es reemplazada por una primitiva aleatoria diferente.

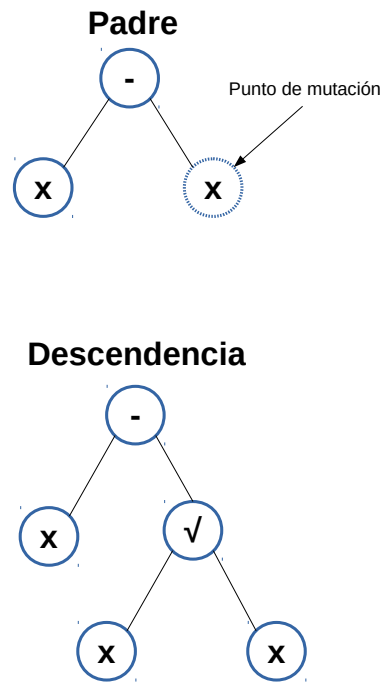


Figura 3.8: Ejemplo de mutacion.

### 3.3. Parámetros de control

En base a Koza [53] el paradigma GP está controlado por diez y nueve parámetros de control, los cuales se dividen en dos parámetros numéricos principales, once parámetros numéricos menores, y seis variables cualitativas que se utilizan entre varias formas alternativas de ejecución. Los más importantes se en listan a continuación:

- **Tamaño de la población:** Este parámetro indica el número de individuos que conformaran la población.
- **Altura máxima de los árboles:** Indica la altura máxima que van a poder al-



canzar los árboles durante el proceso evolutivo.

- **Altura máxima de los árboles que conformaran la población inicial.**
- **Altura máxima que tendrán los sub-arboles.**
- **Probabilidad de cruce:** Define la probabilidad de cruce de los individuos. Suele ser mayor a 90 %.
- **Probabilidad de mutación:** Define la probabilidad de mutación de los individuos. Suele ser menor a 10 %.
- **Elección de los algoritmos de selección, cruce y mutación.**
- **Probabilidad de selección nodo terminal:** Probabilidad que se utiliza para elegir los nodos de árbol el proceso de cruza.
- **Coefficiente de parsimonia:** Penalización de árboles grandes.
- **Criterio de paro:** Parecido a los algoritmos genéticos, se puede utilizar un número máximo de generaciones, valor de *fitness* de ajuste aceptable, convergencia de población.

### 3.4. Algoritmo GPTIPS

GPTIPS (Por sus siglas en ingles *Genetic Programming Toolbox for the Identification of Physical System*) es un algoritmo de aprendizaje automático de Programación Genética prepuesto por Dominic P. Searson [59] en 2010. Este algoritmo utiliza una variante de múltiples genes en los cuales cada solución está compuesta por múltiples árboles independientes. Este algoritmo trabaja con una población de tamaño fijo, sub-mutaciones, cruce de sub-árboles, selección por torneos y procedimientos



### *CAPÍTULO 3. PROGRAMACIÓN GENÉTICA*

---

de inicialización estándar. Para limitar la complejidad de los modelos GPTIPS utiliza por defecto la presión lexicográfica de parsimonia [60]. La salida de GPTIPS es la combinación lineal de todos los genes representados como un solo modelo.

## Capítulo 4

# Funciones de prueba

Durante mucho tiempo se subestimó la necesidad de contar con funciones de prueba estándar. No obstante, hoy en día existen propuestas que incluyen funciones escalables. Entre sus características destaca la multi-frontalidad, frentes de Pareto cóncavos, segmentados y degenerados. Otra característica deseable es que las funciones puedan ser escalables a cualquier número de funciones objetivo.

### 4.1. Conjunto de prueba DTLZ

Deb, Thiele, Laumanns y Zitzler propusieron un conjunto de prueba conocido como DTLZ [61, 62], en este conjunto todos sus problemas son de múltiples objetivos  $n$ -dimensionales, escalables en dimensión de aptitud física. Este conjunto de problemas es muy popular en la literatura especializada utilizándose sobre todo con tres funciones objetivo. Este trabajo de tesis utilizó este conjunto con tres y cuatro funciones objetivo. Los problemas utilizados se describen a continuación.

### 4.1.1. Problema DTLZ1

El problema DTLZ1 es un problema de minimización lineal, separable y multimodal. Su conjunto de óptimos de Pareto se encuentra en  $x_M^* = 0$ , el valor de sus funciones objetivo están el hiperplano lineal  $\sum_{m=1}^M = 0,5$ . El espacio de búsqueda contiene  $11^k - 1$  frentes de Pareto locales con  $n = M + k - 1$  variables de decisión. Su frente de Pareto real se muestra en la figura 4.1.

**Minimizar:**

$$f_1(x) = \frac{1}{2}x_1x_2 \cdots x_{M-1}(1 + g(x_M))$$

$$f_2(x) = \frac{1}{2}x_1x_2 \cdots (1 - x_{M-1})(1 + g(x_M))$$

$\vdots$

$$f_{M-1}(x) = \frac{1}{2}x_1(1 - x_2)(1 + g(x_M))$$

$$f_M(x) = \frac{1}{2}(1 - x_2)(1 + g(x_M))$$

**Sujeto a:**

$$0 \leq x_i \leq 1 \forall i = 1, 2, \dots, n$$

**Donde:**

$$g(x_M) = 100 \left[ |x_M| + \sum_{x_i \in x_M} (x_i - 0,5)^2 - \cos(20\pi(x_i - 0,5)) \right]$$

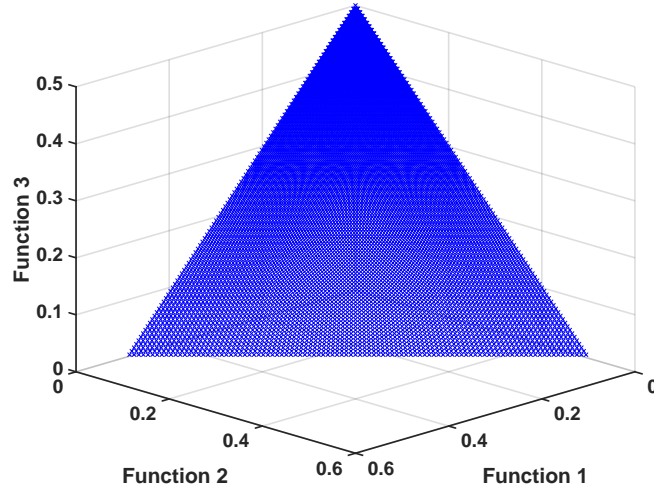


Figura 4.1: Frente de Pareto problema DTLZ1.

#### 4.1.2. Problema DTLZ2

El problema DTLZ2 tiene un frente de Pareto cóncavo, con un espacio de búsqueda continuo y unimodal. El conjunto de óptimos de Pareto está en  $x_i = 0,5$  para toda  $x_i \in x_M$ , sus funciones objetivo están en  $\sum_{i=1}^M (f_i)^2$ , con  $n = M + k - 1$ . Su frente de Pareto real se muestra en la figura 4.2.

**Minimizar:**

$$f_1(x) = (1 + g(x_M)) \cos(x_1 \frac{\pi}{2}) \cos(x_2 \frac{\pi}{2}) \cdots \cos(x_{M-2} \frac{\pi}{2}) \cos(x_{M-1} \frac{\pi}{2})$$

$$f_2(x) = (1 + g(x_M)) \cos(x_1 \frac{\pi}{2}) \cos(x_2 \frac{\pi}{2}) \cdots \cos(x_{M-2} \frac{\pi}{2}) \sin(x_{M-1} \frac{\pi}{2})$$

$$f_3(x) = (1 + g(x_M)) \cos(x_1 \frac{\pi}{2}) \cos(x_2 \frac{\pi}{2}) \cdots \sin(x_{M-2} \frac{\pi}{2})$$

⋮

$$f_{M-1}(x) = (1 + g(x_M)) \cos(x_1 \frac{\pi}{2}) \sin(x_2 \frac{\pi}{2})$$

$$f_M(x) = (1 + g(x_M)) \sin(x_1 \frac{\pi}{2})$$

**Sujeto a:**

$$0 \leq x_i \leq 1 \forall i = 1, 2, \dots, n$$

**Donde:**

$$g(x_M) = \sum_{x_i \in X_M} (x_i - 0,5)^2$$

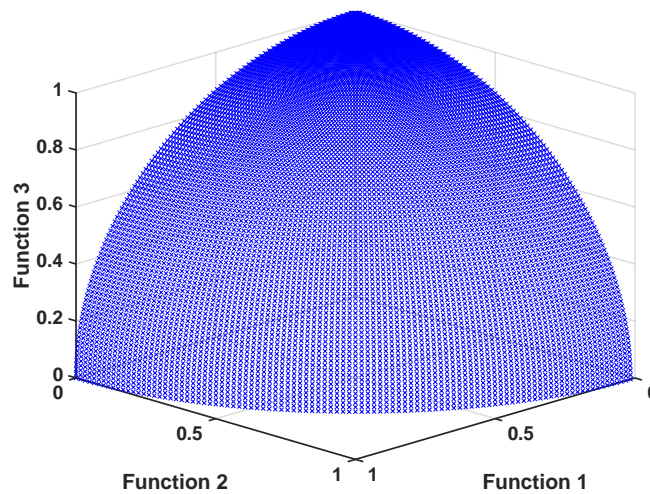


Figura 4.2: Frente de Pareto problema DTLZ2.

### 4.1.3. Problema DTLZ3

En el problema DTLZ3 el espacio de búsqueda es continuo, unimodal con un frente de Pareto cóncavo. Con  $(3^k - 1)$  frentes de Pareto locales y un frente de Pareto global que se encuentra en  $g^* = 0$ , con  $n = M + k - 1$  variables de decisión. Su frente de Pareto real se muestra en la figura 4.3

**Minimizar:**

$$f_1(x) = (1 + g(x_M))\cos(x_1 \frac{\pi}{2})\cos(x_2 \frac{\pi}{2}) \cdots \cos(x_{M-2} \frac{\pi}{2})\cos(x_{M-1} \frac{\pi}{2})$$

$$f_2(x) = (1 + g(x_M))\cos(x_1 \frac{\pi}{2})\cos(x_2 \frac{\pi}{2}) \cdots \cos(x_{M-2} \frac{\pi}{2})\sin(x_{M-1} \frac{\pi}{2})$$

$$f_3(x) = (1 + g(x_M))\cos(x_1 \frac{\pi}{2})\cos(x_2 \frac{\pi}{2}) \cdots \sin(x_{M-2} \frac{\pi}{2})$$

⋮

$$f_{M-1}(x) = (1 + g(x_M))\cos(x_1 \frac{\pi}{2})\sin(x_2 \frac{\pi}{2})$$

$$f_M(x) = (1 + g(x_M))\sin(x_1 \frac{\pi}{2})$$

**Sujeto a:**

$$0 \leq x_i \leq 1 \forall i = 1, 2, \dots, n$$

**Donde:**

$$g(x_M) = 100 \left[ |x_M| + \sum_{x_i \in x_M} (x_i - 0,5)^2 - \cos(20\pi(x_i - 0,5)) \right]$$

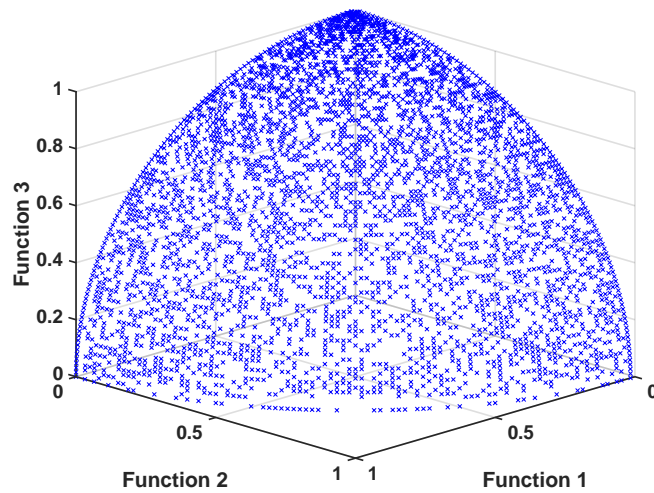


Figura 4.3: Frente de Pareto problema DTLZ3.

#### 4.1.4. Problema DTLZ4

El problema DTLZ4 tiene un frente de Pareto cóncavo, separable y unimodal que contiene un área densa de soluciones cerca del plano  $f_M - f_1$ . Todas las variables  $x_1$  a  $x_{M-1}$  están en el rango  $(0 : 1)$ , Con  $n = M + k - 1$  variables de decisión. Su frente de Pareto real se muestra en la figura 4.4.

**Minimizar:**

$$f_1(x) = (1 + g(x_M)) \cos(x_1^{\frac{\pi}{2}}) \cos(x_2^{\frac{\pi}{2}}) \cdots \cos(x_{M-2}^{\frac{\pi}{2}}) \cos(x_{M-1}^{\frac{\pi}{2}})$$

$$f_2(x) = (1 + g(x_M)) \cos(x_1^{\frac{\pi}{2}}) \cos(x_2^{\frac{\pi}{2}}) \cdots \cos(x_{M-2}^{\frac{\pi}{2}}) \sin(x_{M-1}^{\frac{\pi}{2}})$$

$$f_3(x) = (1 + g(x_M)) \cos(x_1^{\frac{\pi}{2}}) \cos(x_2^{\frac{\pi}{2}}) \cdots \sin(x_{M-2}^{\frac{\pi}{2}})$$

⋮

$$f_{M-1}(x) = (1 + g(x_M)) \cos(x_1^{\frac{\pi}{2}}) \sin(x_2^{\frac{\pi}{2}})$$

$$f_M(x) = (1 + g(x_M)) \sin(x_1^{\frac{\pi}{2}})$$

**Sujeto a:**

$$0 \leq x_i \leq 1 \forall i = 1, 2, \dots, n$$

**Donde:**

$$g(x_M) = \sum_{x_i \in X_M} (x_i - 0,5)^2$$

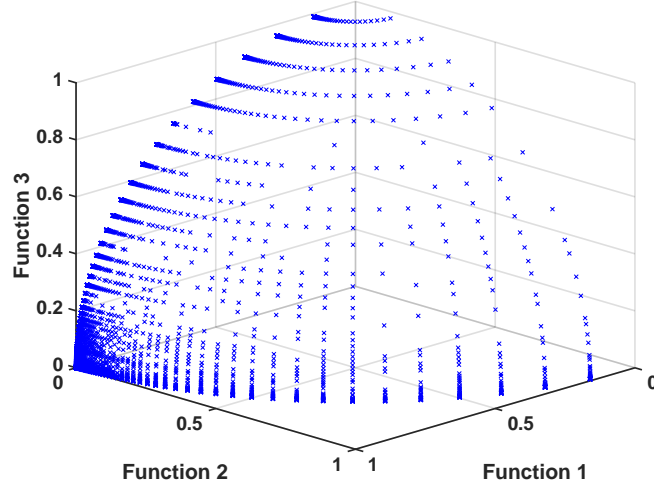


Figura 4.4: Frente de Pareto problema DTLZ4.

#### 4.1.5. Problema DTLZ5

El problema DTLZ5 tiene frentes óptimos degenerados de Pareto; los frentes óptimos de Pareto están destinados a ser un arco incrustado en el espacio objetivo. El conjunto de óptimos de Pareto corresponde a  $x_i = 0,5$  para todo  $x_i \in x_M$ , todas sus funciones objetivo se deben satisfacer para  $\sum_{i=1}^M (f_i)^2 = 1$ , con  $n = M + k - 1$  variables de decisión. Su frente de Pareto real se muestra en la figura 4.5.

**Minimizar:**

$$f_1(x) = (1 + g(x_M)) \cos(\theta_1 \frac{\pi}{2}) \cos(\theta_2 \frac{\pi}{2}) \cdots \cos(\theta_{M-2} \frac{\pi}{2}) \cos(\theta_{M-1} \frac{\pi}{2})$$

$$f_2(x) = (1 + g(x_M)) \cos(\theta_1 \frac{\pi}{2}) \cos(\theta_2 \frac{\pi}{2}) \cdots \cos(\theta_{M-2} \frac{\pi}{2}) \sin(\theta_{M-1} \frac{\pi}{2})$$

$$f_3(x) = (1 + g(x_M)) \cos(\theta_1 \frac{\pi}{2}) \cos(\theta_2 \frac{\pi}{2}) \cdots \sin(\theta_{M-2} \frac{\pi}{2})$$

⋮

$$f_{M-1}(x) = (1 + g(x_M))\cos(\theta_1 \frac{\pi}{2})\sin(\theta_2 \frac{\pi}{2})$$

$$f_M(x) = (1 + g(x_M))\sin(\theta_1 \frac{\pi}{2})$$

**Sujeto a:**

$$0 \leq x_i \leq 1 \forall i = 1, 2, \dots, n$$

**Donde:**

$$\theta = \frac{\pi}{4(1 + g(x_M))} (1 + 2g(x_M)x_i), \text{ for } i = 2, 3, \dots, (M - 1)$$

$$g(x_M) = \sum_{x_i \in X_M} (x_i - 0,5)^2$$

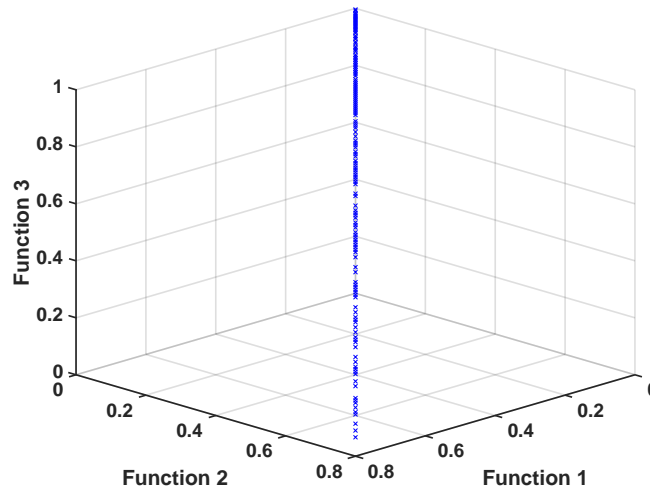


Figura 4.5: Frente de Pareto problema DTLZ5.

#### 4.1.6. Problema DTLZ6

El problema DTLZ6 tiene su conjunto de óptimos de Pareto en  $x_i = 0$  para toda  $x_i \in x_M$  su forma al igual que el DTLZ5 es un arco incrustado en el espacio objetivo. El número total de variables es idéntico al DTLZ5. Su frente de Pareto real



se muestra en la figura 4.6.

**Minimizar:**

$$f_1(x) = (1 + g(x_M))\cos(\theta_1 \frac{\pi}{2})\cos(\theta_2 \frac{\pi}{2}) \cdots \cos(\theta_{M-2} \frac{\pi}{2})\cos(\theta_{M-1} \frac{\pi}{2})$$

$$f_2(x) = (1 + g(x_M))\cos(\theta_1 \frac{\pi}{2})\cos(\theta_2 \frac{\pi}{2}) \cdots \cos(\theta_{M-2} \frac{\pi}{2})\sin(\theta_{M-1} \frac{\pi}{2})$$

$$f_3(x) = (1 + g(x_M))\cos(\theta_1 \frac{\pi}{2})\cos(\theta_2 \frac{\pi}{2}) \cdots \sin(\theta_{M-2} \frac{\pi}{2})$$

⋮

$$f_{M-1}(x) = (1 + g(x_M))\cos(\theta_1 \frac{\pi}{2})\sin(\theta_2 \frac{\pi}{2})$$

$$f_M(x) = (1 + g(x_M))\sin(\theta_1 \frac{\pi}{2})$$

**Sujeto a:**

$$0 \leq x_i \leq 1 \forall i = 1, 2, \dots, n$$

**Donde:**

$$\theta = \frac{\pi}{4(1 + g(x_M))}(1 + 2g(x_M)x_i), \text{ for } i = 2, 3, \dots, (M - 1)$$

$$g(x_M) = \sum_{x_i \in X_M} (x_i)^{0,1}$$

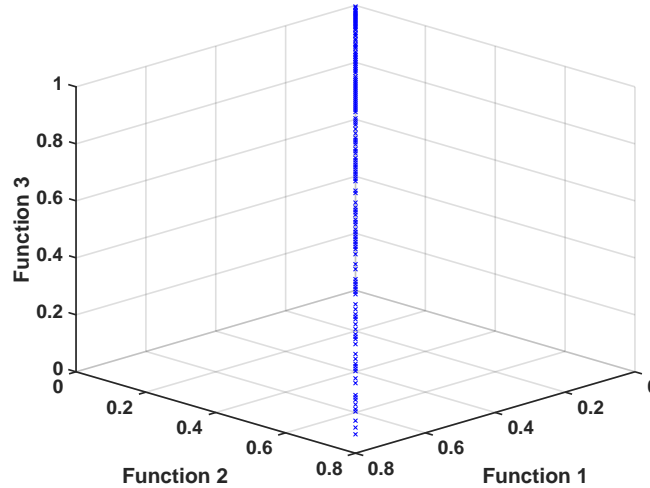


Figura 4.6: Frente de Pareto problema DTLZ6.

#### 4.1.7. Problema DTLZ7

El problema DTLZ7 es un problema que desconecta las regiones óptimas de Pareto del espacio de búsqueda. Este problema tiene  $2M - 1$  regiones de Pareto óptimas desconectadas, con  $n = M + k - 1$  variables de decisión. Su conjunto de óptimos de Pareto está en  $x_M = 0$ , este problema pretende evaluar la capacidad de un MOEA para mantener simultáneamente, soluciones en diferentes regiones del espacio de búsqueda. Su frente de Pareto real se muestra en la figura 4.7.

**Minimizar:**

$$f_1(x) = x_1$$

$$f_2(x) = x_2$$

⋮

$$f_{M-1}(x) = x_{M-1}$$

$$f_M(x) = (1 + g(x_M))h(f_1, f_2, \dots, f_{M-1}g(x))$$

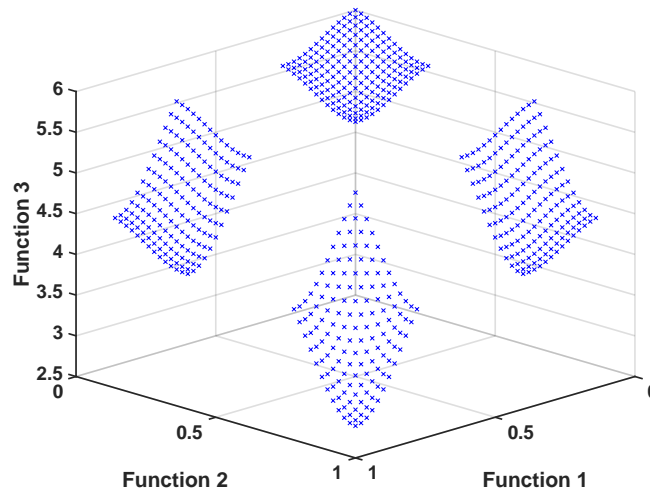
**Sujeto a:**

$$0 \leq x_i \leq 1 \forall i = 1, 2, \dots, n$$

**Donde:**

$$g(x) = 1 + \frac{9}{|x_M|} \sum_{x_i \in x_M} (x_i)$$

$$h(f_1, f_2, \dots, f_{M-1}, g) = M - \sum_{i=1}^{M-1} \left( \frac{f_i}{1 + g(x)} (1 + \sin(3\pi f_i)) \right)$$



**Figura 4.7:** Frente de Pareto problema DTLZ7.

Las propiedades de los siete problemas de prueba utilizados se resumen en la tabla 4.1.



**Cuadro 4.1:** Características problemas de prueba DTLZ.

Problema	Propiedades
DTLZ1	Lineal, separable, multimodal
DTLZ2	Cóncavo, continuo, unimodal
DTLZ3	Cóncavo, continuo, unimodal
DTLZ4	Cóncavo, separable, unimodal
DTLZ5	Cóncavo, degenerado
DTLZ6	Cóncavo, degenerado, sesgado
DTLZ7	Mixto, discontinuo, multimodal

## Capítulo 5

# Experimentos y resultados obtenidos

En este capítulo se presentan la metodología, experimentos y los resultados obtenidos de los estudios experimentales realizados a los modelos entrenados con Programación Genética. Estos modelos fueron utilizados para evaluar el desempeño de una aproximación al frente de Pareto realizado por un algoritmo de búsqueda. Del mismo modo, se presentan los resultados de la validación experimental del algoritmo que utiliza los modelos como método de selección. Para validar el desempeño se utilizó el algoritmo evolutivo basado en indicadores SMS-EMOA [17]. Se utilizaron siete problemas del conjunto de prueba DTLZ [61, 62]. La característica por la que se utilizan estos problemas es que son escalables conforme al número de objetivos. Los estudios experimentales realizados en este trabajo de tesis se dividen en cuatro partes: Primero se presentan los resultados obtenidos de los modelos utilizados como indicador para evaluar la calidad de conjuntos de soluciones no dominadas aproximadas por un algoritmo de búsqueda en problemas con tres funciones objetivo. La Segunda parte presenta los resultados de los modelos utilizados como método de selección del algoritmo SMS-EMOA en problemas con tres funciones objetivo. La



tercera parte muestra los resultados de los modelos como indicador para problemas con cuatro funciones objetivo. Y finalmente la cuarta parte muestra los resultados de los modelos como guía del algoritmo SMS-EMOA en problemas con cuatro funciones objetivo.

Todos los experimentos fueron ejecutados en la misma computadora, la cual tiene las siguientes características: Procesador intel i7 7th Gen @ 2.80Ghz, 16Gb de memoria RAM, GPU Geforce GTX 1050TI, Sistema operativo Windows 10 de 64 bits.

## 5.1. Metodología

El principal objetivo de este trabajo es generar modelos entrenados por GP que aproximen el valor de hipervolumen y puedan ser utilizados como criterio de selección en un algoritmo de búsqueda basado en el indicador hipervolumen. Se evaluará el comportamiento de los modelos generados en exactitud y tiempo. Se realizan dos tipos de experimento para los problemas DTLZ [62] con tres y cuatro funciones objetivo. En un primer tipo de experimentos se utilizarán los modelos como criterio de selección en el algoritmo SMS-EMOA [17] del framework PlatEMO [63] para aproximar los siete problemas del conjunto DTLZ. El desempeño se evaluará contra el algoritmo HypE [15] el cual es utilizado por defecto como criterio de selección en el evolutivo SMSEMOA en el framework PlatEMO. El segundo tipo de experimentos consiste en utilizar los modelos como indicador para medir la calidad de las soluciones de varios archivos extraídos del evolutivo SMS-EMOA para los siete problemas DTLZ. Este experimento se comparará en desempeño y tiempo contra el algoritmo WFG [47] de Walking Fish Group, el cual calcula el valor exacto de hipervolumen.



### 5.1.1. Bases de datos.

La fase previa de recolección y preparación de los datos utilizados para entrenar los modelos de GP se realizó de la siguiente manera: Primero todos los datos fueron extraídos del evolutivo SMS-EMOA [17] el cual fue configurado con los parámetros definidos en la tabla 5.1. La misma configuración fue utilizada para las bases de datos de tres y cuatro objetivos, variando únicamente el número de objetivos. Utilizando esta configuración se realizaron corridas independientes del algoritmo para cada uno de los siete problemas del benchmark DTLZ [62]. Durante el proceso del algoritmo en su búsqueda del verdadero frente de Pareto una gran cantidad de soluciones no dominadas son evaluadas en aptitud en su proceso de selección. De esta parte del proceso los datos son extraídos y almacenados. Cada décimo conjunto de puntos no dominados evaluado en aptitud por el algoritmo SMS-EMOA es guardado. Dependiendo de las características y propiedades de los diferentes problemas la cantidad de datos almacenados varía entre 4000 y 11000. Al tener como finalidad que los datos guarden como característica la evolución hacia el verdadero frente de Pareto de los distintos problemas, todos los datos son respetados, circunstancia que hace variar el tamaño de las bases para cada problema. Los tamaños de las bases de datos para los diferentes problemas DTLZ se definen en la tabla 5.2.

**Cuadro 5.1:** Parámetros utilizados en el algoritmo SMSEMOA.

Parámetro	Configuración
Tamaño Población	100
No. de generaciones	100
Simulación binaria de cruza(SBX)	
Probabilidad de cruza	0.9
Indice de distribución de cruza	20
Mutación polinomial	
Probabilidad de mutación	$\frac{1}{n}$
Indice de distribución de mutación	20



## CAPÍTULO 5. EXPERIMENTOS Y RESULTADOS OBTENIDOS

**Cuadro 5.2:** Tamaño de las bases de datos de problemas DTLZ.

Problema	No. de conjuntos de soluciones no dominadas
DTLZ1	428
DTLZ2	597
DTLZ3	504
DTLZ4	1083
DTLZ5	494
DTLZ6	492
DTLZ7	1146

Para cada conjunto de puntos no dominados se calculó el valor real de hipervolumen utilizando el algoritmo WFG [47]. Del mismo modo para cada conjunto de soluciones se obtuvo un vector de rasgos estadísticos el cual está conformado por la media, mediana, desviación estándar, percentil 25, percentil 75, curtosis y skewness. Este vector estadístico contiene las variables predictoras que en conjunto con el valor real de hipervolumen el cual representa la variable objetivo, permiten plantear un problema de aprendizaje supervisado. En la figura 5.3 se muestra las características que conforman el vector de variables predictoras de cada solución no dominada.

**Cuadro 5.3:** Rasgos estadísticos que describen un conjunto de puntos no dominados de un problemas de 3 funciones objetivo.

Rasgos	Variables predictoras en el vector de características		
	Objetivo 1	Objetivo 2	Objetivo 3
Media	$X_1$	$X_2$	$X_3$
Mediana	$X_4$	$X_5$	$X_6$
Desviación estándar	$X_7$	$X_8$	$X_9$
Percentil 25	$X_{10}$	$X_{11}$	$X_{12}$
Percentil 75	$X_{13}$	$X_{14}$	$X_{15}$
Curtosis	$X_{16}$	$X_{17}$	$X_{18}$
Skewness	$X_{19}$	$X_{20}$	$X_{21}$
Reference point	$X_{22}$	$X_{23}$	$X_{24}$



### 5.1.2. Entrenamiento y evaluación de desempeño

Una vez conformadas las bases de datos para los siete problemas DTLZ [62]. Estas fueron empleadas para entrenar los modelos utilizando el algoritmo GPTIPS de Searson [64]. Los parámetros de configuración utilizados para este algoritmo se describen en la tabla 5.4.

Cuadro 5.4: Parámetros de entrenamiento GPTIPS.

Parámetro	Configuración
Tamaño de población	100
No. de generaciones	200
Probabilidad de cruza	0.95
Probabilidad de mutación	0.05
Selección	Torneo 2 individuos.
No. de genes	4
Nodo terminal	Rasgos estadísticos.
Nodo función	Suma, resta, multiplicación, división, raíz, seno, coseno, potencia, exponencial y logaritmo natural.

La técnica elegida para el entrenamiento tiene como objetivo principal que los modelos aparte de aproximar el valor de hipervolumen puedan generalizar los conjuntos de soluciones de todos los problemas DTLZ. Como cada problema tiene propiedades diferentes la estrategia elegida fue realizar combinaciones de bases de datos. Por consiguiente, se trabajó con combinaciones de dos bases de datos. esta forma de entrenamiento permitió generar un total de veinte y uno modelos. Cada modelo generado fue evaluado sobre las bases de datos de los siete problemas DTLZ.

De cada base de datos se generó su versión normalizada utilizando *MinMax Normalization*. La normalización permitió ajustar los valores de las distintas escalas de los diferentes problemas a una escala común. Este método se utilizó siguiendo la misma estrategia de conseguir que un solo modelo generalice todos los problemas del benchmark DTLZ. El entrenamiento con las bases de datos normalizadas



permitió generar veinte y uno modelos normalizados.

Todos los modelos generados fueron evaluados en desempeño con la finalidad de seleccionar los más eficientes. Las métricas de desempeño utilizadas para evaluar los modelos son *RMSE* (*Root mean square error*) y el coeficiente de correlación lineal de *Pearson*.

### 5.1.3. Selección de modelos 3 funciones objetivo.

La selección de los modelos se realizó en base al desempeño obtenido con las métricas *RMSE* y el coeficiente de correlación de *Pearson*. Un modelo es considerado bueno cuando para la mayoría de los problemas su valor de *RMSE* es cercano a cero y el coeficiente de correlación lineal es cercano a uno. El desempeño obtenido por los modelos entrenados con las bases de datos normalizadas se muestra en la tabla 5.5 y el desempeño obtenido por los modelos entrenados con las bases de datos no normalizadas se muestra en la tabla 5.6.



## CAPÍTULO 5. EXPERIMENTOS Y RESULTADOS OBTENIDOS

**Cuadro 5.5:** Desempeño de los modelos entrenados con bases de datos normalizadas problemas con 3 funciones objetivo.

No. Modelo	Problemas de entrenamiento	Problemas de prueba																				
		DTLZ1			DTLZ2			DTLZ3			DTLZ4			DTLZ5			DTLZ6			DTLZ7		
		RMSE	Pearson		RMSE	Pearson		RMSE	Pearson		RMSE	Pearson		RMSE	Pearson		RMSE	Pearson		RMSE	Pearson	
1	DTLZ1-DTLZ2	0.1694	0.9212	0.0591	0.9334	0.1863	0.906	0.3345	0.7211	0.1154	0.695	0.17	0.8706	0.1022	0.8869							
2	DTLZ1-DTLZ3	0.1546	0.9347	0.2034	0.8185	0.1563	0.9349	0.1711	0.8842	0.1609	0.6048	0.1653	0.9012	0.1931	0.8035							
3	DTLZ1-DTLZ4	0.1509	0.9381	0.0803	0.8958	0.1759	0.925	0.1403	0.9119	0.0877	0.8008	0.1563	0.8819	0.1576	0.8491							
4	DTLZ1-DTLZ5	0.1587	0.9311	0.1605	0.8959	0.1683	0.9237	0.221	0.8674	0.0407	0.9157	0.1053	0.917	0.1554	0.8661							
5	DTLZ1-DTLZ6	0.1542	0.9371	0.1287	0.8854	0.1633	0.9286	0.3646	0.6791	0.0565	0.88	0.088	0.949	0.1265	0.9047							
6	DTLZ1-DTLZ7	0.1624	0.928	0.0596	0.9336	0.1713	0.9225	0.3567	0.6706	0.1887	0.8341	0.1562	0.9254	0.0732	0.9341							
7	DTLZ2-DTLZ3	0.1623	0.9302	0.0523	0.948	0.1674	0.9244	0.2043	0.835	0.0933	0.7172	0.1593	0.8518	0.173	0.8443							
8	DTLZ2-DTLZ4	0.1869	0.9053	0.0566	0.939	0.1831	0.9146	0.1282	0.9279	0.26	0.4343	0.2487	0.8533	0.1597	0.8322							
9	DTLZ2-DTLZ5	0.23	0.9034	0.046	0.9613	0.2065	0.9112	0.2038	0.8089	0.0386	0.9333	0.1053	0.9396	0.1679	0.8723							
10	DTLZ2-DTLZ6	0.2623	0.8716	0.0537	0.9522	0.2311	0.8923	0.3074	0.6062	0.0788	0.7772	0.105	0.9259	0.1467	0.8538							
11	DTLZ2-DTLZ7	0.1936	0.9066	0.151	0.9316	0.1754	0.9256	0.2711	0.7645	0.0232	0.9613	0.0967	0.9316	0.0659	0.9438							
12	DTLZ3-DTLZ4	0.1658	0.9251	0.3091	0.7103	0.1556	0.935	0.1167	0.9415	0.5564	0.0562	0.4415	0.8057	0.4091	0.6275							
13	DTLZ3-DTLZ5	0.1671	0.9241	0.1215	0.9175	0.1532	0.9371	0.235	0.823	0.0266	0.9531	0.0895	0.9397	0.0999	0.8908							
14	DTLZ3-DTLZ6	0.1751	0.9165	0.1594	0.8472	0.1603	0.9311	0.2988	0.6774	0.0496	0.9184	0.0821	0.9494	0.1116	0.9068							
15	DTLZ3-DTLZ7	0.1778	0.9137	0.1267	0.9198	0.1623	0.9297	0.4168	0.6822	0.0737	0.8959	0.1155	0.9181	0.0639	0.9497							
16	DTLZ4-DTLZ5	--	--	0.2596	0.8667	0.3364	0.7475	0.1286	0.9267	0.0281	0.9398	0.1459	0.8797	--	--							
17	DTLZ4-DTLZ6	0.1819	0.9104	0.1196	0.8791	0.1877	0.904	0.1489	0.9011	0.0516	0.869	0.0967	0.9286	0.1239	0.85							
18	DTLZ4-DTLZ7	0.3907	0.6485	0.0556	0.9416	0.2225	0.8912	0.1359	0.9195	0.0946	0.7149	0.1571	0.8671	0.0778	0.9207							
19	DTLZ5-DTLZ6	0.2472	0.845	0.1515	0.9031	0.1935	0.924	0.205	0.8394	0.0252	0.9593	0.0905	0.937	0.0804	0.9178							
20	DTLZ5-DTLZ7	0.5489	0.7554	0.048	0.9564	0.2175	0.886	57.1768	-0.0281	0.2771	0.7694	0.2377	0.772	0.0667	0.9434							
21	DTLZ6-DTLZ7	0.2575	0.8358	0.1397	0.878	0.2042	0.9086	0.2786	0.7655	0.0734	0.9209	0.0864	0.9451	0.076	0.9274							



## CAPÍTULO 5. EXPERIMENTOS Y RESULTADOS OBTENIDOS

**Cuadro 5.6:** Desempeño de los modelos entrenados con bases de datos no normalizadas problemas con 3 funciones objetivo.

No. Modelo	Problemas de prueba																																			
	Problemas de entrenamiento						DTLZ3						DTLZ4						DTLZ5						DTLZ6						DTLZ7					
	DTLZ1		DTLZ2		DTLZ3		DTLZ4		DTLZ5		DTLZ6		DTLZ7		DTLZ1		DTLZ2		DTLZ3		DTLZ4		DTLZ5		DTLZ6		DTLZ7									
RMSE	Pearson	RMSE	Pearson	RMSE	Pearson	RMSE	Pearson	RMSE	Pearson	RMSE	Pearson	RMSE	Pearson	RMSE	Pearson	RMSE	Pearson	RMSE	Pearson	RMSE	Pearson	RMSE	Pearson	RMSE	Pearson	RMSE	Pearson									
1	DTLZ1 - DTLZ2	1.60E+05	0.9981	232.121	0.9336	5.79E+07	0.9883	7.86E+07	0.963	232.0964	0.3971	284.9874	0.8356	231.1447	0.8972																					
2	DTLZ1 - DTLZ3	1.41E+06	0.9621	0.5741	0.8544	5.48E+06	0.9994	5.01E+07	0.9526	0.1856	0.8646	75.0997	0.9131	1.1256	0.8586																					
3	DTLZ1 - DTLZ4	1.20E+06	0.9675	46.232	0.4145	2.00E+07	0.9935	1.07E+07	0.9978	44.0525	0.4946	98.4272	0.9117	47.7784	-0.3656																					
4	DTLZ1 - DTLZ5	3.89E+05	0.9892	29.2063	0.719	5.07E+08	0.9698	6.27E+08	0.9628	19.2697	0.8049	764.5208	0.8942	17.3464	0.3391																					
5	DTLZ1 - DTLZ6	5.72E+05	0.9826	0.4944	0.8579	6.30E+07	0.922	2.92E+09	-0.207	0.3317	0.9249	98.3829	0.8316	1.6295	0.8032																					
6	DTLZ1 - DTLZ7	1.05E+06	0.9209	1.81E+05	0.2023	1.48E+08	0.8696	1.49E+08	0.8357	1.29E+05	0.2855	1.67E+05	-0.1227	1.32E+05	0.2691																					
7	DTLZ2 - DTLZ3	3.61E+05	0.9909	0.3146	0.9456	6.62E+06	0.9992	2.54E+07	0.9909	0.4852	0.9734	87.6794	0.9616	2.0101	0.9138																					
8	DTLZ2 - DTLZ4	1.81E+06	0.9039	1.1475	0.918	3.88E+07	0.9717	2.94E+07	0.9842	0.4058	0.9227	187.8121	0.8859	1.6135	0.8987																					
9	DTLZ2 - DTLZ5	1.96E+08	0.7331	0.2033	0.9668	5.49E+10	0.9823	6.67E+10	0.9141	0.143	0.916	789.5434	0.8849	19.4894	0.8031																					
10	DTLZ2 - DTLZ6	1.89E+08	-0.7822	1.1754	0.893	4.37E+11	-0.727	8.23E+11	-0.5151	2.3654	0.8849	48.0603	0.9609	3.0319	0.4832																					
11	DTLZ2 - DTLZ7	3.02E+16	0.6918	0.2502	0.9491	2.26E+21	0.9041	6.57E+21	0.5968	0.2004	0.9489	3.03E+05	0.732	0.2455	0.9805																					
12	DTLZ3 - DTLZ4	2.28E+05	0.9965	0.2788	0.9709	1.15E+07	0.9974	1.18E+07	0.9975	0.1538	0.9627	59.32	0.9377	1.2362	0.8683																					
13	DTLZ3 - DTLZ5	7.54E+05	0.9658	10.4956	0.1739	2.28E+07	0.9909	5.55E+07	0.966	23.9559	0.0794	261.8288	-0.3364	19.9173	-0.5831																					
14	DTLZ3 - DTLZ6	1.03E+06	0.921	0.925	0.7875	6.44E+06	0.9992	8.53E+07	0.9572	0.2703	0.8041	131.3328	0.8522	1.6149	-0.1951																					
15	DTLZ3 - DTLZ7	1.31E+07	0.4398	1.33E+06	-0.6333	3.22E+07	0.9803	7.48E+07	0.9148	1.42E+06	-0.7916	1.15E+06	-0.8528	1.42E+06	-0.2935																					
16	DTLZ4 - DTLZ5	4.43E+06	0.6418	7.64E+04	0.7087	5.82E+07	0.9333	4.47E+07	0.9633	5.65E+04	0.3379	3.19E+05	0.8341	5.75E+04	0.3217																					
17	DTLZ4 - DTLZ6	3.96E+05	0.9886	0.5511	0.9218	2.17E+07	0.9922	2.45E+07	0.9892	0.4372	0.8947	153.288	0.9506	3.9641	0.9117																					
18	DTLZ4 - DTLZ7	1.14E+06	0.9668	1.2331	0.9396	3.48E+07	0.9766	1.39E+07	0.9962	0.7383	0.7262	66.6276	0.9448	1.4813	0.8629																					
19	DTLZ5 - DTLZ6	1.27E+11	-0.1761	0.4359	0.8855	1.52E+16	-0.6889	3.68E+17	-0.2578	0.582	0.8321	36.1438	0.9768	3.6921	-0.2111																					
20	DTLZ5 - DTLZ7	3.94E+09	0.8465	0.4158	0.9113	5.22E+12	0.9547	7.14E+12	0.8915	0.1024	0.9592	871.1115	0.8909	0.1807	0.9892																					
21	DTLZ6 - DTLZ7	5.29E+07	0.9615	4.6314	0.6293	1.04E+10	0.9836	1.21E+10	0.9659	2.5709	0.326	37.5144	0.9764	3.6333	0.5585																					

Aun cuando varios modelos obtuvieron un desempeño aceptable fueron seleccionados un modelo normalizado y un modelo no normalizado. Los análisis de tiempo y eficiencia para aproximar el valor de hipervolumen se realizaron sobre estos dos modelos. El modelo normalizado fue entrenado con la combinación de las bases de datos DTLZ2 y DTLZ7 y se define como:

$$\begin{aligned} \text{Modelo}_{11} = & \frac{4,62 * 10^{15} * \exp(-\exp(-\exp(-X_9)))}{5,62 * 10^{14}} - \frac{5,49 * 10^{15} * \log(X_{16})^{X_3}}{1,80 * 10^{16}} \\ & - \frac{2,27 * 10^{15} * X_3^3}{1,12 * 10^{15}} - \frac{2,39 * 10^{15} * (X_2 + X_{13} + \exp(-X_3))^3}{3,60 * 10^{16}} - 3,45. \end{aligned}$$

El modelo no normalizado se entrenó con la combinación de las bases de datos DTLZ3 y DTLZ4 y se define como:

$$\begin{aligned} \text{Modelo}_{12} = & \frac{7,75 * 10^{15} * X_{16} * X_{17} * X_{24} * X_2^2 * X_{22}^2}{7,55 * 10^{22}} \\ & - \frac{4,98 * 10^{15} * X_{15} * X_2 * X_{22}}{2,51 * 10^{15}} + \frac{4,80 * 10^{15} * X_{23} * X_{24} * X_{22}}{4,50 * 10^{15}} \\ & - \frac{8,72 * 10^{15} * X_1 * X_4 * X_{15} * X_{16} * X_{17} * X_{24}^2}{3,77 * 10^{22}} - 1,93 * 10^{-9}. \end{aligned}$$

#### 5.1.4. Selección de modelos 4 funciones objetivo.

Los experimentos realizados para problemas con cuatro funciones objetivo son una réplica de los experimentos de tres funciones objetivo con pequeñas pero importantes variaciones en la forma de construir las bases de datos. Primero es necesario incluir los rasgos estadísticos que describen la función objetivo adicional. Esto se consigue variando la configuración del algoritmo SMS-EMOA [17] en un objetivo y modificando la forma en la que se almacenan los conjuntos de soluciones no dominadas. Los conjuntos de soluciones no dominadas de un problema de cuatro funciones objetivo extienden el vector de rasgos en una variable más para cada esta-



## CAPÍTULO 5. EXPERIMENTOS Y RESULTADOS OBTENIDOS

dística. La tabla 5.7 muestra las variables de los diferentes rasgos para un problema de cuatro funciones objetivo.

**Cuadro 5.7:** Rasgos estadísticos que describen un conjunto de puntos no dominados de un problemas de 4 funciones objetivo.

Rasgos	Variables predictoras en el vector de características			
	Objetivo 1	Objetivo 2	Objetivo 3	Objetivo 4
Media	$X_1$	$X_2$	$X_3$	$X_4$
Mediana	$X_5$	$X_6$	$X_7$	$X_8$
Desviación estándar	$X_9$	$X_{10}$	$X_{11}$	$X_{12}$
Percentil 25	$X_{13}$	$X_{14}$	$X_{15}$	$X_{16}$
Percentil 75	$X_{17}$	$X_{18}$	$X_{19}$	$X_{20}$
Curtosis	$X_{21}$	$X_{22}$	$X_{23}$	$X_{24}$
Skewness	$X_{25}$	$X_{26}$	$X_{27}$	$X_{28}$
Reference point	$X_{29}$	$X_{30}$	$X_{31}$	$X_{32}$

Siguiendo con la misma estrategia de entrenamiento para buscar que un solo modelo generalice los siete problemas del conjunto DTLZ [62], pero ahora con cuatro funciones objetivo. Se realizó el mismo entrenamiento basado en combinaciones de dos bases de datos. Combinaciones que generan veinte y uno modelos que son evaluados en desempeño para seleccionar solamente dos de los mejores. El desempeño obtenido por los modelos entrenados con las bases de datos normalizadas para cuatro funciones objetivo se muestra en la tabla 5.8 y el desempeño obtenido por los modelos entrenados con las bases de datos no normalizadas se muestra en la tabla 5.9.

Cuadro 5.8: Desempeño de los modelos entrenados con bases de datos normalizadas problemas con 4 funciones objetivo.

No. Modelo	Problemas de entrenamiento	Problemas de prueba																				
		DTLZ1			DTLZ2			DTLZ3			DTLZ4			DTLZ5			DTLZ6			DTLZ7		
		RMSE	Pearson	RMSE	Pearson	RMSE	Pearson	RMSE	Pearson	RMSE	Pearson	RMSE	Pearson	RMSE	Pearson	RMSE	Pearson	RMSE	Pearson	RMSE	Pearson	
1	DTLZ1 - DTLZ2	0.1389	0.9665	0.041	0.9647	0.1939	0.9205	0.1642	0.8711	0.2235	0.5829	0.1585	0.9009	0.2034	0.8413							
2	DTLZ1 - DTLZ3	0.1514	0.9603	0.0475	0.9545	0.1715	0.9366	0.2591	0.7883	0.5433	0.4121	0.2366	0.8238	0.1242	0.9221							
3	DTLZ1 - DTLZ4	0.1518	0.9599	0.1193	0.9584	0.1962	0.917	0.0873	0.9183	0.1165	0.4658	0.1499	0.872	0.244	0.8433							
4	DTLZ1 - DTLZ5	0.1498	0.9611	0.1847	0.8935	0.1931	0.9217	0.2428	0.8702	0.0391	0.8897	0.095	0.9457	0.1393	0.8856							
5	DTLZ1 - DTLZ6	0.1559	0.9576	0.121	0.923	0.1937	0.9219	0.2918	0.8082	0.183	0.8724	0.0671	0.971	0.122	0.9284							
6	DTLZ1 - DTLZ7	0.1499	0.961	0.1209	0.9355	---	---	0.2013	0.8041	0.0698	0.5802	0.1496	0.8619	0.0956	0.9411							
7	DTLZ2 - DTLZ3	0.2304	0.9356	0.0286	0.9832	0.1671	0.9398	0.3657	0.7279	0.2331	0.5111	0.178	0.8645	0.1043	0.9481							
8	DTLZ2 - DTLZ4	0.2795	0.8767	0.0395	0.9676	0.2202	0.8958	0.0874	0.9184	0.1558	0.4575	0.1499	0.843	0.175	0.8201							
9	DTLZ2 - DTLZ5	0.2739	0.8881	0.0349	0.9752	0.2151	0.9061	0.7694	0.7155	0.0309	0.9405	0.0798	0.9642	0.2041	0.7993							
10	DTLZ2 - DTLZ6	0.2464	0.9403	0.0507	0.9465	0.235	0.8984	0.4519	0.7409	0.1174	0.7549	0.0592	0.977	0.2537	0.753							
11	DTLZ2 - DTLZ7	0.2753	0.9414	0.0408	0.9655	0.2662	0.9128	7.3062	-0.0431	0.1146	0.7323	0.206	0.8216	0.0832	0.9554							
12	DTLZ3 - DTLZ4	0.1923	0.946	0.0749	0.9384	0.1813	0.9287	0.0969	0.9008	0.1027	0.6489	0.1383	0.885	0.1588	0.8794							
13	DTLZ3 - DTLZ5	0.2233	0.9272	0.2331	0.8858	0.175	0.9335	0.2285	0.8219	0.0352	0.9061	0.1175	0.9406	0.214	0.8692							
14	DTLZ3 - DTLZ6	0.2196	0.9332	0.1079	0.9337	0.1632	0.9424	0.3791	0.8181	0.0726	0.9174	0.0634	0.9734	0.1759	0.9275							
15	DTLZ3 - DTLZ7	0.2244	0.9316	0.2568	0.8575	0.1706	0.9369	0.2965	0.7303	0.2061	0.7351	0.0975	0.9363	0.0878	0.9503							
16	DTLZ4 - DTLZ5	0.3405	0.927	0.3942	0.7886	0.2963	0.874	0.0887	0.9161	0.0456	0.8271	0.3206	0.9275	0.1783	0.8979							
17	DTLZ4 - DTLZ6	0.1664	0.9515	0.0792	0.919	0.2049	0.9081	0.0916	0.9102	0.0924	0.7571	0.0854	0.9515	0.1362	0.8977							
18	DTLZ4 - DTLZ7	0.3455	0.8446	0.3953	0.7099	0.2579	0.9014	0.0881	0.9166	0.2235	0.4774	0.173	0.8194	0.0948	0.9419							
19	DTLZ5 - DTLZ6	0.2124	0.9302	0.0658	0.9377	0.2177	0.9163	0.4673	0.8333	0.0332	0.9206	0.0644	0.9736	0.2243	0.8769							
20	DTLZ5 - DTLZ7	0.176	0.9474	0.1707	0.9289	0.1777	0.9341	0.2952	0.8199	0.0355	0.8984	0.1	0.9447	0.0797	0.9592							
21	DTLZ6 - DTLZ7	0.1879	0.9453	0.1191	0.9078	0.2519	0.8804	0.3884	0.8456	0.078	0.7455	0.086	0.951	0.1033	0.9308							

Cuadro 5.9: Desempeño de los modelos entrenados con bases de datos no normalizadas problemas con 4 funciones objetivo.

No. Modelo	Problemas de entrenamiento				Problemas de prueba											
	DTLZ1		DTLZ2		DTLZ3		DTLZ4		DTLZ5		DTLZ6		DTLZ7			
	RMSE	Pearson	RMSE	Pearson	RMSE	Pearson	RMSE	Pearson	RMSE	Pearson	RMSE	Pearson	RMSE	Pearson		
1	9.38E+09	0.9296	26.2726	0.855	6.31E+12	0.9459	0.137	0.6676	25.0136	0.9319	3.75E+04	0.9709	2152.2071	0.8498		
2	3.91E+07	0.9986	0.642	0.9657	1.31E+10	0.9995	0.0389	0.9404	0.7566	0.9772	781.7504	0.993	4.8842	0.9445		
3	7.18E+06	0.9999	0.6508	0.9667	2.16E+11	0.8443	0.0342	0.991	0.9646	0.9788	7.95E+02	0.9946	3.8252	0.4231		
4	4.00E+07	0.9978	1.7512	0.827	4.35E+11	0.5321	0.0268	0.8655	0.5899	0.6777	2.12E+03	0.9027	1.3211	0.7831		
5	2.64E+07	0.9999	105.7588	0.9212	1.27E+12	0.9786	4.4516	0.7096	79.2524	0.9365	19691.9623	0.9859	5.87E+03	-0.7376		
6	1.08E+08	0.9837	8.94E+06	-0.5464	3.32E+11	0.9331	9.15E+06	-0.1017	8.25E+06	-0.6303	3.99E+06	-0.8335	5.07E+06	-0.6725		
7	2.62E+08	0.9893	177.7549	-0.7389	1.57E+10	0.9992	0.0524	-0.8076	84.9332	-0.5131	2.93E+04	-0.909	890.2333	-0.5875		
8	6.16E+08	-0.1919	0.6157	0.8809	4.15E+11	-0.0755	0.1364	-0.1357	1.467	0.5942	2512.0137	0.2647	178.7753	0.1886		
9	---	0	0.2305	0.9842	---	---	4.8932	0.4486	0.1211	0.9661	640898.0176	-0.8019	1.8467	0.8956		
10	3.72E+12	0.0557	13.9885	-0.5113	---	---	25.1205	-0.269	17.9984	-0.7271	113.2798	0.9987	2261.1981	0.7301		
11	---	---	0.2547	0.9807	---	---	240.9761	0.6508	0.5846	0.95	2.44E+03	0.0723	0.2334	0.9918		
12	4.29E+08	0.9757	180.2937	0.8531	4.31E+10	0.9941	270.8439	0.3701	381.9832	0.6109	4.72E+04	0.9508	8277.3589	-0.7079		
13	1.07E+09	0.8898	1.14E+03	0.9625	4.34E+10	0.9941	1.11E+03	0.0141	644.7918	0.9184	229574.2466	0.9814	177.0914	0.944		
14	6.33E+08	0.7579	472.0626	0.7657	5.02E+10	0.992	665.412	0.3784	1431.1655	0.4502	143592.3052	0.8678	187.2164	0.3071		
15	2.52E+07	0.9995	0.3144	0.9792	1.37E+10	0.9994	0.0108	0.942	0.3393	0.9823	2.00E+02	0.9961	3.2829	0.9841		
16	1.09E+14	0.7078	1.2033	0.7126	1.96E+19	0.6561	0.0265	0.5396	0.0632	0.9908	1.87E+05	0.7491	270.2524	0.3423		
17	1.62E+10	-0.6859	3.1615	0.7617	1.87E+14	-0.5311	23.5358	0.3477	1.3851	0.7595	121.174	0.9986	37.9087	0.6152		
18	DTLZ4 - DTLZ7	1.45E+11	0.9357	1.2222	0.9147	1.58E+15	0.8291	0.0897	0.688	0.4182	2.00E+03	0.983	0.2389	0.9915		
19	DTLZ5 - DTLZ6	---	0	8.2837	0.9233	---	---	8.9177	-0.2152	3.6715	0.9385	133.0979	0.9983	4.24E+09	-0.2322	
20	DTLZ5 - DTLZ7	---	---	0.9843	0.7282	---	---	4.2368	0.2556	0.2378	0.8617	---	---	0.5942	0.9459	
21	DTLZ6 - DTLZ7	3.46E+10	0.0113	3.8324	0.9494	7.86E+30	0.006	0.8477	1.6881	0.9447	81.8848	0.9993	7.8104	0.6882		



De los modelos entrenados para aproximar el valor de hipervolumen de conjuntos de soluciones con 4 funciones objetivos fueron seleccionados dos: El modelo entrenado con la combinación de las bases de datos normalizadas DTLZ3 y DTLZ7, el cual se define como:

$$\begin{aligned} \text{Modelo}_{15} = & \frac{3,23 * 10^{15} * X_1 * \log(X_3)}{4,50 * 10^{15}} - \frac{5,56 * 10^{15} * X_{12}^{\exp(-X_2 - X_8)}}{4,50 * 10^{15}} \\ & - \frac{1,70 * 10^{15} * \cos(\text{abs}(X_4) * \log(X_{11}))}{4,50 * 10^{15}} \\ & - \frac{2,35 * \exp(X_3)}{2,81^{14}} + 2,6236. \end{aligned}$$

Y el modelo entrenado con la combinación de bases de datos no normalizadas DTLZ1 y DTLZ2 que define como:

$$\begin{aligned} \text{Modelo}_2 = & \frac{4,50 * 10^{15} * X_{29} * X_{30} * X_{31} * X_{32}}{4,50 * 10^{15}} - 0,67 * X_4 * X_{19}^2 * X_{29} \\ & - 0,0054 * X_2 * X_9 * X_{20}^2 * X_{30} - 0,0041 * X_5 * X_{19}^2 \\ & * X_{26} * X_{31} * (X_2 - X_{28}) + 7,35 * 10^{-13}. \end{aligned}$$

## 5.2. Resultados

### 5.2.1. Modelo como indicador para evaluar la calidad de conjuntos de soluciones no dominadas en problemas con 3 funciones objetivo

Los resultados obtenidos se presentan en cuatro partes, primero los resultados del modelo como indicador. En estos resultados se reporta la medida de tiempo y calidad del algoritmo propuesto que utiliza el modelo para aproximar el valor de hipervolumen contra el algoritmo WFG [12]. Estas pruebas se realizaron sobre quince archivos generados para cinco diferentes tamaños de N. Donde N representa el



## CAPÍTULO 5. EXPERIMENTOS Y RESULTADOS OBTENIDOS

número de individuos o puntos del conjunto no dominado. Estos archivos fueron generados para los siete diferentes problemas del conjunto DTLZ [62]. En la tabla 5.10 se muestran las comparativas de tiempo para los siete problemas DTLZ. Sobre esta tabla se resalta el factor de tiempo, valor que representa cuantas veces es más rápido el modelo respecto al algoritmo WFG [12]. Como podemos observar para  $N=100$  el algoritmo que tiene implementado el modelo es doce veces más rápido, conforme se incrementa el tamaño del archivo el factor se hace más grande. También se realizó un experimento para el cálculo de contribución de hipervolumen, el cual representa la contribución de cada solución al valor total de la maximización del hipervolumen. La comparación de estos resultados puede observarse en la parte inferior de la tabla. El modelo comienza con un factor de siete y se incrementa conforme se hace más grande el archivo.

**Cuadro 5.10:** Resultados de tiempo para los problemas DTLZ con 3 funciones objetivo.

	N=100		N=300		N=600		N=1000		N=2000	
	Time(Seconds)	Factor	Time(Seconds)	Factor	Time(Seconds)	Factor	Time(Seconds)	Factor	Time(Seconds)	Factor
<b>DTLZ1</b>										
GP-Model	2.8733e-05	---	7.4267e-05	---	0.00015876	---	0.00023783	---	0.0004845	---
WFG	0.00036833	<b>12.819</b>	0.00094184	<b>12.682</b>	0.0028569	<b>17.995</b>	0.0066337	<b>27.892</b>	0.024732	<b>51.046</b>
GP-Model Contributions	0.001526753	---	0.00049266	---	0.00102026	---	0.001434307	---	0.003294707	---
WFG Contributions	0.01127962	<b>7.387</b>	0.213613167	<b>433.591</b>	1.491838247	<b>1462.213</b>	6.58627472	<b>4591.957</b>	51.34678867	<b>15584.631</b>
<b>DTLZ2</b>										
GP-Model	3.25E-05	---	7.80E-05	---	0.0001402	---	0.000215667	---	0.000493613	---
WFG	0.000335767	<b>10.335</b>	0.001053067	<b>13.499</b>	0.003132907	<b>22.345</b>	0.00750226	<b>34.786</b>	0.027687053	<b>56.090</b>
GP-Model Contributions	0.000259873	---	0.000531207	---	0.000920867	---	0.001373473	---	0.003761673	---
WFG Contributions	0.011194513	<b>43.076</b>	0.222523867	<b>418.902</b>	1.5502697	<b>1683.489</b>	6.796069967	<b>4948.090</b>	53.35371925	<b>14183.506</b>
<b>DTLZ3</b>										
GP-Model	2.45E-05	---	5.98E-05	---	0.0001061	---	0.000154427	---	---	---
WFG	0.000336707	<b>13.746</b>	0.000990047	<b>16.548</b>	0.00258378	<b>24.352</b>	0.006443353	<b>41.724</b>	---	---
GP-Model Contributions	0.00023144	---	0.000553367	---	0.000979433	---	0.00111208	---	---	---
WFG Contributions	0.01145852	<b>49.509</b>	0.20848218	<b>376.752</b>	1.530013107	<b>1562.141</b>	6.244425213	<b>5615.086</b>	---	---
<b>DTLZ4</b>										
GP-Model	2.61E-05	---	8.85E-05	---	0.000138693	---	0.0002288	---	0.000380907	---
WFG	0.000421017	<b>16.102</b>	0.00115275	<b>13.020</b>	0.003178717	<b>22.919</b>	0.008409225	<b>36.753</b>	0.031381167	<b>82.385</b>
GP-Model Contributions	0.000213527	---	0.00066244	---	0.00094564	---	0.00146346	---	0.002606033	---
WFG Contributions	0.012892783	<b>60.380</b>	0.22280535	<b>336.340</b>	1.634411942	<b>1728.365</b>	7.853736108	<b>5366.553</b>	59.35976748	<b>22777.823</b>
<b>DTLZ5</b>										
GP-Model	3.27E-05	---	7.87E-05	---	0.00014288	---	0.000242167	---	0.000471207	---
WFG	0.000357987	<b>10.949</b>	0.001046933	<b>13.309</b>	0.00271842	<b>19.0258</b>	0.007113607	<b>29.374</b>	0.026915447	<b>57.12025863</b>
GP-Model Contributions	0.000243867	---	0.000537553	---	0.00093142	---	0.0015129	---	0.002741793	---
WFG Contributions	0.01136378	<b>46.598</b>	0.217634207	<b>404.860</b>	1.69172026	<b>1816.280</b>	7.325417333	<b>4841.970</b>	54.60903688	<b>19917.269</b>
<b>DTLZ6</b>										
GP-Model	2.83E-05	---	8.71E-05	---	0.000152027	---	0.00023856	---	0.000471547	---
WFG	0.000437247	<b>13.612</b>	0.000985993	<b>11.764</b>	0.003024747	<b>18.584</b>	0.006897493	<b>30.425</b>	0.026688087	<b>55.646</b>
GP-Model Contributions	0.00025078	---	0.000531613	---	0.000970753	---	0.001381907	---	0.002834347	---
WFG Contributions	0.011896207	<b>47.436</b>	0.215290973	<b>404.976</b>	1.515310793	<b>1560.963</b>	6.832208473	<b>4944.044803</b>	55.48114966	<b>19574.581</b>
<b>DTLZ7</b>										
GP-Model	3.21E-05	---	0.83E-05	---	0.00016276	---	0.0002267	---	0.0004796	---
WFG	0.000414173	<b>14.614</b>	0.001012227	<b>11.618</b>	0.0028545	<b>18.776</b>	0.00689268	<b>28.892</b>	0.025530393	<b>54.141</b>
GP-Model Contributions	0.00022502	---	0.000568093	---	0.000975833	---	0.001465967	---	0.002723667	---
WFG Contributions	0.01237158	<b>54.979</b>	0.234721393	<b>413.173</b>	1.465330993	<b>1501.620</b>	6.633312187	<b>4524.872</b>	51.07862556	<b>18753.625</b>

En los resultados anteriores, observamos que nuestro algoritmo logra una gran ventaja en tiempo, la cual crece conforme se aumenta el número de puntos en los



## CAPÍTULO 5. EXPERIMENTOS Y RESULTADOS OBTENIDOS

---

diferentes archivos. Es importante resaltar que para el cálculo de contribución de hipervolumen se obtiene una mayor aceleración, por lo que nuestra propuesta ayudara a ejecutar el evolutivo SMS-EMOA [17] en un menor tiempo. Sin embargo, es importante conocer la calidad de las soluciones logradas por los modelos, así como la calidad de la aproximación al conjunto de óptimos de Pareto alcanzada por el evolutivo utilizando el modelo como método de selección.

Los resultados de la calidad de las soluciones del modelo como indicador se obtuvieron comparando el valor real de hipervolumen obtenido con el algoritmo WFG [12] contra la aproximación que logran los modelos. Los valores reportados son error absoluto, error relativo y el porcentaje de precisión. Al igual que el experimento de tiempo, los cálculos de error y precisión son el promedio de 15 archivos de diferentes tamaños generados de los siete problemas del benchmark DTLZ [62]. En la tabla 5.11 se muestran los resultados obtenidos para el *Modelo*<sub>11</sub> entrenado con la combinación de las bases de datos DTLZ2 y DTLZ5.



## CAPÍTULO 5. EXPERIMENTOS Y RESULTADOS OBTENIDOS

**Cuadro 5.11:** Resultado de errores del  $Modelo_{11}$  para los problemas DTLZ con 3 funciones objetivo.

	N=100	N=300	N=600	N=1000	N=2000
<b>DTLZ1</b>					
Absolute error	0.01029271	0.012366737	0.012930316	0.012651831	0.01316205
Relative error (Percentage)	0.922703263	1.108810263	1.159437117	1.133991247	1.180112981
Percent of accuracy (Percentage)	99.08573271	98.90334951	98.85385175	98.87872393	98.83365125
<b>DTLZ2</b>					
Absolute error	0.011924981	0.00908963	0.008708726	0.008263439	
Relative error (Percentage)	1.576142673	1.20173109	1.151210076	1.092498708	
Percent of accuracy (Percentage)	98.42385733	98.79826891	98.84878992	98.90750129	
<b>DTLZ3</b>					
Absolute error	0.005602621	0.000920224	0.002907241	0.003528618	0.004260889
Relative error (Percentage)	0.774487035	0.127166761	0.401950665	0.487843271	0.589284655
Percent of accuracy (Percentage)	99.22551296	99.87299475	99.59965851	99.51452509	99.41416757
<b>DTLZ4</b>					
Absolute error	0.013514193	0.010140166	0.009189634	0.010339215	0.010591629
Relative error (Percentage)	1.786360858	1.342178675	1.216736727	1.366515874	1.398950041
Percent of accuracy (Percentage)	98.21363914	98.65782132	98.78326327	98.63348413	98.60104996
<b>DTLZ5</b>					
Absolute error	0.007739569	0.006338318	0.006282665	0.00495506	0.005747659
Relative error (Percentage)	2.709808925	2.218046702	2.196041515	1.739127084	2.01090673
Percent of accuracy (Percentage)	97.29019107	97.7819533	97.80395848	98.26087292	97.98909327
<b>DTLZ6</b>					
Absolute error	0.004622532	0.005169901	0.00630248	0.006053669	0.009299246
Relative error (Percentage)	1.73901002	1.942182662	2.359644713	2.273476463	3.486392562
Percent of accuracy (Percentage)	98.26098998	98.05781734	97.64035529	97.72652354	96.51360744
<b>DTLZ7</b>					
Absolute error	0.005111798	0.004462845	0.005875476	0.006038633	0.008171364
Relative error (Percentage)	0.878164229	0.766846649	1.006773372	1.034164558	1.402188361
Percent of accuracy (Percentage)	99.12183577	99.23315335	98.99322663	98.96583544	98.59781164

El análisis de error también fue realizado sobre el modelo no normalizado el cual está definido como  $Modelo_{12}$ , entrenado con la combinación de bases de datos DTLZ3 y DTLZ4 no normalizadas. De la misma manera que el análisis anterior este se realiza comparando la aproximación que logra el modelo contra el valor real de hipervolumen calculado con el algoritmo WFG [12]. Los valores reportados son error absoluto, error relativo y el porcentaje de precisión. Los resultados se muestran en la tabla 5.12.



## CAPÍTULO 5. EXPERIMENTOS Y RESULTADOS OBTENIDOS

**Cuadro 5.12:** Resultado de errores del *Modelo*<sub>12</sub> para los problemas DTLZ con 3 funciones objetivo.

	N=100	N=300	N=600	N=1000	N=2000
<b>DTLZ1</b>					
Absolute error	5.759187879	5.79536701	5.774016857	5.777411442	5.796796309
Relative error(Percentage)	4.358768796	4.387351847	4.370482425	4.373164228	4.388481376
Percent of accuracy(Percentage)	95.82328457	95.79704651	95.81253021	95.81006836	95.79600994
<b>DTLZ2</b>					
Absolute error	0.3327053627	0.3246753679	0.3173944961	0.3293495077	0.3276079412
Relative error(Percentage)	2.043559263	1.994479291	1.950625374	2.022611754	2.012131595
Percent of accuracy(Percentage)	97.95644074	98.00552071	98.04937463	97.97738825	97.98786841
<b>DTLZ3</b>					
Absolute error	297.8439427	331.9690103	328.7005586	333.3504278	334.0637087
Relative error(Percentage)	2.413048761	2.691160381	2.663958301	2.702661728	2.70860134
Percent of accuracy(Percentage)	97.64380732	97.37936511	97.40516697	97.3684599	97.36282911
<b>DTLZ4</b>					
Absolute error	0.3228853952	0.1940981291	0.3183044892	0.3158398653	0.3184534628
Relative error(Percentage)	1.983521837	1.231147865	1.955280938	1.940405081	1.956044846
Percent of accuracy(Percentage)	98.01647816	98.78382505	98.04471906	98.05959492	98.04395515
<b>DTLZ5</b>					
Absolute error	0.7654888476	0.7669754931	0.7653426201	0.7568065938	0.7644591627
Relative error(Percentage)	10.54978013	10.57243486	10.5475523	10.41765801	10.53409439
Percent of accuracy(Percentage)	90.45698678	90.43845342	90.45880973	90.56522462	90.46982341
<b>DTLZ6</b>					
Absolute error	99.58189366	104.8536051	104.6845684	104.6599813	104.7153024
Relative error(Percentage)	6.877421935	6.880510258	6.868656158	6.866932145	6.870811254
Percent of accuracy(Percentage)	93.56513115	93.56242757	93.57280572	93.57431527	93.57091878
<b>DTLZ7</b>					
Absolute error	0.09714554113	0.09316246746	0.09255305945	0.09697952301	0.09639587105
Relative error(Percentage)	5.639302972	5.410544996	5.377055806	5.619768056	5.587836489
Percent of accuracy(Percentage)	94.36069703	94.589455	94.62294419	94.38023194	94.41216351

El modelo no normalizado está entrenado para realizar las aproximaciones sin importar la variación de escalas de los diferentes problemas del benchmark DTLZ [62]. Si se comparan los resultados del modelo no normalizado con el modelo normalizado, se puede observar que el modelo normalizado tiene una precisión entre 97 % y 99 %, mientras que el modelo no normalizado obtiene una precisión entre 90 % y 98 %.

### 5.2.2. Modelo como método de selección del evolutivo SMS-EMOA para problemas con 3 funciones objetivo

En esta sección se presentan los resultados del algoritmo que utiliza los modelos para aproximar la contribución de hipervolumen. Este algoritmo es utilizado como

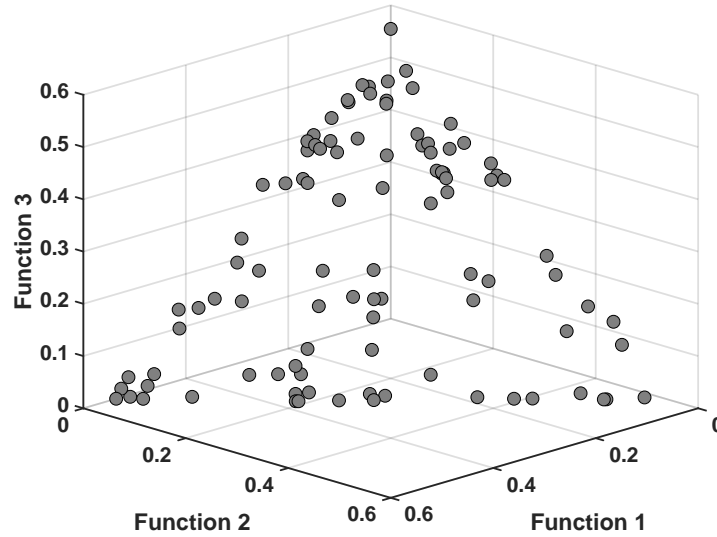


## CAPÍTULO 5. EXPERIMENTOS Y RESULTADOS OBTENIDOS

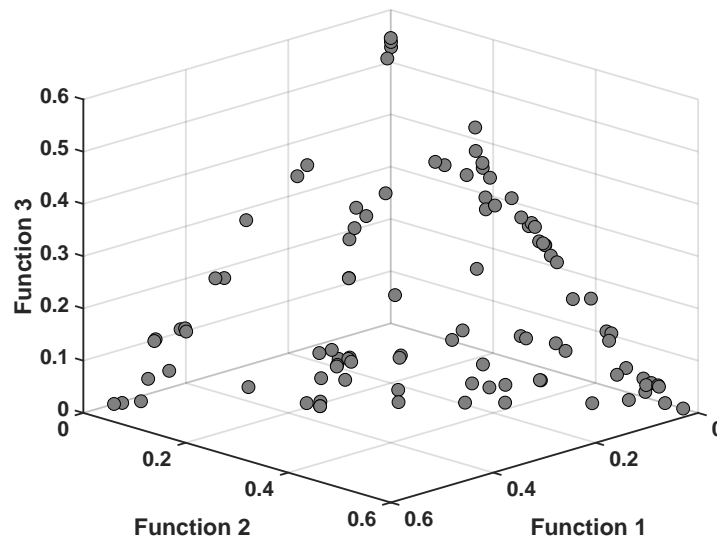
---

método de selección en el algoritmo SMS-EMOA [17]. El análisis realizado se enfoca solamente en el modelo normalizado, modelo que arrojo mejores resultados en el análisis anterior.

En primer lugar, se presenta una comparación de las aproximaciones realizadas al frente de Pareto por el algoritmo basado en indicadores SMS-EMOA. Una de las aproximaciones es realizada utilizando HypE [15] como método de selección. Este algoritmo es utilizado por defecto para el cálculo de contribución de hipervolumen en el algoritmo SMS-EMOA del *framework* PlatEMO [63]. La otra aproximación es realizada utilizando el modelo normalizado que es utilizado para aproximar las contribuciones de hipervolumen en el proceso de selección del algoritmo SMS-EMOA. La comparación de las aproximaciones de los frentes reales para los siete problemas DTLZ [61] se muestran de la figura 5.1 a la figura 5.7.

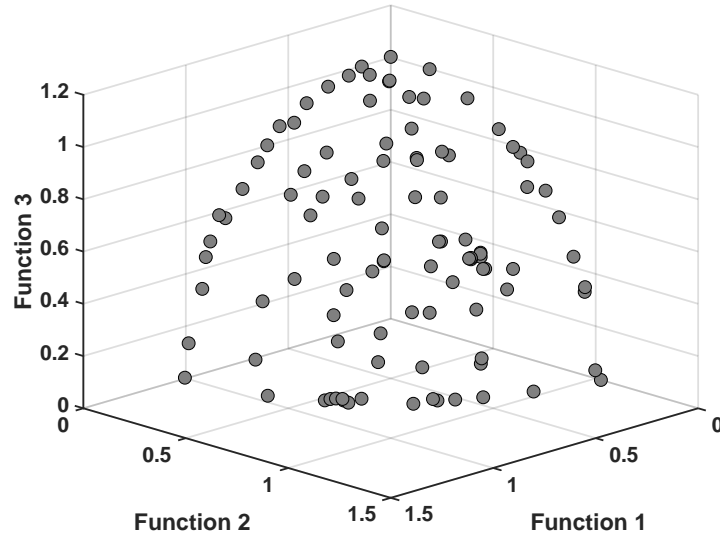


(a) SMS-EMOA with HypE

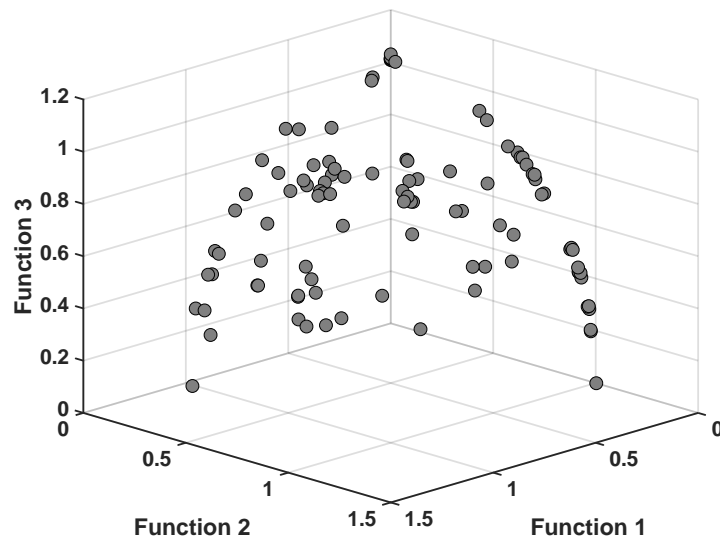


(b) SMS-EMOA with GP-Model

**Figura 5.1:** Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ1 con 3 funciones objetivo.

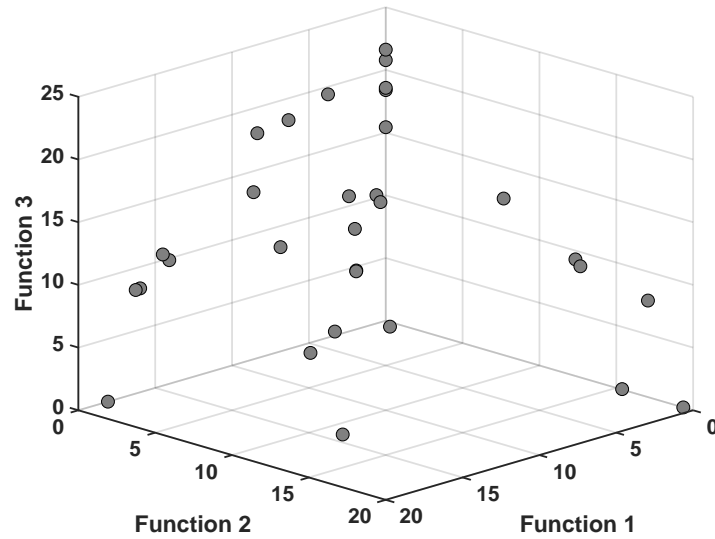


(a) SMS-EMOA with HypE

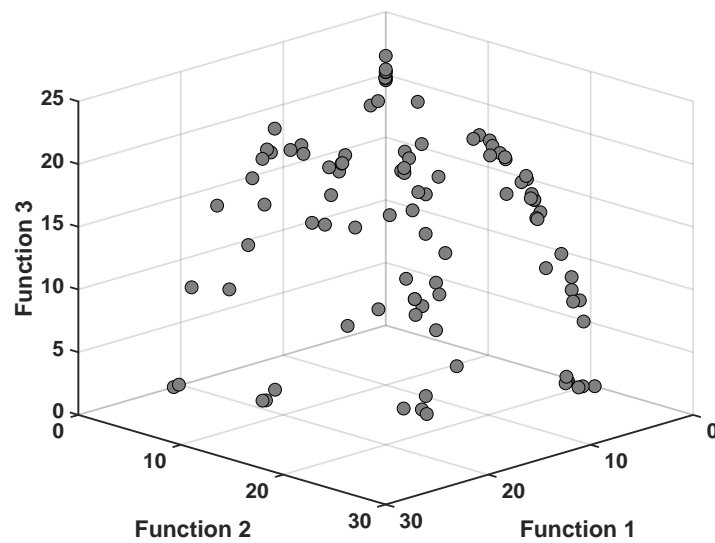


(b) SMS-EMOA with GP-Model

**Figura 5.2:** Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ2 con 3 funciones objetivo.

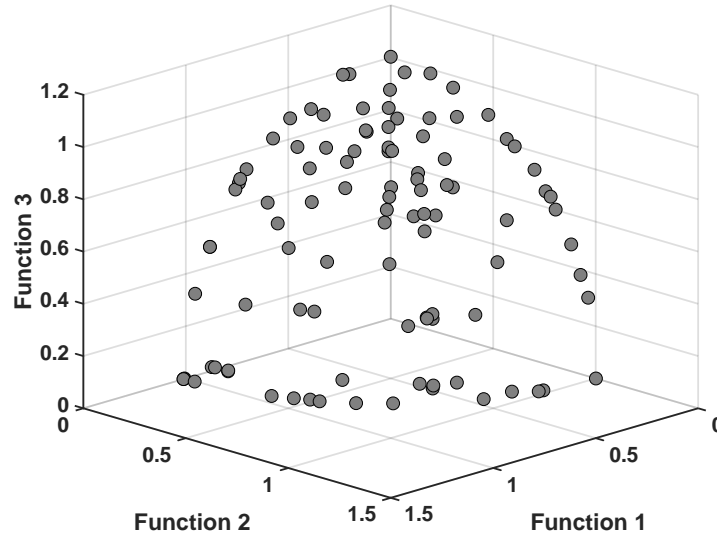


(a) SMS-EMOA with HypE

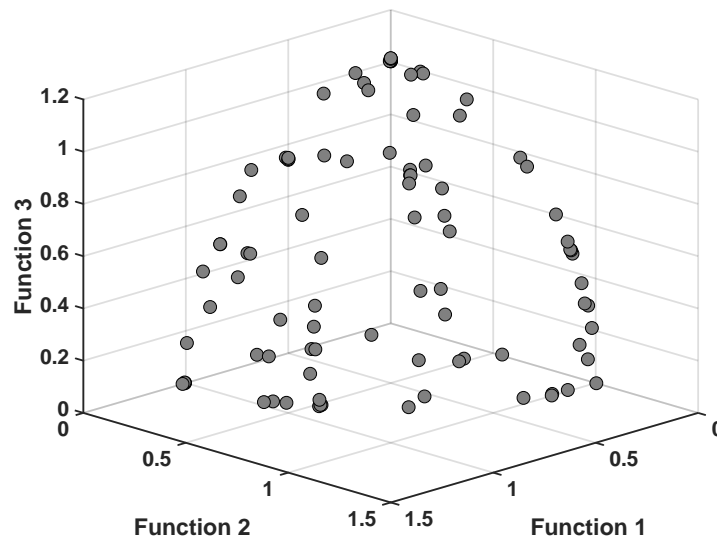


(b) SMS-EMOA with GP-Model

**Figura 5.3:** Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ3 con 3 funciones objetivo.

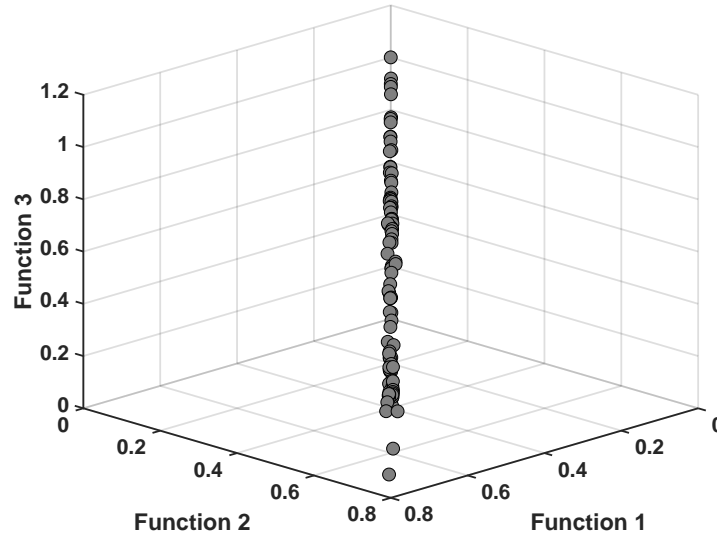


(a) SMS-EMOA with HypE

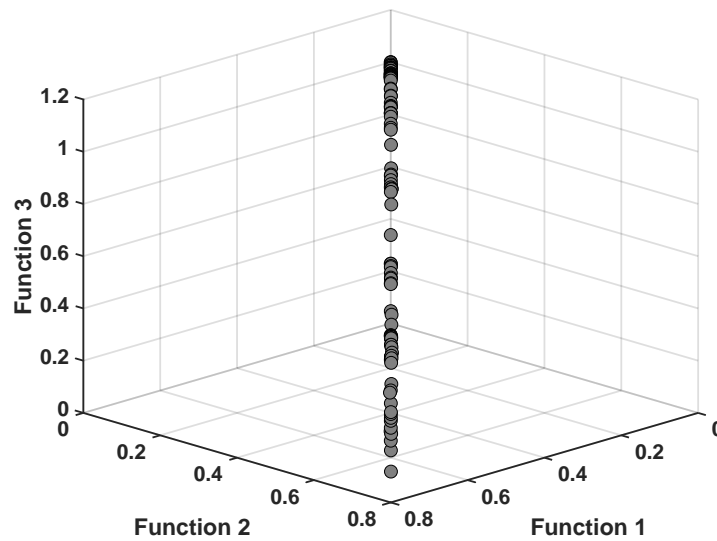


(b) SMS-EMOA with GP-Model

**Figura 5.4:** Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ4 con 3 funciones objetivo.

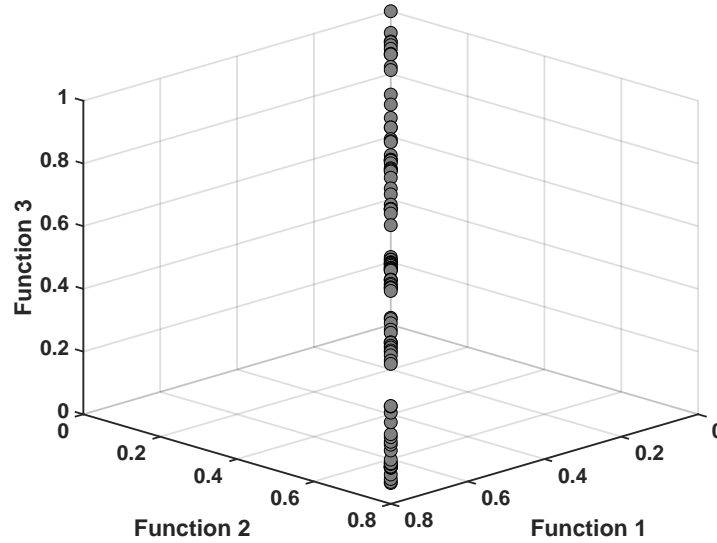


(a) SMS-EMOA with HypE

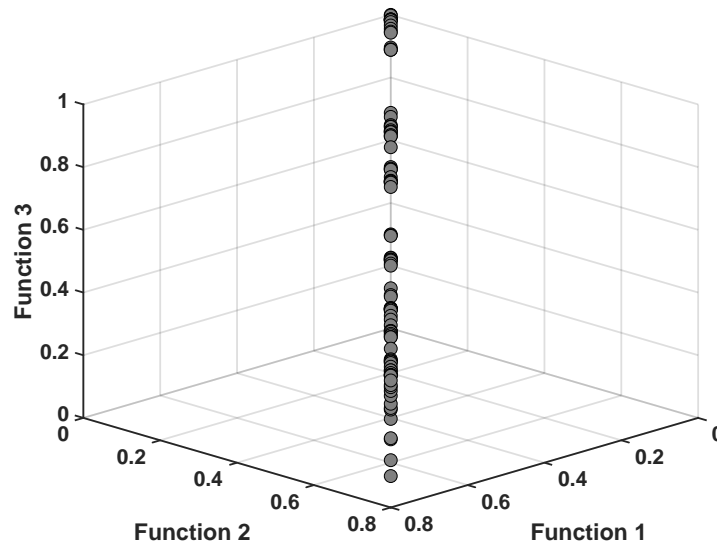


(b) SMS-EMOA with GP-Model

**Figura 5.5:** Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ5 con 3 funciones objetivo.

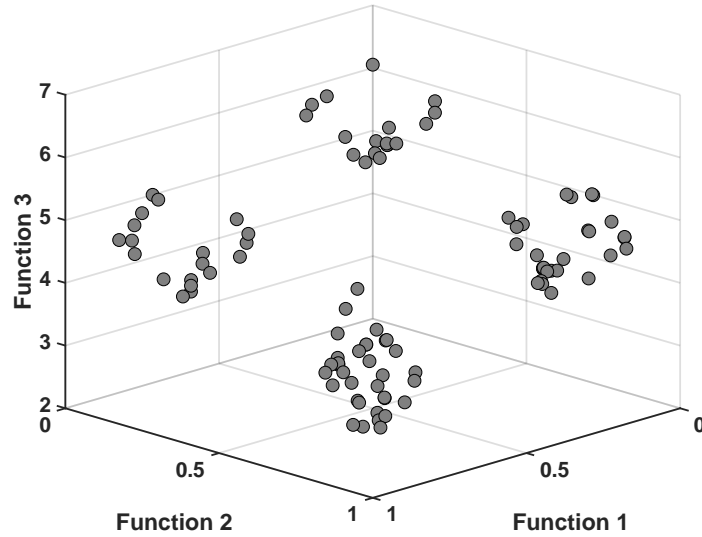


(a) SMS-EMOA with HypE

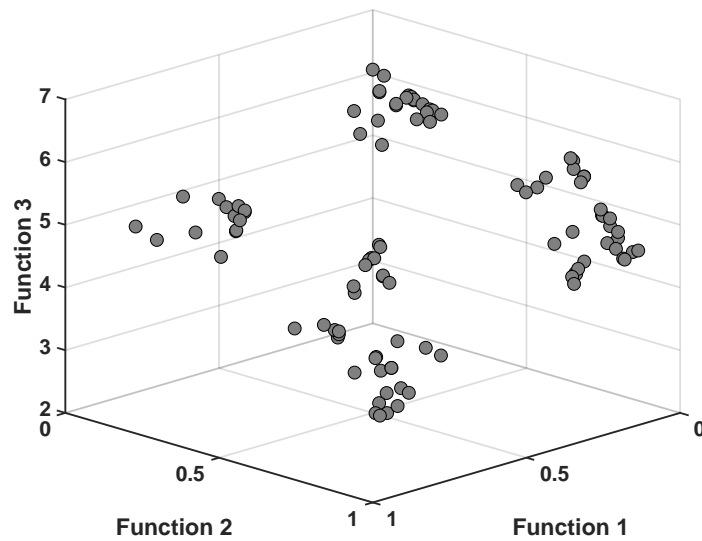


(b) SMS-EMOA with GP-Model

**Figura 5.6:** Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ6 con 3 funciones objetivo.



(a) SMS-EMOA with HypE

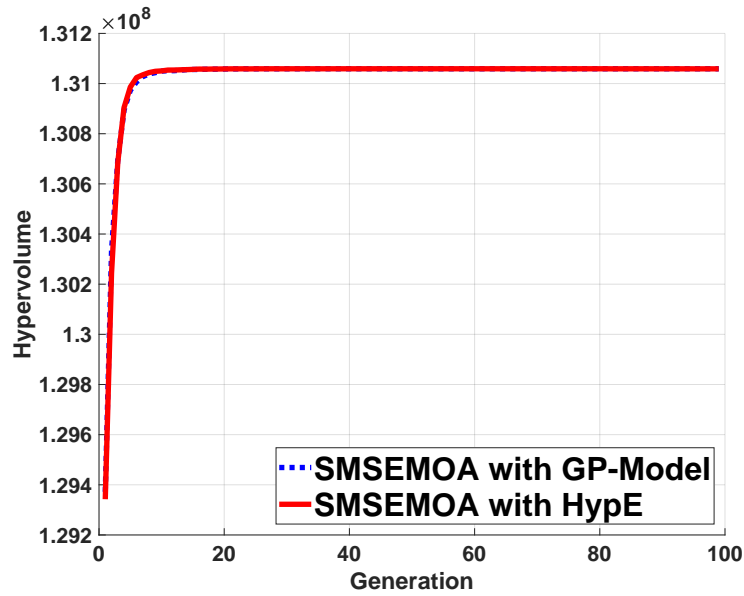


(b) SMS-EMOA with GP-Model

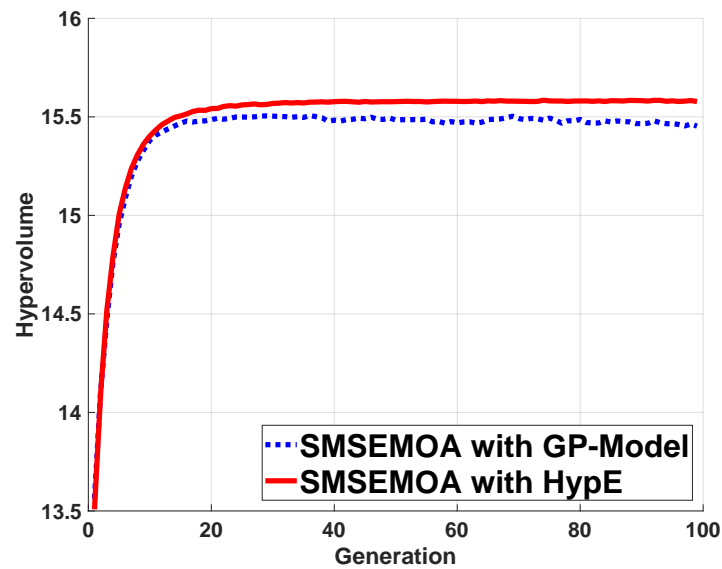
**Figura 5.7:** Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ7 con 3 funciones objetivo.



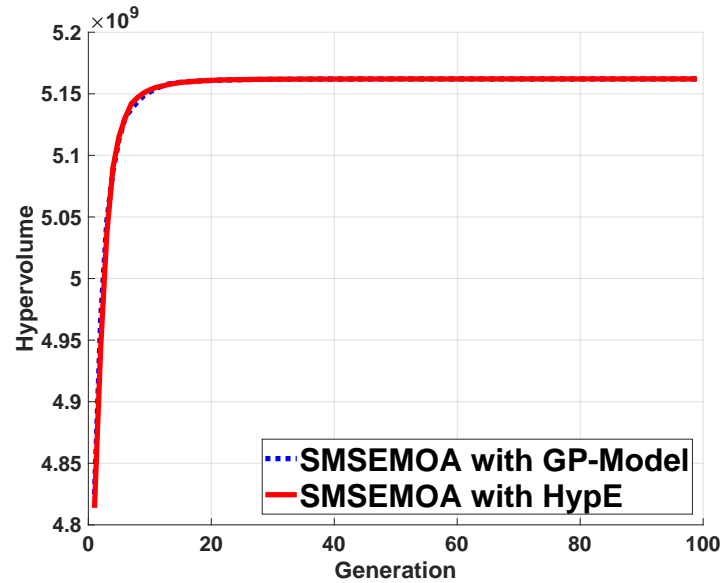
En segundo lugar, se muestra una comparación de las curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA [17]. Este análisis se realizó de la siguiente manera: Se realizaron 30 corridas del algoritmo SMS-EMOA guiado por HypE [15]. Cada corrida se realizó para 100 generaciones y de cada generación se almaceno el mejor conjunto de soluciones no dominadas. De la misma forma se realizaron 30 corridas del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos entrenados con GP utilizando la misma configuración y el mismo número de generaciones. De los conjuntos de soluciones almacenados se calculó el valor real de hipervolumen utilizando el indicador WFG [12]. Los valores de hipervolumen fueron promediados por generación obteniendo una curva promedio de hipervolumen de 30 corridas diferentes. En estas curvas se puede observar la evolución de algoritmo SMS-EMOA guiado por los diferentes algoritmos en busca del frente real. Ambas curvas son sobrepuestas en una gráfica para cada problema DTLZ [61]. Estas graficas se muestran de la figura 5.8 a la figura 5.14.



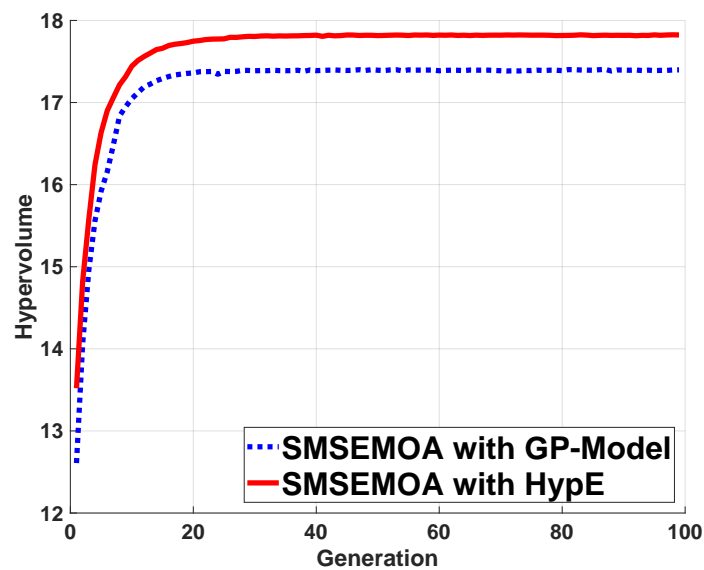
**Figura 5.8:** Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ1 con 3 funciones objetivo.



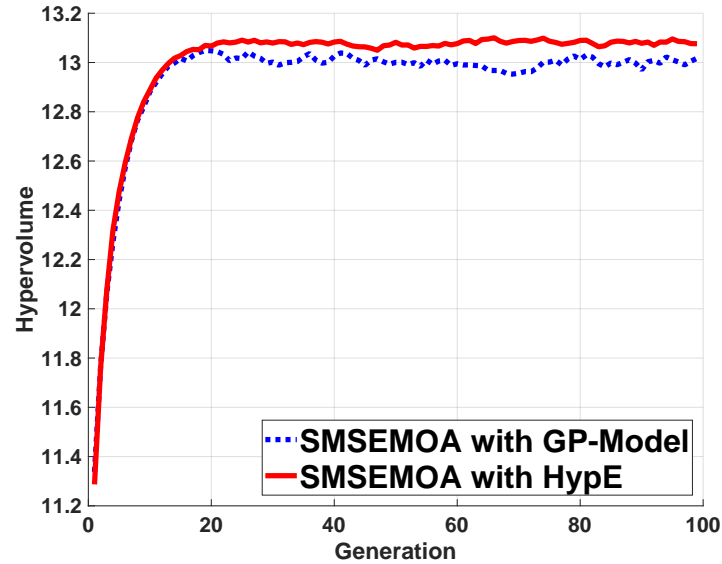
**Figura 5.9:** Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ2 con 3 funciones objetivo.



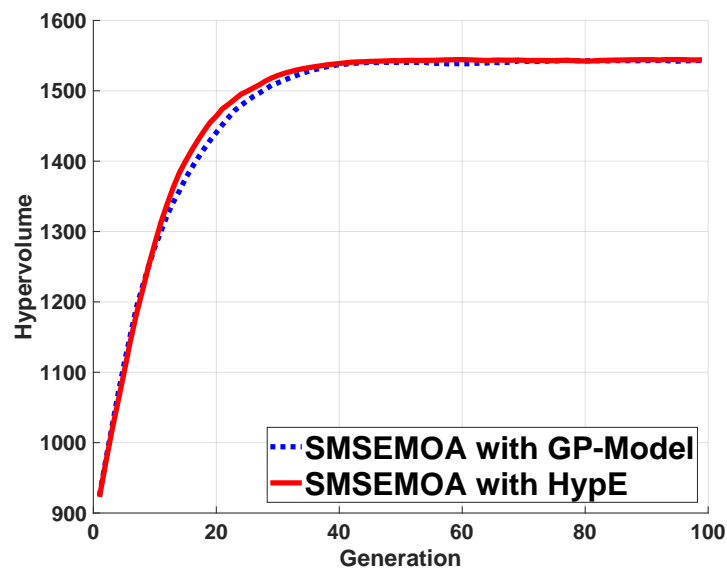
**Figura 5.10:** Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ3 con 3 funciones objetivo.



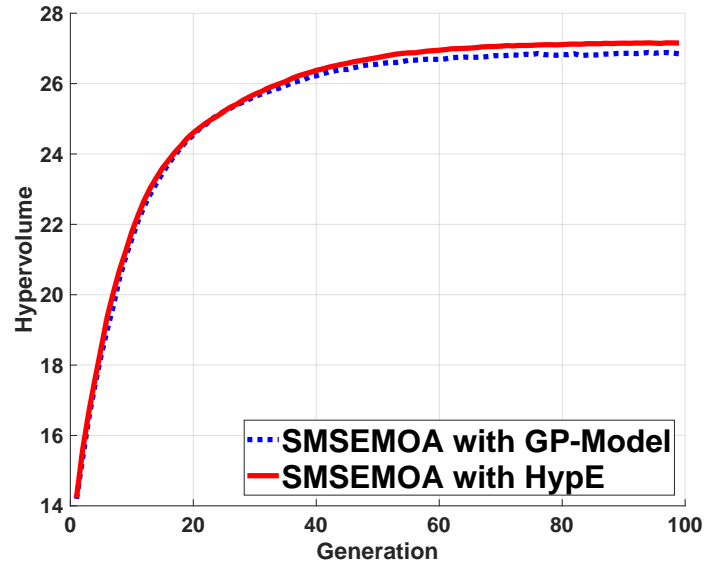
**Figura 5.11:** Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ4 con 3 funciones objetivo.



**Figura 5.12:** Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ5 con 3 funciones objetivo.



**Figura 5.13:** Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ6 con 3 funciones objetivo.



**Figura 5.14:** Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ7 con 3 funciones objetivo.



### 5.2.3. Modelo como indicador para evaluar la calidad de conjuntos de soluciones no dominadas en problemas con 4 funciones objetivo

Los resultados de cuatro funciones se estructuran de la siguiente manera: Primero se muestra la comparación de tiempo de los modelos para un promedio de 15 archivos que contienen conjuntos de soluciones no dominadas de cuatro objetivos. Los resultados de tiempo para seis problemas DTLZ se muestran en la tabla 5.13.

**Cuadro 5.13:** Resultados de tiempo para los problemas DTLZ con 4 funciones objetivo.

	N = 100		N = 300		N = 600		N = 1000		N = 2000	
	Time(Seconds)	Factor	Time(Seconds)	Factor	Time(Seconds)	Factor	Time(Seconds)	Factor	Time(Seconds)	Factor
<b>DTLZ1</b>										
GP-Model	3.34E-05	---	9.15E-05	---	0.0002029	---	0.00044952	---	0.00080467	---
WFG	0.00084603	25.294997	0.00187241	20.473975	0.00431830	21.282930	0.01152374	25.635275	0.03627803	45.084174
GP-Model Contributions	0.00031072	---	0.00084234	---	0.00168651	---	0.00380318	---	0.00555379	---
WFG Contributions	0.04136814	133.13641	0.490415	582.20552	2.4468147	1450.8125	11.651780	3063.6887	74.220927	13364.005
<b>DTLZ2</b>										
GP-Model	3.59E-05	---	1.24E-04	---	0.00021448	---	0.00035043	---	0.00070048	---
WFG	0.00115048	32.046889	0.00273675	21.999598	0.00646586	30.14624	0.01270445	36.253543	0.03841521	54.841014
GP-Model Contributions	0.0003255	---	0.00123146	---	0.00183951	---	0.00271666	---	0.00516306	---
WFG Contributions	0.04827471	148.3094	0.58581531	475.70537	3.50532455	1905.5682	12.222960	4499.2491	78.06318	15119.538
<b>DTLZ4</b>										
GP-Model	2.60E-05	---	9.03E-05	---	0.00019136	---	0.00028596	---	0.00070606	---
WFG	0.00053382	20.521271	0.00204651	22.658473	0.00523105	27.335237	0.0146408	51.197575	0.04450455	63.031658
GP-Model Contributions	0.00021858	---	0.00082008	---	0.00162126	---	0.00232500	---	0.00605844	---
WFG Contributions	0.02257478	103.27613	0.51400782	626.777663	2.9933821	1846.3231	14.951233	6430.619424	93.613618	15451.752568
<b>DTLZ5</b>										
GP-Model	2.12E-05	---	9.43E-05	---	0.00025162	---	0.00035396	---	0.00073157	---
WFG	0.00047932	22.566854	0.00161646	17.138040	0.00422840	16.804732	0.01186291	33.51421	0.03513910	48.032241
GP-Model Contributions	0.00018355	---	0.00070270	---	0.00172725	---	0.00228330	---	0.00438774	---
WFG Contributions	0.01821019	99.209276	0.44592036	634.57539	2.5776836	1492.3599	12.578093	5508.7182	78.353431	17857.32804
<b>DTLZ6</b>										
GP-Model	3.83E-05	---	1.37E-04	---	0.00031481	---	0.00050875	---	0.00092261	---
WFG	0.00104981	27.386521	0.00215515	15.685225	0.00624355	19.832336	0.00999355	19.6433415233415	0.03028561	32.825785
GP-Model Contributions	0.00029796	---	0.00100418	---	0.00227653	---	0.00326718	---	0.00630351	---
WFG Contributions	0.03944133	132.36827	0.45661546	454.71324	3.040452	1335.5624	10.659024	3262.450571	64.089591	10167.275676
<b>DTLZ7</b>										
GP-Model	3.04E-05	---	9.43E-05	---	0.0001919	---	0.00034088	---	0.00069716	---
WFG	0.00052224	17.164110	0.00195923	20.766322	0.00738557	38.486572	0.01841277	54.015411	0.06432178	92.261697
GP-Model Contributions	0.00028163	---	0.00078479	---	0.00157858	---	0.00262788	---	0.00511725	---
WFG Contributions	0.02513036	89.230796	0.52184942	664.951401	4.45736618	2823.643626	18.795712	7152.424161	138.384756	27042.780151

En segundo lugar, se reporta error absoluto, error relativo y el porcentaje de precisión de los modelos entrenado para aproximar el valor de hipervolumen de conjuntos de soluciones no dominadas con cuatro funciones objetivo.

El *Modelo*<sub>15</sub> fue entrenado con la combinación de bases de datos normalizadas DTLZ3 y DTLZ7. Los resultados reportados para este modelo son promedios de 15 archivos diferentes para cada problema DTLZ. En la tabla 5.14 se pueden observar los resultados obtenidos para seis de los siete problemas.

El *Modelo*<sub>2</sub> fue entrenado con la combinación de las bases de datos no normalizadas DTLZ1 y DTLZ2. En la tabla 5.15 se pueden observar los resultados obtenidos



## CAPÍTULO 5. EXPERIMENTOS Y RESULTADOS OBTENIDOS

por este modelo para seis problemas DTLZ.

**Cuadro 5.14:** Resultado de errores del *Modelo*<sub>15</sub> para los problemas DTLZ con 4 funciones objetivo.

	N = 100	N = 300	N = 600	N = 1000	N = 2000
<b>DTLZ1</b>					
Absolute error	0.041712753451457	0.028101511228543	0.046130216396838	0.045671054421808	0.052990862298191
Relative error (Percentage)	3.15793747822468	2.0206734921935	3.24964141454573	3.21729576593217	4.01176657517091
Percent of accuracy (Percentage)	96.9387353456041	97.9793265078065	96.7503585854543	96.7827042340678	96.1429685243625
<b>DTLZ2</b>					
Absolute error	0.027091682909287	0.011097910968834	0.01057262815122	0.028723719515219	0.013822602470837
Relative error (Percentage)	2.59183799733768	1.06172759501163	1.01147423974385	2.30934167465435	1.32239648699445
Percent of accuracy (Percentage)	97.4081620026623	98.9382724049884	98.9885257602562	97.6906583253457	98.6776035130056
<b>DTLZ4</b>					
ine	N=100	N=300	N=600	N=1000	N=2000
Absolute error	0.03085262415733	0.034260009165021	0.053830917123575	0.044361681673143	0.035290647049298
Relative error (Percentage)	2.72097046740897	3.02813873041866	4.84629292777124	3.99379604963922	3.17714841860986
Percent of accuracy (Percentage)	97.279029532591	96.9718612695814	95.1537070722288	96.0062039503608	96.8228515813901
<b>DTLZ5</b>					
Absolute error	0.027091682909287	0.011097910968834	0.01057262815122	0.028723719515219	0.013822602470837
Relative error (Percentage)	2.59183799733768	1.06172759501163	1.01147423974385	2.30934167465435	1.32239648699445
Percent of accuracy (Percentage)	97.4081620026623	98.9382724049884	98.9885257602562	97.6906583253457	98.6776035130056
<b>DTLZ6</b>					
Absolute error	0.052231098086083	0.051600404101185	0.069913427642806	0.038817100563414	0.063245232852154
Relative error (Percentage)	4.42912959291445	4.51398372031629	5.81668607517898	3.0788533247275	5.60383378430328
Percent of accuracy (Percentage)	95.5708704070856	95.4860162796837	94.183313924821	96.9211466752725	94.3961662156967
<b>DTLZ7</b>					
Absolute error	0.006687033895631	0.000780161220062	0.01274539188677	0.013376121543479	0.009629086779271
Relative error (Percentage)	0.962463543123507	0.124288625917542	1.84986118351556	1.94140503870676	1.38591266670217
Percent of accuracy (Percentage)	99.0467115110435	99.8757113740825	98.183737157793	98.0955677058114	98.6330323116405

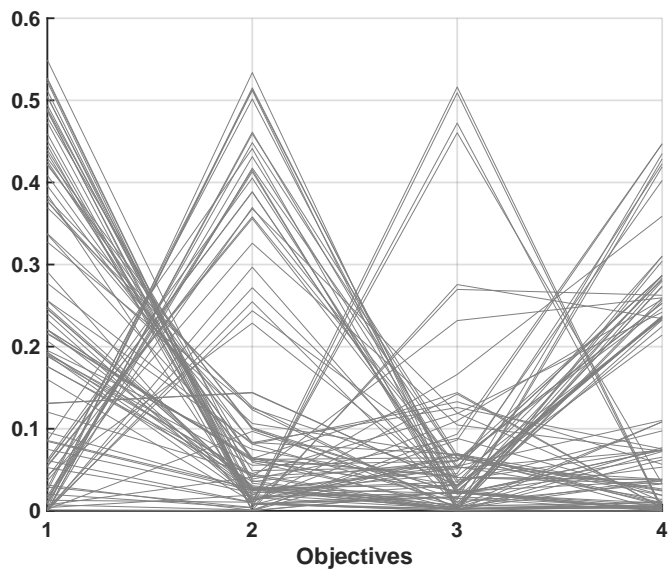
**Cuadro 5.15:** Resultado de errores del *Modelo*<sub>2</sub> para los problemas DTLZ con 4 funciones objetivo.

	N = 100	N = 300	N = 600	N = 1000	N = 2000
<b>DTLZ1</b>					
Absolute error	0.888104591733736	0.749761809493521	0.736419418373316	0.726167786935321	0.677829097590575
Relative error (Percentage)	0.067999378618805	0.057406906384913	0.056385321411805	0.055600386197979	0.055606259412383
Percent of accuracy (Percentage)	99.9320006213812	99.9425930936151	99.9436146785882	99.944399613802	99.9443937405876
<b>DTLZ2</b>					
Absolute error	0.272555089045518	0.165615045455679	0.219029197975015	0.201309259152641	0.258515678314714
Relative error (Percentage)	1.97364653093713	1.19926419675109	1.58605079901897	1.45773583741737	1.87198328780236
Percent of accuracy (Percentage)	98.0645523641852	98.8149477110629	98.43871202144	98.5632087830608	98.162415977989
<b>DTLZ4</b>					
Absolute error	0.280119924842973	0.34754216388964	0.230650628986949	0.268449603707606	0.23235041534653
Relative error (Percentage)	9.21508050336271	11.6923994509112	7.58769343329452	8.8115683425714	7.56295424679866
Percent of accuracy (Percentage)	91.5624468151365	89.5316068878528	92.9474336783705	91.9019930722532	92.9688113349455
<b>DTLZ5</b>					
Absolute error	1.82482105963885	1.86616285247219	1.80573595431384	1.72464059204764	1.62124023787827
Relative error (Percentage)	35.1946988423108	36.2813312231799	34.8266109518628	33.1544388355359	31.1293870325281
Percent of accuracy (Percentage)	73.9673972842963	73.3776219401878	74.1693344466717	75.1007633500779	76.2605562818607
<b>DTLZ6</b>					
Absolute error	519.476517624771	202.861349861771	257.010639107486	479.973317236227	479.973317236227
Relative error (Percentage)	3.41991557888329	1.33551501792173	1.6920008100488	3.15985067538727	3.15985067538727
Percent of accuracy (Percentage)	96.693175042988	98.6820859225065	98.3361515197156	96.9369375249191	96.9369375249191
<b>DTLZ7</b>					
Absolute error	0.256977310059559	0.290735295424533	0.324946526786737	0.375814103268591	0.293768003495303
Relative error (Percentage)	5.46333008572947	6.18102386591313	6.90835364272363	7.98979682896077	6.24549915067829
Percent of accuracy (Percentage)	94.8196874863629	94.17878671643	93.5380600230637	92.6013409937094	94.1216341392299

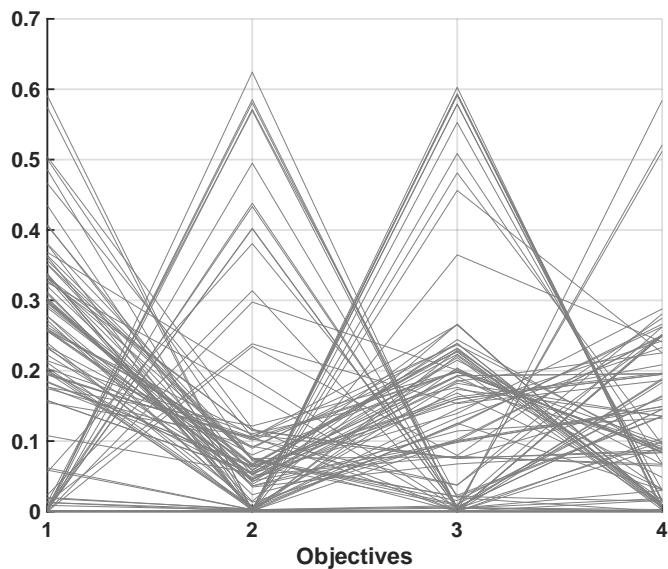


#### **5.2.4. Modelo como método de selección del evolutivo SMS-EMOA para problemas con 4 funciones objetivo**

En este apartado se presentan los resultados de las aproximaciones al frente de Pareto de los problemas DTLZ [62] con cuatro funciones objetivo. Realizadas con las dos variaciones del algoritmo SMS-EMOA [17]. La primera utilizando HypE [15] y la segunda utilizando los modelos entrenado con GP. Los resultados se muestran de la figura 5.15 a la figura 5.21.

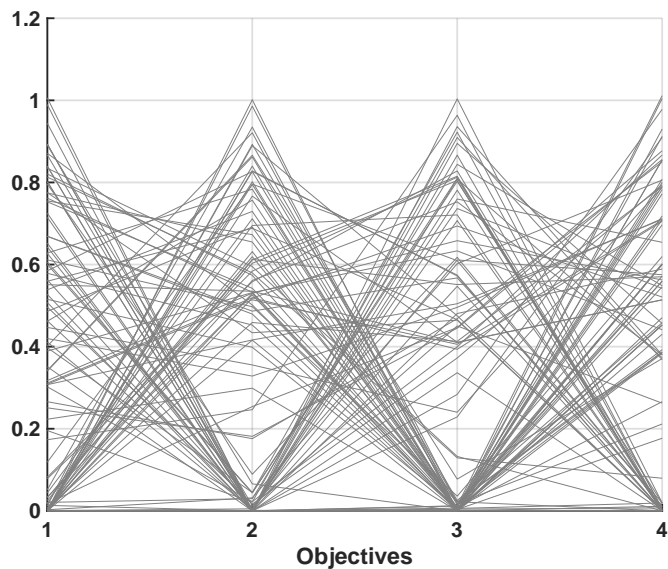


(a) SMS-EMOA with HypE

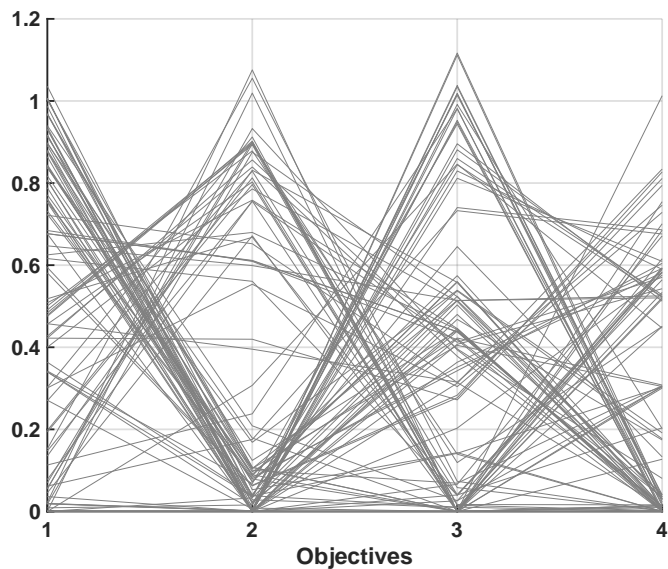


(b) SMS-EMOA with GP-Model

**Figura 5.15:** Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ1 con 4 funciones objetivo.

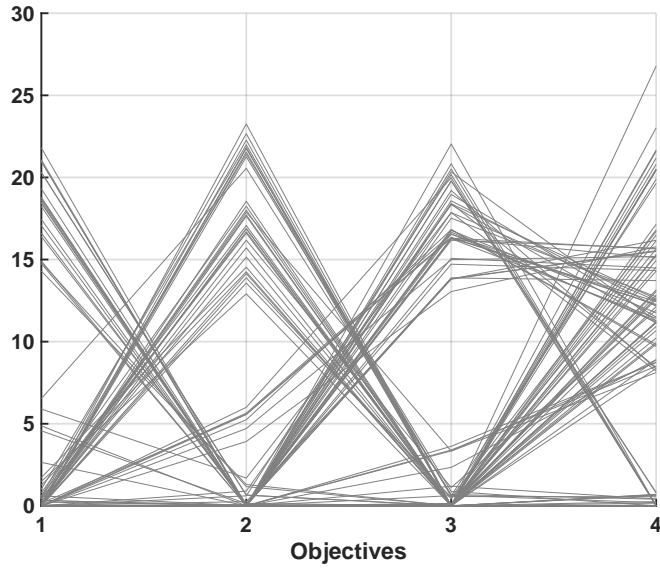


(a) SMS-EMOA with HypE

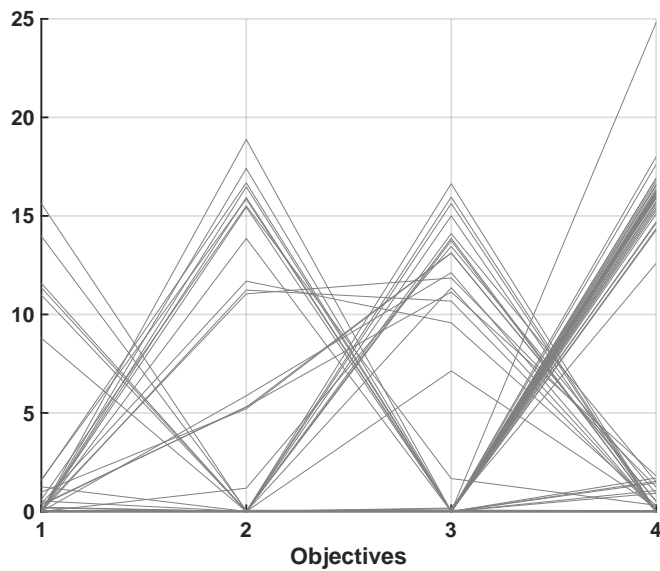


(b) SMS-EMOA with GP-Model

**Figura 5.16:** Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ2 con 4 funciones objetivo.

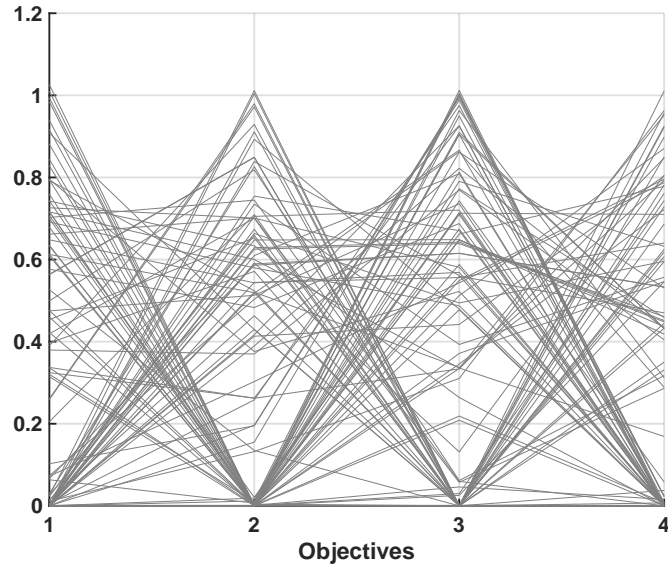


(a) SMS-EMOA with HypE

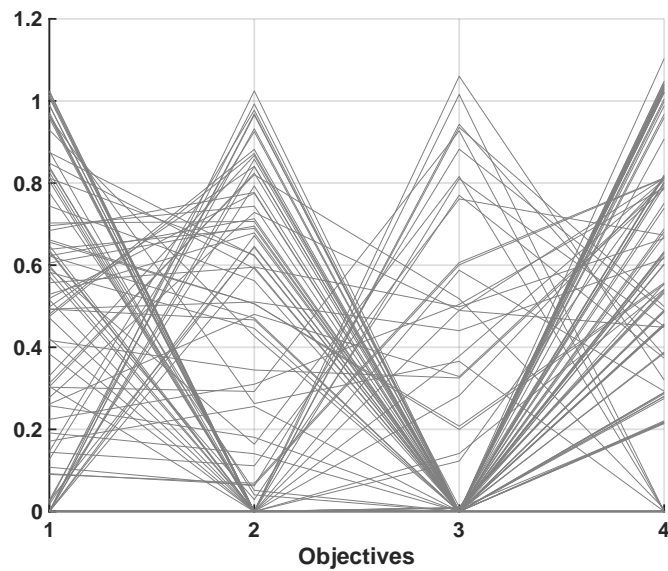


(b) SMS-EMOA with GP-Model

**Figura 5.17:** Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ3 con 4 funciones objetivo.

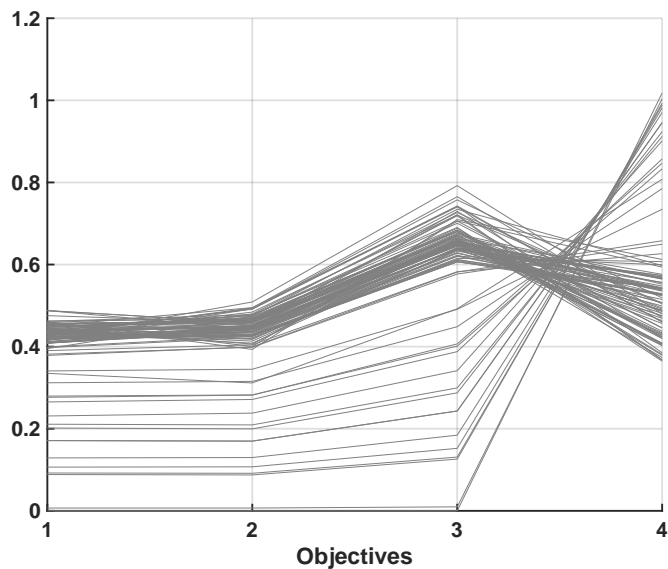


(a) SMS-EMOA with HypE

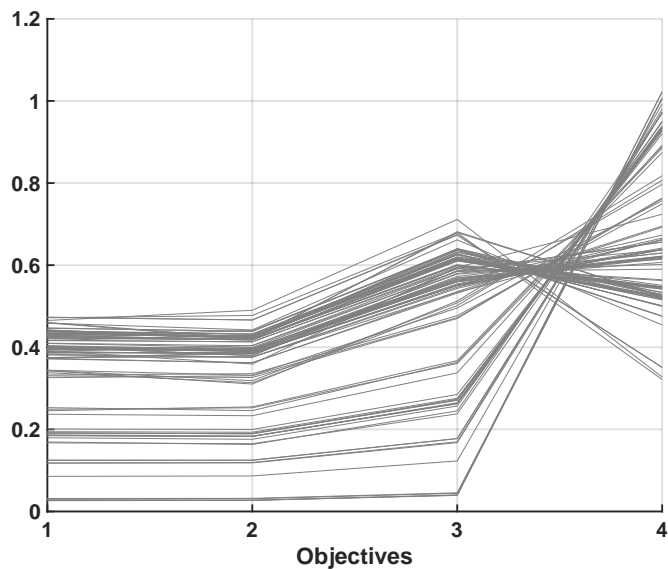


(b) SMS-EMOA with GP-Model

**Figura 5.18:** Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ4 con 4 funciones objetivo.

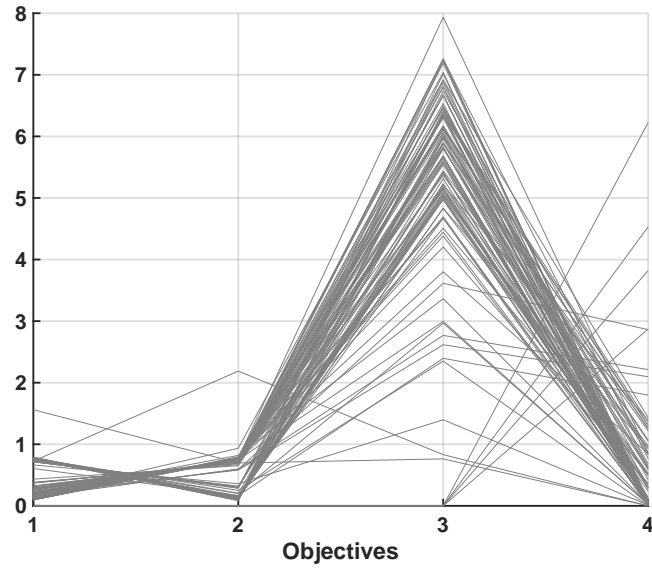


(a) SMS-EMOA with HypE

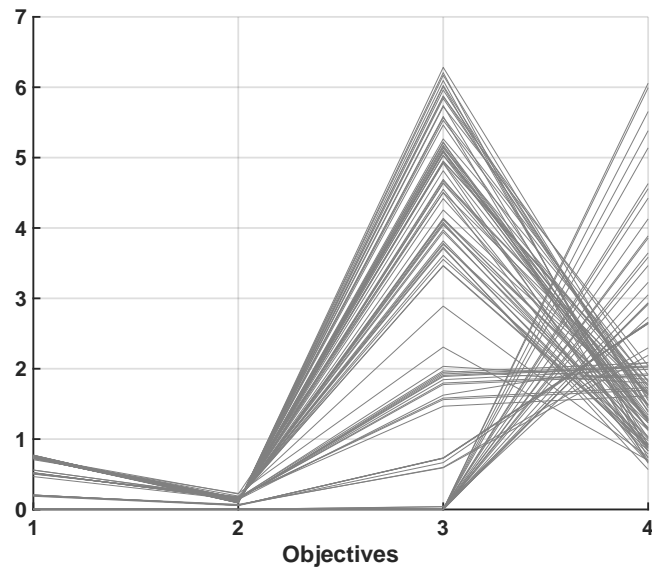


(b) SMS-EMOA with GP-Model

**Figura 5.19:** Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ4 con 3 funciones objetivo.

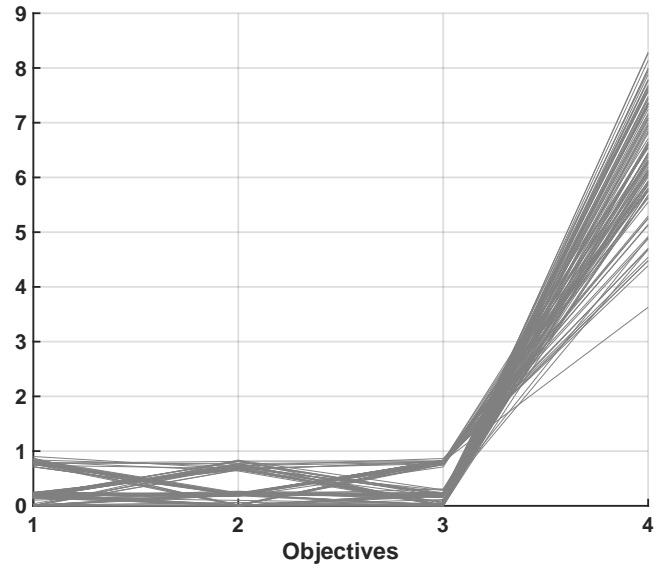


(a) SMS-EMOA with HypE

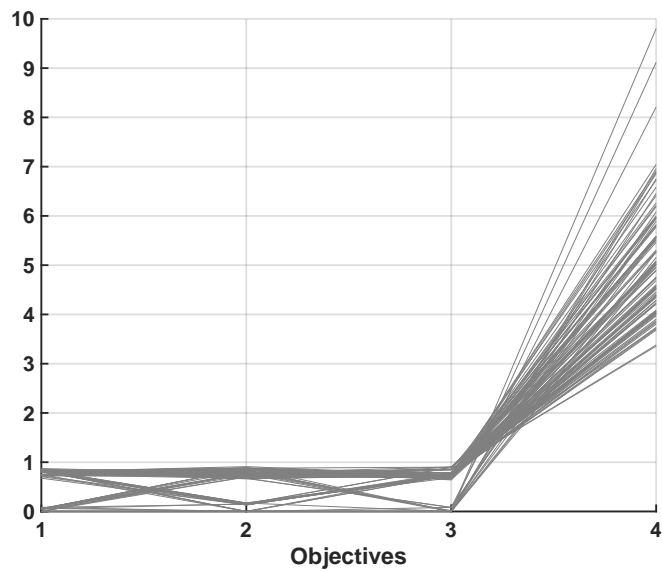


(b) SMS-EMOA with GP-Model

**Figura 5.20:** Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ6 con 4 funciones objetivo.



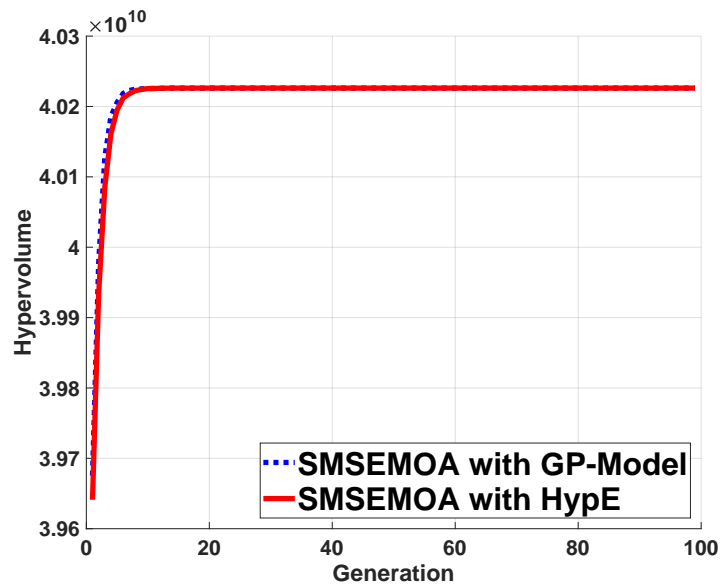
(a) SMS-EMOA with HypE



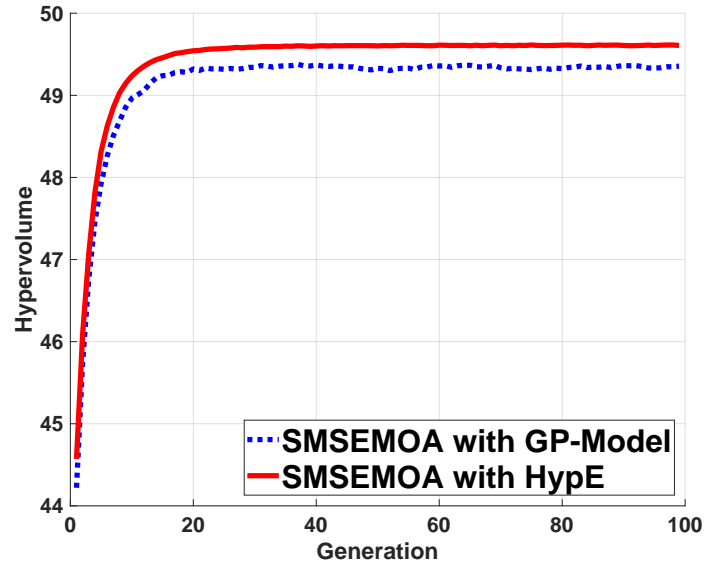
(b) SMS-EMOA with GP-Model

**Figura 5.21:** Aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por HypE(a) y aproximaciones al frente de Pareto del algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP(b) para el problema DTLZ7 con 4 funciones objetivo.

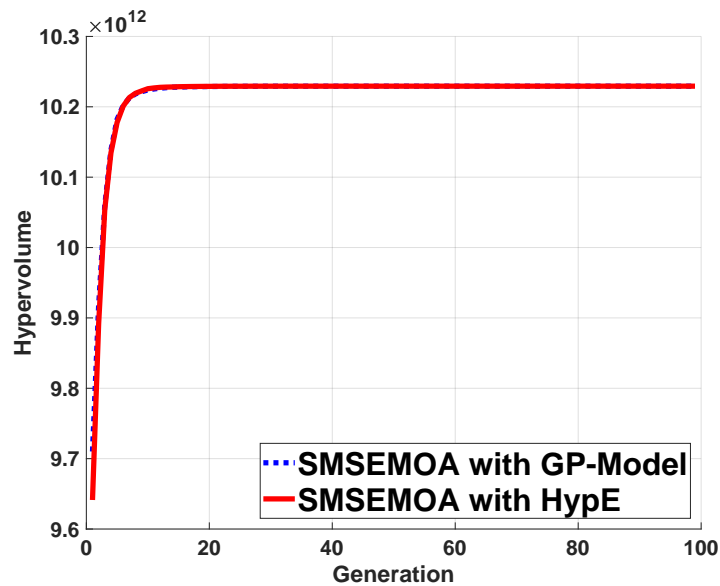
Finalmente se muestra la comparación de las curvas basadas en valor de hipervolumen del algoritmo SMS-EMOA [17] con los métodos HypE [15] y el modelo GP en la búsqueda del frente real para los problemas escalables DTLZ [62] con cuatro funciones objetivo. Los resultados se muestran de la figura 5.22 a la figura 5.28.



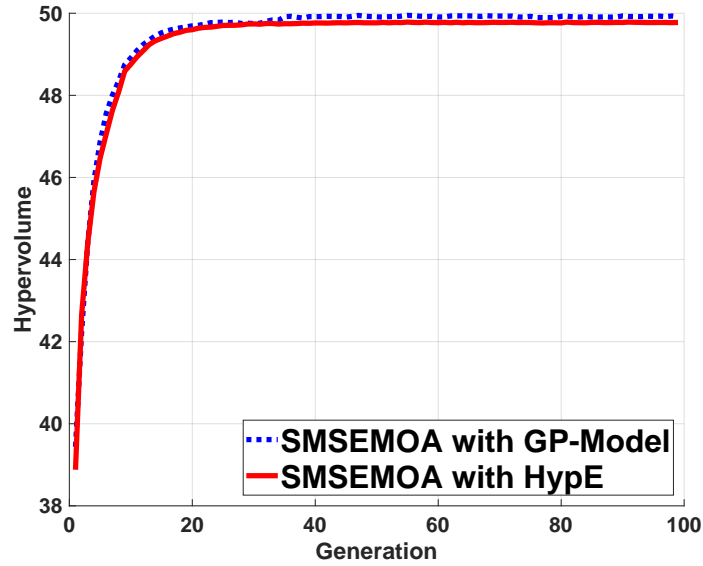
**Figura 5.22:** Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ1 con 4 funciones objetivo.



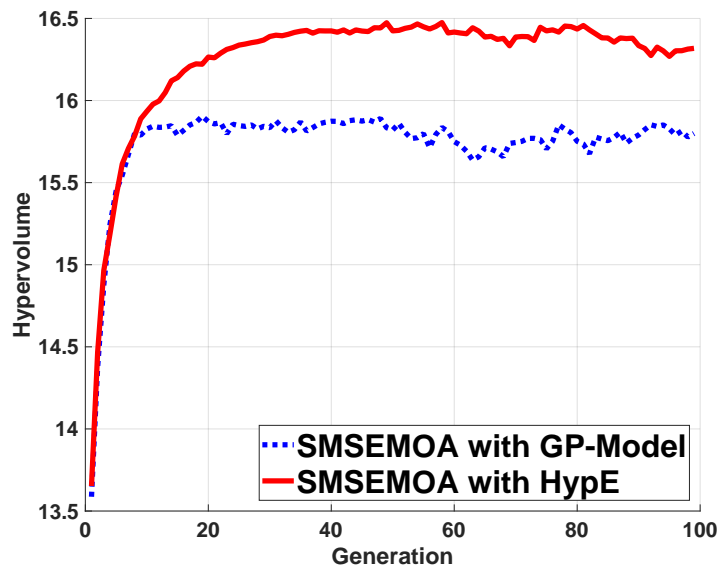
**Figura 5.23:** Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ2 con 4 funciones objetivo.



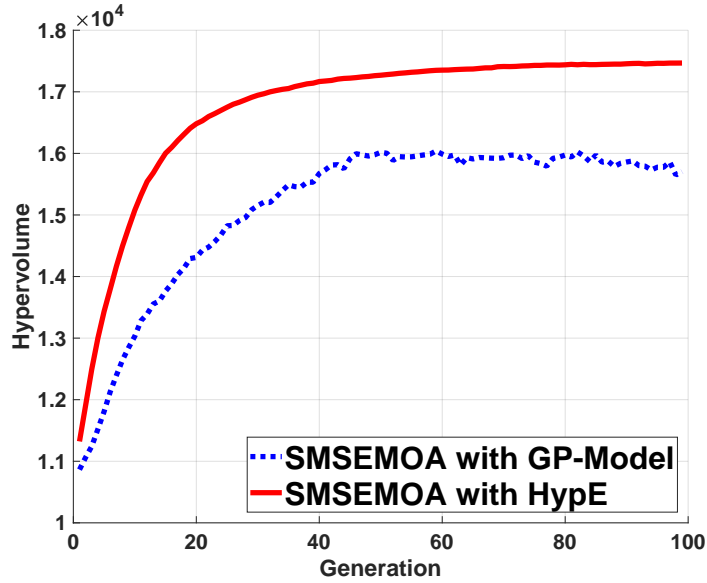
**Figura 5.24:** Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ3 con 4 funciones objetivo.



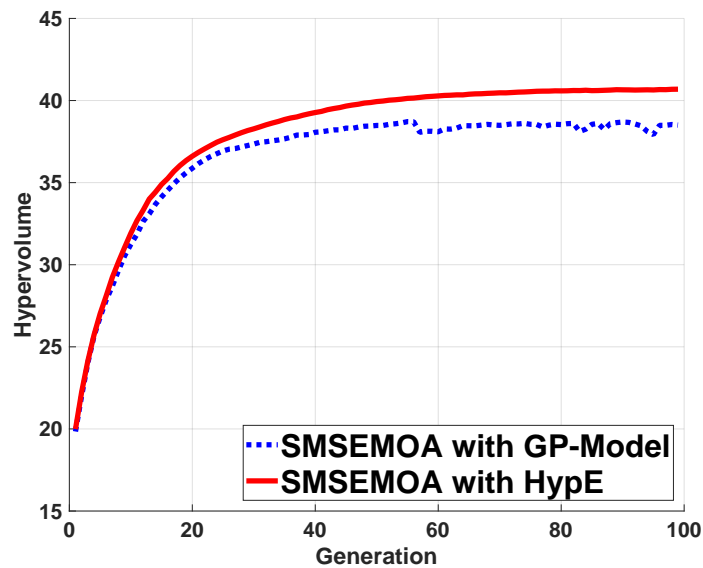
**Figura 5.25:** Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ4 con 4 funciones objetivo.



**Figura 5.26:** Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ5 con 4 funciones objetivo.



**Figura 5.27:** Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ6 con 4 funciones objetivo.



**Figura 5.28:** Curvas de hipervolumen obtenidas por el algoritmo SMS-EMOA guiado por HypE y el algoritmo SMS-EMOA guiado por el algoritmo basado en modelos GP en la búsqueda del frente de Pareto del problema DTLZ7 con 4 funciones objetivo.

## Capítulo 6

# Conclusiones y trabajo futuro

### 6.1. Conclusiones

El objetivo principal de este trabajo de tesis propone entrenar modelos utilizando Programación Genética para aproximar el valor de hipervolumen. Este trabajo de investigación es motivado por el problema del costo computacional que implica el cálculo del valor de hipervolumen.

Para llevar a cabo este objetivo se realizaron estudios para evaluar los modelos como indicador, para ello se utilizaron archivos generados de siete problemas de prueba del benchmark DTLZ. El valor real de hipervolumen fue calculado con el algoritmo WFG y comparado contra las aproximaciones realizadas por los modelos donde los resultados fueron reportados en error absoluto, error relativo y la precisión del modelo.

Los modelos propuestos mostraron ser capaces de reducir el tiempo de ejecución a un tiempo polinomial obteniendo una gran ventaja sobre uno de los algoritmos que está catalogado como uno de los más rápidos para el cálculo de hipervolumen. En los resultados reportados se puede observar como la diferencia en tiempo se incrementa conforme aumenta la cantidad de puntos en el conjunto de soluciones.



Las aproximaciones realizadas por los modelos varían dependiendo de la forma en la que son entrenados. Los modelos entrenados con bases no normalizadas aproximan el valor de hipervolumen con una exactitud entre 90 % y 98 %, mientras que un entrenamiento normalizado permite aproximar el valor de hipervolumen con una exactitud entre 97 % y 99 %. La exactitud varía para cada problema dependiendo de la combinación de bases de datos utilizada para el entrenamiento. Los problemas que presentaron una mayor complejidad al ser generalizados fueron el DTLZ5 y DTLZ6.

Los modelos utilizados como indicador muestran tener buenos resultados en la calidad de las aproximaciones al valor de hipervolumen. Sin embargo, no son 100 % exactos. Conforme a los resultados obtenidos podemos concluir que tenemos una forma muy rápida de aproximar el valor de hipervolumen con un pequeño margen de error.

En el segundo enfoque se realizaron estudios para evaluar el desempeño del algoritmo SMS-EMOA que utiliza los modelos como método de selección. Para realizar este análisis se utilizó el indicador WFG para calcular el valor de hipervolumen del mejor conjunto de soluciones no dominadas de cada generación. Los resultados reportados son la aproximación realizada por el algoritmo al conjunto de óptimos de Pareto y una curva de hipervolumen donde se puede observar la evolución del optimizador en la búsqueda del frente de Pareto.

Los resultados experimentales confirman que el algoritmo SMS-EMOA guiado por los modelos es capaz de realizar una buena aproximación al conjunto de óptimos de Pareto, esto a pesar de la pérdida de precisión que tienen los modelos. Conforme los resultados obtenidos para este segundo enfoque podemos concluir que la precisión de los modelos está dentro de los límites tolerables y que pueden reducir el tiempo de ejecución significativamente de los algoritmos basados en indicadores. El éxito de la técnica propuesta depende en gran medida de la forma de entrena-



miento que reciben los diferentes modelos ya que en base a eso es posible reducir el error y lograr mejores aproximaciones. Para este trabajo se cumplió el objetivo al poder emplear un modelo entrenado para aproximar el valor de hipervolumen además de utilizarlo para guiar la búsqueda de un algoritmo multi-objetivo basado en indicadores.

## 6.2. Trabajo futuro

En este apartado se mencionan algunas ideas y estrategias que se pueden seguir para extender este trabajo de tesis.

- Se pueden explorar otras formas de generar las bases de datos, el éxito de un modelo generalmente depende de su entrenamiento. Pueden realizarse diferentes configuraciones de los parámetros de control del optimizador. De igual modo se puede variar la cantidad de datos que son almacenados y la parte del evolutivo de donde son extraídos. Algunos de estos ajustes alternativos podrían resultar en una mejor precisión de predicción.
- Incorporar nuevas estadísticas o reemplazar algunas de las que están siendo utilizadas. Generalmente los modelos no utilizan todas las características que conforman el vector de variables predictoras. Algunas otras características podrían resultar en una mejora en el rendimiento de los modelos.
- Utilizar otros conjuntos de prueba que permitan tener una visión más amplia del alcance de los modelos y los casos para los cuales resultan más adecuados.
- Ampliar el estudio a problemas con más de cuatro funciones objetivo que permita medir el alcance de los modelos en problemas de alta dimensionalidad.

# Referencias

- [1] J. Branke, J. Branke, K. Deb, K. Miettinen, and R. Slowiński, *Multiobjective optimization: Interactive and evolutionary approaches*. Springer Science & Business Media, 2008, vol. 5252.
- [2] R. Rosenberg, "Simulation of genetic populations with biochemical properties (ph. d. thesis). university of michigan," *Ann Arbor, Michigan*, 1967.
- [3] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multi-objective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.
- [4] E. Zitzler, D. Brockhoff, and L. Thiele, "The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2007, pp. 862–876.
- [5] J. Wu and S. Azarm, "Metrics for quality assessment of a multiobjective design optimization solution set," *Journal of Mechanical Design*, vol. 123, no. 1, pp. 18–25, 2001.
- [6] N. Beume, "S-metric calculation by considering dominated hypervolume as kleeś measure problem," *Evolutionary Computation*, vol. 17, no. 4, pp. 477–492, 2009.
- [7] V. Klee, "Can the measure of be computed in less than  $o(n \log n)$  steps?" *The American Mathematical Monthly*, vol. 84, no. 4, pp. 284–285, 1977.
- [8] M. Fleischer, "The measure of pareto optima applications to multi-objective metaheuristics," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2003, pp. 519–533.



- [9] L. While, "A new analysis of the lebmeasure algorithm for calculating hypervolume," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2005, pp. 326–340.
- [10] E. Zitzler, "Hypervolume metric calculation," *Computer Engineering and Networks Laboratory (TIK), Zürich*, 2001.
- [11] J. D. Knowles, "Local-search and hybrid evolutionary algorithms for pareto optimization," Ph.D. dissertation, University of Reading Reading, 2002.
- [12] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *IEEE transactions on evolutionary computation*, vol. 10, no. 1, pp. 29–38, 2006.
- [13] C. M. Fonseca, L. Paquete, and M. López-Ibáñez, "An improved dimension-sweep algorithm for the hypervolume indicator," in *2006 IEEE international conference on evolutionary computation*. IEEE, 2006, pp. 1157–1163.
- [14] L. While, L. Bradstreet, and L. Barone, "A fast way of calculating exact hypervolumes," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 86–95, 2011.
- [15] J. Bader and E. Zitzler, "Hype: An algorithm for fast hypervolume-based many-objective optimization," *Evolutionary computation*, vol. 19, no. 1, pp. 45–76, 2011.
- [16] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2004, pp. 832–842.
- [17] N. Beume, B. Naujoks, and M. Emmerich, "Sms-emoa: Multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 2007.
- [18] K. Bringmann and T. Friedrich, "Approximating the least hypervolume contributor: Np-hard in general, but fast in practice," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2009, pp. 6–20.
- [19] —, "Approximating the volume of unions and intersections of high-dimensional geometric objects," *Computational Geometry*, vol. 43, no. 6-7, pp. 601–610, 2010.



- [20] W. Stadler, "A survey of multicriteria optimization or the vector maximum problem, part i: 1776–1960," *Journal of Optimization Theory and Applications*, vol. 29, no. 1, pp. 1–52, 1979.
- [21] I. Rechenberg, "Evolutionsstrategien," in *Simulationsmethoden in der Medizin und Biologie*. Springer, 1978, pp. 83–114.
- [22] W. Vent, "Rechenberg, Ingo, evolutionsstrategie—optimierung technischer systeme nach prinzipien der biologischen evolution. 170 s. mit 36 abb. frommann-holzboog-verlag. stuttgart 1973. broschiert," *Feddes Repertorium*, vol. 86, no. 5, pp. 337–337, 1975.
- [23] J. H. Holland, "Outline for a logical theory of adaptive systems," *Journal of the ACM (JACM)*, vol. 9, no. 3, pp. 297–314, 1962.
- [24] J. H. Holland et al., *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.
- [25] L. J. Fogel, "On the organization of intellect," Ph.D. dissertation, University of California, Los Angeles–Engineering, 1964.
- [26] L. J. Fogel, A. J. Owens, and M. J. Walsh, "Intelligent decision making through a simulation of evolution," *Behavioral science*, vol. 11, no. 4, pp. 253–272, 1966.
- [27] E. K. Chong and S. H. Zak, *An introduction to optimization*. John Wiley & Sons, 2013, vol. 76.
- [28] F. Y. Edgeworth, *Mathematical psychics: An essay on the application of mathematics to the moral sciences*. Kegan Paul, 1881, vol. 10.
- [29] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proceedings of the First International Conference on Genetic Algorithms and Their Applications, 1985*. Lawrence Erlbaum Associates. Inc., Publishers, 1985.
- [30] E. G. David et al., "Genetic algorithms in search," *Optimization and Machine Learning, Reading, Massachusetts*, 1989.
- [31] P. Hajela and C.-Y. Lin, "Genetic search strategies in multicriterion optimal design," *Structural optimization*, vol. 4, no. 2, pp. 99–107, 1992.



- [32] K. Deb and D. E. Goldberg, "An investigation of niche and species formation in genetic algorithms," in *Proceedings of the Third International 152 Conference on Genetic Algorithms*, 1989.
- [33] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
- [34] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on evolutionary computation*, vol. 8, no. 3, pp. 256–279, 2004.
- [35] T. E. Koch and A. Zell, "Mocs: Multi-objective clustering selection evolutionary algorithm," in *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*. Morgan Kaufmann Publishers Inc., 2002, pp. 423–430.
- [36] D. E. Goldberg, J. Richardson *et al.*, "Genetic algorithms with sharing for multimodal function optimization," in *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*. Hillsdale, NJ: Lawrence Erlbaum, 1987, pp. 41–49.
- [37] X. Cui, M. Li, and T. Fang, "Study of population diversity of multiobjective evolutionary algorithm based on immune and entropy principles," in *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546)*, vol. 2. IEEE, 2001, pp. 1316–1321.
- [38] H. Kita, Y. Yabumoto, N. Mori, and Y. Nishikawa, "Multi-objective optimization by means of the thermodynamical genetic algorithm," in *International Conference on Parallel Problem Solving from Nature*. Springer, 1996, pp. 504–512.
- [39] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii," in *International conference on parallel problem solving from nature*. Springer, 2000, pp. 849–858.
- [40] J. D. Knowles, L. Thiele, and E. Zitzler, "A tutorial on the performance assessment of stochastic multiobjective optimizers," *TIK-Report*, vol. 214, 2006.
- [41] E. Zitzler, J. Knowles, and L. Thiele, "Quality assessment of pareto set approximations," in *Multiobjective Optimization*. Springer, 2008, pp. 373–404.

- [42] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary computation*, vol. 8, no. 2, pp. 173–195, 2000.
- [43] M. Li and X. Yao, "Quality evaluation of solution sets in multiobjective optimization: A survey," *ACM Computing Surveys (CSUR)*, vol. 52, no. 2, p. 26, 2019.
- [44] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms—a comparative case study," in *International conference on parallel problem solving from nature*. Springer, 1998, pp. 292–301.
- [45] L. Paquete, C. M. Fonseca, and M. López-Ibáñez, "An optimal algorithm for a special case of klee's measure problem in three dimensions," 2006.
- [46] M. H. Overmars and C.-K. Yap, "New upper bounds in klee's measure problem," *SIAM Journal on Computing*, vol. 20, no. 6, pp. 1034–1045, 1991.
- [47] L. While and L. Bradstreet, "Applying the wfg algorithm to calculate incremental hypervolumes," in *2012 IEEE Congress on Evolutionary Computation*. IEEE, 2012, pp. 1–8.
- [48] L. Bradstreet, *The hypervolume indicator for multi-objective optimisation: calculation and use*. University of Western Australia Perth, 2011.
- [49] V. Braberman, J. L. Obes, A. Olivero, and F. Schapachnik, "Hypervolume approximation in timed automata model checking," in *International Conference on Formal Modeling and Analysis of Timed Systems*. Springer, 2007, pp. 69–81.
- [50] J. Bader, K. Deb, and E. Zitzler, "Faster hypervolume-based search using monte carlo sampling," in *Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems*. Springer, 2010, pp. 313–326.
- [51] R. M. Everson, J. E. Fieldsend, and S. Singh, "Full elite sets for multi-objective optimisation," in *Adaptive Computing in Design and Manufacture V*. Springer, 2002, pp. 343–354.
- [52] H. Ishibuchi, N. Tsukamoto, Y. Sakane, and Y. Nojima, "Hypervolume approximation using achievement scalarizing functions for evolutionary many-objective optimization," in *2009 IEEE Congress on Evolutionary Computation*. IEEE, 2009, pp. 530–537.



- [53] J. R. Koza and J. R. Koza, *Genetic programming: on the programming of computers by means of natural selection*. MIT press, 1992, vol. 1.
- [54] N. L. Cramer, "A representation for the adaptive generation of simple sequential programs," in *Proceedings of the first international conference on genetic algorithms*, 1985, pp. 183–187.
- [55] J. F. Hicklin, "Application," in *Application of the Genetic Algorithm to Automatic Program Generation*. M.S. thesis, Department of Computer Science, University of Idaho, 1986.
- [56] C. Fujiki, "An evaluation of holland's genetic algorithm applied to a program generator. m.s. thesis, department of computer science, university of idaho," 1986.
- [57] C. Fujiki and J. Dickinson, "Using the genetic algorithm to generate lisp source code to solve the prisoner's dilemma," in *Genetic algorithms and their applications: proceedings of the second International Conference on Genetic Algorithms: July 28-31, 1987 at the Massachusetts Institute of Technology, Cambridge, MA*. Hillsdale, NJ: L. Erlbaum Associates, 1987., 1987.
- [58] T. Soule, J. A. Foster *et al.*, "Code size and depth flows in genetic programming," *Genetic programming*, pp. 313–320, 1997.
- [59] D. P. Searson, D. E. Leahy, and M. J. Willis, "Gptips: an open source genetic programming toolbox for multigene symbolic regression," in *Proceedings of the International multiconference of engineers and computer scientists*, vol. 1. IMECS Hong Kong, 2010, pp. 77–80.
- [60] S. Luke and L. Panait, "Lexicographic parsimony pressure," in *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*. Morgan Kaufmann Publishers Inc., 2002, pp. 829–836.
- [61] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, vol. 1. IEEE, 2002, pp. 825–830.
- [62] S. Huband, L. Barone, L. While, and P. Hingston, "A scalable multi-objective test problem toolkit," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2005, pp. 280–295.



- [63] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.
- [64] D. P. Searson, "Gptips 2: an open-source software platform for symbolic data mining," in *Handbook of genetic programming applications*. Springer, 2015, pp. 551–573.