



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS  
DEL INSTITUTO POLITÉCNICO NACIONAL

UNIDAD ZACATENCO  
DEPARTAMENTO DE COMPUTACIÓN

# **Diseño de una nueva función de escalarización usando programación genética**

## **T E S I S**

Que presenta

**Amín Vanya Bernabé Rodríguez**

Para obtener el grado de

**Maestro en Ciencias en Computación**

Director de la Tesis:

**Dr. Carlos Artemio Coello Coello**

Ciudad de México

Diciembre, 2019

---

# Resumen

Una gran variedad de problemas del mundo real pueden ser modelados como problemas en los que es necesario optimizar de forma simultánea dos o más funciones objetivo, las cuales se encuentran normalmente en conflicto entre sí. Éstos se conocen como Problemas de Optimización Multi-objetivo (POMs) y se han desarrollado distintas técnicas para su solución. En particular, los Algoritmos Evolutivos Multi-objetivo (AEMOs) han sido ampliamente utilizados para este fin.

Se han propuesto diferentes tipos de AEMOs para la solución de POMs. Uno de los más utilizados es el basado en optimalidad de Pareto, el cual clasifica las soluciones con base en el número de individuos que la dominan. Sin embargo, existe evidencia experimental que indica que los algoritmos basados en este criterio deterioran su desempeño conforme se aumenta el número de objetivos. Esto ha motivado el desarrollo de otros enfoques, tales como los algoritmos basados en indicadores y los basados en descomposición. Los métodos de descomposición y algunos de los basados en indicadores utilizan funciones de escalarización.

Una función de escalarización es un tipo de función matemática que permite transformar un POM en varios problemas mono-objetivo, adoptando pesos para definir diferentes direcciones de búsqueda. Dos de las principales ventajas de estas funciones son su facilidad de implementación y la posibilidad de incorporar información sobre las preferencias del usuario en el proceso de solución. Si bien existen en la literatura diversas funciones de escalarización, todas ellas han sido propuestas por humanos.

En esta tesis, proponemos el uso de programación genética para generar nuevas funciones de escalarización de forma automática. Existen estudios que muestran que el desempeño de los AEMOs basados en funciones de escalarización depende directamente de la función utilizada, por lo que consideramos relevante el proponer nuevas funciones de escalarización. Mediante la implementación propuesta en esta tesis pudimos producir nuevas funciones de escalarización, de las cuales elegimos dos ( $F4A$  y  $F4B$ ) para realizar una validación experimental. Para fines de dicho estudio, utilizamos la heurística MOMBI-II como nuestro AEMO base, adoptando un conjunto de problemas de prueba estándar y comparando resultados mediante el uso de tres indicadores de desempeño: el hipervolumen, la energía-s y la cobertura de conjuntos.

Los resultados obtenidos indican que la función  $F4A$  logra un desempe-

---

ño similar al de la función  $ASF$ , que es la utilizada por defecto en MOMBI-II. Por otro lado, la función  $F4B$  logra un mejor desempeño en la mayoría de los problemas de prueba en comparación con  $ASF$ . De esta forma, consideramos que la implementación propuesta es una forma válida de generar nuevas funciones de escalarización que sean competitivas con respecto a funciones utilizadas en el área.

# Abstract

A great variety of real-world problems can be modelled as problems in which it is necessary to simultaneously optimize two or more objective functions, which are normally in conflict with each other. These are known as Multi-objective Optimization Problems (MOPs) and different techniques have been developed to solve them. Particularly, Multi-objective Evolutionary Algorithms (MOEAs) have been widely used for this sake.

Different types of MOEAs have been proposed to solve MOPs. One of the most commonly used is the one based on Pareto optimality, which classifies solutions based on the number of individuals that dominate it. However, there is experimental evidence indicating that algorithms based on this criterion deteriorate their performance as the number of objectives increases. This has motivated the development of other approaches, such as algorithms based on performance indicators and on decomposition. Algorithms based on decomposition, as well as some indicator-based approaches rely on the use of scalarizing functions.

A scalarizing function is a mathematical function that allows to transform a MOP into several single-objective problems, using weights to define different search directions. Two of the main advantages of scalarizing functions are their ease of implementation and their possibility to incorporate information about the user's preferences during the search process. Although a variety of scalarizing functions are available in the literature, all of them have been proposed by humans.

In this thesis, genetic programming is proposed to automatically generate new scalarizing functions. There are studies indicating that the performance of MOEAs based on scalarizing functions directly depends on the adopted function. Thus, it is relevant to propose new scalarizing functions. Through the implementation proposed in this thesis, we were able to produce new scalarizing functions, from which two were selected (*F4A* and *F4B*) to perform an experimental validation. For the purposes of this study, we used MOMBI-II as our baseline MOEA, and we adopted a set of standard test problems, comparing results through the use of three performance indicators: hypervolume, s-energy and coverage of two sets.

The obtained results indicate that *F4A* has a similar performance to that of the scalarizing function *ASF*, which is the one adopted by default in MOMBI-II. On the other hand, the function *F4B* has a better performance in most of the test problems as compared to *ASF*. This way, we consider that

---

the proposed implementation is a valid way of generating new scalarizing functions that are competitive with respect to other functions adopted in the area.

# Agradecimientos

En primer lugar, agradezco a Dios, por prestarme vida y por las oportunidades que me permitieron culminar esta etapa de mi vida.

Agradezco enormemente a mis padres, Débora Rodríguez y Amín Bernabé, sin cuyas enseñanzas y amor no sería hoy la persona que soy. A mis hermanas, Betsie y Deneb, gracias por su incondicional apoyo.

Agradezco especialmente al Dr. Carlos Coello, por su incansable labor al momento de orientarme en el desarrollo de este trabajo, y también por cada consejo otorgado. Sin su conocimiento, disposición y paciencia, no me habría sido posible concluir esta maestría.

Agradezco a mis revisores de tesis, Dr. Amilcar Meneses y Dr. Hugo Escalante, por el tiempo invertido y por sus atinados comentarios que permitieron enriquecer este trabajo.

A mis compañeros de generación, que en este proceso se convirtieron en buenos amigos: David, Fernanda, Michel. Gracias por todos los buenos momentos compartidos y por las palabras de apoyo que creo que todos necesitamos en algún momento. A mis amigos: Ricardo, Ximena, Cristina, Itzel, gracias por estar ahí sin importar las circunstancias.

Agradezco al CINVESTAV y a CONACyT por los apoyos económicos otorgados durante la realización de mi maestría. Este trabajo de tesis fue derivado del proyecto CONACyT titulado “Esquemas de Selección Alternativos para Algoritmos Evolutivos Multi-Objetivo” (Ref. 2016-01-1920) y del proyecto SEP-Cinvestav titulado “Nuevos Esquemas de Selección para Algoritmos Evolutivos Multi-Objetivo basados en Indicadores de Desempeño” (solicitud no. 4). El investigador responsable de estos dos proyectos es el Dr. Carlos A. Coello Coello.

---

# Índice general

<b>Lista de figuras</b>	<b>XIII</b>
<b>Lista de tablas</b>	<b>XV</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Antecedentes y motivación . . . . .	1
1.2. Planteamiento del problema y propuesta de solución . . . . .	2
1.3. Hipótesis . . . . .	3
1.4. Estructura de la tesis . . . . .	4
<b>2. Computación evolutiva</b>	<b>5</b>
2.1. Conceptos básicos . . . . .	5
2.2. Paradigmas de cómputo evolutivo . . . . .	7
2.2.1. Estrategias Evolutivas . . . . .	8
2.2.2. Algoritmos Genéticos . . . . .	9
2.2.3. Programación Evolutiva . . . . .	11
2.3. Programación Genética . . . . .	11
2.3.1. Terminales y funciones . . . . .	12
2.3.2. Representación . . . . .	13
2.3.3. Inicialización . . . . .	15
2.3.4. Operadores genéticos . . . . .	17
2.4. Resumen . . . . .	20
<b>3. Problemas de Optimización Multiobjetivo</b>	<b>21</b>
3.1. Problemas de Optimización . . . . .	21
3.2. Conceptos básicos . . . . .	23
3.3. Técnicas de solución de Problemas de Optimización Multi- objetivo . . . . .	25
3.3.1. Técnicas de solución enumerativas . . . . .	26
3.3.2. Técnicas de solución deterministas . . . . .	26
3.3.3. Técnicas de solución estocásticas . . . . .	28
3.4. Problemas de optimización con muchas funciones objetivo . . . . .	29
3.5. Resumen . . . . .	31

<b>4. Funciones de escalarización</b>	<b>33</b>
4.1. Solución de POMs mediante descomposición . . . . .	33
4.2. Clasificación de funciones de escalarización . . . . .	35
4.2.1. Funciones de escalarización basadas en clasificación . .	35
4.2.2. Funciones de escalarización basadas en puntos de re- ferencia . . . . .	36
4.3. Revisión de funciones de escalarización . . . . .	36
4.4. Resumen . . . . .	42
<b>5. Implementación propuesta</b>	<b>43</b>
5.1. Programación Genética . . . . .	43
5.1.1. Representación de los individuos . . . . .	44
5.1.2. Búsqueda local inspirada en mecanismos epigenéticos	45
5.2. Algoritmo Evolutivo Multi-objetivo . . . . .	47
5.3. Implementación propuesta . . . . .	49
5.3.1. Parámetros necesarios . . . . .	52
5.3.2. Archivos generados . . . . .	53
5.3.3. Requerimientos de software . . . . .	53
5.4. Generación de nuevas funciones . . . . .	53
5.5. Resumen . . . . .	55
<b>6. Validación experimental</b>	<b>57</b>
6.1. Problemas de prueba utilizados . . . . .	57
6.1.1. Conjunto Deb-Thiele-Laumanns-Zitzler (DTLZ) . . . .	58
6.1.2. Walking Fish Group (WFG) . . . . .	65
6.1.3. I-DTLZ . . . . .	74
6.2. Indicadores de desempeño . . . . .	76
6.2.1. Hipervolumen . . . . .	77
6.2.2. Energía-s . . . . .	77
6.2.3. Cobertura de conjuntos . . . . .	78
6.3. Resultados . . . . .	78
6.3.1. Hipervolumen . . . . .	79
6.3.2. Energía-s . . . . .	81
6.3.3. Cobertura de conjuntos . . . . .	84
6.4. Resumen . . . . .	85
<b>7. Conclusiones y trabajo futuro</b>	<b>87</b>

<b>A. Parámetros de configuración</b>	<b>91</b>
A.1. Parámetros de configuración de ELGP . . . . .	91
A.2. Parámetros de configuración de MOMBI-II . . . . .	96
A.3. Ejemplo de archivo de pesos para MOMBI-II . . . . .	96
<b>B. Parámetros utilizados para la generación de nuevas funciones de   escalarización</b>	<b>101</b>
B.1. Archivo de configuración de ELGP . . . . .	101
B.2. Archivo de configuración de MOMBI-II . . . . .	103
<b>C. Puntos de referencia</b>	<b>107</b>
<b>D. Resultados numéricos</b>	<b>111</b>
D.1. Hipervolumen . . . . .	112
D.2. Energía-s . . . . .	116
D.3. Cobertura de conjuntos . . . . .	120
<b>E. Prueba de suma de rangos de Wilcoxon</b>	<b>125</b>
E.1. Hipervolumen . . . . .	127
E.2. Energía-s . . . . .	134
<b>Bibliografía</b>	<b>141</b>



# Índice de figuras

2.1.	<i>Programa almacenado en representación de árbol.</i>	14
2.2.	<i>Programa almacenado en representación lineal.</i>	15
2.3.	<i>Árbol inicializado con el algoritmo Grow.</i>	16
2.4.	<i>Árbol inicializado con el algoritmo Full.</i>	16
2.5.	<i>Cruza de dos programas en representación de árbol.</i>	18
2.6.	<i>Cruza de dos programas en representación lineal.</i>	19
2.7.	<i>Mutación de un individuo en representación de árbol.</i>	19
2.8.	<i>Mutación de un individuo en representación lineal.</i>	20
4.1.	<i>Curvas de nivel de la función WS con <math>\vec{w} = (0.5 \ 0.5)^T</math>.</i>	37
4.2.	<i>Curvas de nivel de la función CH con <math>\vec{w} = (0.5 \ 0.5)^T</math>.</i>	38
4.3.	<i>Curvas de nivel de la función ACH con <math>\vec{w} = (0.5 \ 0.5)^T</math>.</i>	39
4.4.	<i>Curvas de nivel de la función ASF con <math>\vec{w} = (0.5 \ 0.5)^T</math>.</i>	40
4.5.	<i>Curvas de nivel de la función AASF con <math>\vec{w} = (0.5 \ 0.5)^T</math>.</i>	40
4.6.	<i>Curvas de nivel de la función PBI con <math>\vec{w} = (0.5 \ 0.5)^T</math>.</i>	41
5.1.	<i>Diagrama del funcionamiento de MOMBI-II.</i>	49
5.2.	<i>Diagrama del funcionamiento de la implementación propuesta para generar funciones de escalarización.</i>	50
6.1.	<i>Frente de Pareto de DTLZ1 con 3 funciones objetivo.</i>	59
6.2.	<i>Frente de Pareto de DTLZ2 con 3 funciones objetivo.</i>	60
6.3.	<i>Frente de Pareto de DTLZ3 con 3 funciones objetivo.</i>	61
6.4.	<i>Frente de Pareto de DTLZ4 con 3 funciones objetivo.</i>	62
6.5.	<i>Frente de Pareto de DTLZ5 con 3 funciones objetivo.</i>	63
6.6.	<i>Frente de Pareto de DTLZ6 con 3 funciones objetivo.</i>	64
6.7.	<i>Frente de Pareto de DTLZ7 con 3 funciones objetivo.</i>	65
6.8.	<i>Frente de Pareto de WFG1 con 3 funciones objetivo.</i>	66
6.9.	<i>Frente de Pareto de WFG2 con 3 funciones objetivo.</i>	67
6.10.	<i>Frente de Pareto de WFG3 con 3 funciones objetivo.</i>	68
6.11.	<i>Frente de Pareto de WFG4 con 3 funciones objetivo.</i>	69
6.12.	<i>Frente de Pareto de WFG5 con 3 funciones objetivo.</i>	70
6.13.	<i>Frente de Pareto de WFG6 con 3 funciones objetivo.</i>	71
6.14.	<i>Frente de Pareto de WFG7 con 3 funciones objetivo.</i>	72
6.15.	<i>Frente de Pareto de WFG8 con 3 funciones objetivo.</i>	73
6.16.	<i>Frente de Pareto de WFG9 con 3 funciones objetivo.</i>	74
6.17.	<i>Frente de Pareto de IDTLZ1 con 3 funciones objetivo.</i>	74

6.18. Frente de Pareto de IDTLZ2 con 3 funciones objetivo. . . . .	75
6.19. Frente de Pareto de IDTLZ3 con 3 funciones objetivo. . . . .	75
6.20. Frente de Pareto de IDTLZ4 con 3 funciones objetivo. . . . .	75
6.21. Frente de Pareto de IDTLZ5 con 3 funciones objetivo. . . . .	76
6.22. Frente de Pareto de IDTLZ6 con 3 funciones objetivo. . . . .	76
6.23. Frente de Pareto de IDTLZ7 con 3 funciones objetivo. . . . .	76
6.24. Comparación de problemas mejorados por las funciones F4A y ASF en relación a la familia de problemas de prueba y el número de objetivos usando el indicador de hipervolumen. . . . .	80
6.25. Comparación de problemas mejorados por las funciones F4B y ASF en relación a la familia de problemas de prueba y el número de objetivos usando el indicador de hipervolumen. . . . .	81
6.26. Comparación de problemas mejorados por las funciones F4A y ASF en relación a la familia de problemas de prueba y el número de objetivos usando el indicador de energía-s. . . . .	83
6.27. Comparación de problemas mejorados por las funciones F4B y ASF en relación a la familia de problemas de prueba y el número de objetivos usando el indicador de energía-s. . . . .	84
6.28. Comparación de funciones F4A, F4B y ASF en función del número de objetivos usando el indicador de cobertura de conjuntos. . . . .	85

# Índice de tablas

C.1. Puntos de referencia utilizados para problemas de prueba en 2 dimensiones. . . . .	107
C.2. Puntos de referencia utilizados para problemas de prueba en 3 dimensiones. . . . .	108
C.3. Puntos de referencia utilizados para problemas de prueba en 4 dimensiones. . . . .	108
C.4. Puntos de referencia utilizados para problemas de prueba en 5 dimensiones. . . . .	109
C.5. Puntos de referencia utilizados para problemas de prueba en 6 dimensiones. . . . .	109
C.6. Puntos de referencia utilizados para problemas de prueba en 10 dimensiones. . . . .	110
D.1. Comparación de indicador de hipervolumen en problemas con 2 funciones objetivo. . . . .	112
D.2. Comparación de indicador de hipervolumen en problemas con 3 funciones objetivo. . . . .	113
D.3. Comparación de indicador de hipervolumen en problemas con 4 funciones objetivo. . . . .	113
D.4. Comparación de indicador de hipervolumen en problemas con 5 funciones objetivo. . . . .	114
D.5. Comparación de indicador de hipervolumen en problemas con 6 funciones objetivo. . . . .	114
D.6. Comparación de indicador de hipervolumen en problemas con 10 funciones objetivo. . . . .	115
D.7. Comparación de indicador de energía-s en problemas con 2 funciones objetivo. . . . .	116
D.8. Comparación de indicador de energía-s en problemas con 3 funciones objetivo. . . . .	117
D.9. Comparación de indicador de energía-s en problemas con 4 funciones objetivo. . . . .	117
D.10. Comparación de indicador de energía-s en problemas con 5 funciones objetivo. . . . .	118
D.11. Comparación de indicador de energía-s en problemas con 6 funciones objetivo. . . . .	118

D.12.Comparación de indicador de energía-s en problemas con 10 funciones objetivo. . . . .	119
D.13.Comparación de indicador de cobertura de conjuntos en problemas con 2 funciones objetivo. . . . .	120
D.14.Comparación de indicador de cobertura de conjuntos en problemas con 3 funciones objetivo. . . . .	121
D.15.Comparación de indicador de cobertura de conjuntos en problemas con 4 funciones objetivo. . . . .	121
D.16.Comparación de indicador de cobertura de conjuntos en problemas con 5 funciones objetivo. . . . .	122
D.17.Comparación de indicador de cobertura de conjuntos en problemas con 6 funciones objetivo. . . . .	122
D.18.Comparación de indicador de cobertura de conjuntos en problemas con 10 funciones objetivo. . . . .	123
E.1. Prueba de suma de rangos de Wilcoxon aplicada al indicador de hipervolumen en problemas de prueba con 2 funciones objetivo. . . . .	127
E.2. Prueba de suma de rangos de Wilcoxon aplicada al indicador de hipervolumen en problemas de prueba con 3 funciones objetivo. . . . .	128
E.3. Prueba de suma de rangos de Wilcoxon aplicada al indicador de hipervolumen en problemas de prueba con 4 funciones objetivo. . . . .	129
E.4. Prueba de suma de rangos de Wilcoxon aplicada al indicador de hipervolumen en problemas de prueba con 5 funciones objetivo. . . . .	130
E.5. Prueba de suma de rangos de Wilcoxon aplicada al indicador de hipervolumen en problemas de prueba con 6 funciones objetivo. . . . .	131
E.6. Prueba de suma de rangos de Wilcoxon aplicada al indicador de hipervolumen en problemas de prueba con 10 funciones objetivo. . . . .	132
E.7. Prueba de suma de rangos de Wilcoxon aplicada al indicador de energía-s en problemas de prueba con 2 funciones objetivo. . . . .	134
E.8. Prueba de suma de rangos de Wilcoxon aplicada al indicador de energía-s en problemas de prueba con 3 funciones objetivo. . . . .	135
E.9. Prueba de suma de rangos de Wilcoxon aplicada al indicador de energía-s en problemas de prueba con 4 funciones objetivo. . . . .	136

- E.10. Prueba de suma de rangos de Wilcoxon aplicada al indicador de energía-s en problemas de prueba con 5 funciones objetivo. 137
- E.11. Prueba de suma de rangos de Wilcoxon aplicada al indicador de energía-s en problemas de prueba con 6 funciones objetivo. 138
- E.12. Prueba de suma de rangos de Wilcoxon aplicada al indicador de energía-s en problemas de prueba con 10 funciones objetivo. 139





# INTRODUCCIÓN

Existen múltiples problemas del mundo real que se pueden modelar como un Problema de Optimización Multi-objetivo (POM) debido a que requieren la minimización o maximización de distintas funciones que se encuentran en conflicto entre sí. Algunas de las áreas en donde es común encontrar problemas de este tipo es en telecomunicaciones, redes, electrónica, robótica, transporte, construcción, economía, ingeniería civil, aeronáutica, hidráulica o naval, entre otras [10]. Para la solución de problemas de esta naturaleza existen distintas técnicas que se pueden clasificar en dos grandes áreas: técnicas de programación matemática y técnicas heurísticas bioinspiradas [14]. En este trabajo incluimos técnicas de ambas áreas y proponemos una forma de generar nuevas funciones de escalarización mediante el uso de programación genética, que es un tipo de técnica de computación evolutiva.

## 1.1 | ANTECEDENTES Y MOTIVACIÓN

La solución de un POM es un conjunto de vectores que representan los mejores “compromisos” entre las distintas funciones, al cual se le conoce como el conjunto de óptimos de Pareto [60]. Una de las técnicas para resolver POMs son los Algoritmos Evolutivos Multi-objetivo (AEMOs), los cuales son heurísticas inspiradas en la teoría sintética de la evolución que resuelven un problema utilizando una población de posibles soluciones las cuales son transformadas mediante diversos operadores a través de múltiples iteraciones.

Existe una gran variedad de AEMOs los cuales varían en el uso de distintos mecanismos de selección y operadores para transformar la población de

soluciones. En particular, existe una clase de AEMO que utilizan un criterio de selección conocido como optimalidad de Pareto. Sin embargo, se ha demostrado que este tipo de AEMO deteriora su desempeño cuando el POM tiene 4 funciones objetivo o más, debido a que el número de soluciones no dominadas incrementa de forma exponencial, disminuyendo la presión de selección del mecanismo de búsqueda [20]. Para mejorar el desempeño en este tipo de problemas existen dos tendencias principales en el área [26]:

1. *Hacer uso de indicadores de desempeño como mecanismo de selección.* Un indicador de desempeño es una métrica que cuantifica la calidad de un frente de Pareto. Si bien existen múltiples indicadores, el más ampliamente utilizado es el indicador del hipervolumen. Esto es debido a que se ha demostrado que dado un conjunto de búsqueda finito y un punto de referencia, la maximización del hipervolumen es equivalente a encontrar el verdadero frente de Pareto [21]. Sin embargo, dicho indicador tiene la desventaja de ser computacionalmente muy costoso, lo cual se ve agravado conforme se aumenta el número de objetivos. Por lo tanto, se han investigado alternativas tales como la aproximación del hipervolumen o el uso de otros indicadores de desempeño, como el indicador R2 [27].
2. *Hacer uso de funciones de escalarización.* Una función de escalarización es una función que permite transformar un problema de optimización multi-objetivo a un problema de optimización con un único objetivo utilizando algún tipo de información de preferencia. Las principales ventajas que presenta este enfoque son su simplicidad y eficiencia. Esto es debido a que se pueden calcular funciones de escalarización de forma relativamente fácil para tantos objetivos como tenga el problema.

## 1.2 | PLANTEAMIENTO DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

Múltiples AEMOs han utilizado un enfoque basado en el uso de funciones de escalarización [54, 74, 29, 32]. Sin embargo, no es común que se justifique la decisión de utilizar una función de escalarización determinada [38]. En un estudio sobre el comportamiento de distintas funciones realizado en [61] se analizó el impacto de 15 funciones de escalarización en el desempeño de dos AEMOs (MOEA/D y MOMBI-II), obteniéndose las siguientes conclusiones:

- El desempeño de los AEMOs probados depende directamente de la función de escalarización utilizada.
- No existe una única función de escalarización, de entre las funciones probadas, que permita obtener el mejor desempeño en todos los problemas de prueba, sino solamente en un subconjunto de éstos.

En primer lugar, buscamos implementar un algoritmo para generar nuevas funciones de escalarización, ya que conocemos su vital importancia en el desempeño de los AEMOs. Sin embargo, buscamos que esta generación de funciones se realice de forma automatizada, es decir, que sea la computadora la que realice una búsqueda inteligente de una nueva función de escalarización sin asistencia humana de ningún tipo.

Por otro lado, a partir de la segunda conclusión obtenida en [61], podemos observar lo siguiente. Si bien la validez de esta conclusión aplica única y específicamente para el conjunto de funciones de escalarización probadas, respalda el teorema de “No Free Lunch”, propuesto en [73]. Dicho teorema plantea que ninguna metaheurística puede obtener un mejor comportamiento que el resto de las metaheurísticas en todos los tipos problemas, sino únicamente en ciertas familias de problemas. Por lo tanto, no buscamos una función de escalarización que mejore el desempeño de todas las funciones existentes, sino únicamente que sea competitiva en la solución de al menos una clase particular de problemas multi-objetivo.

En este trabajo proponemos la creación de una implementación de software que combine dos elementos principales:

1. Una implementación de Programación Genética, la cual permite la generación de funciones de escalarización.
2. Un algoritmo de optimización multi-objetivo, el cual servirá para “guiar” al algoritmo de Programación Genética asignando un valor numérico a cada función generada, en términos de su capacidad para resolver un problema específico.

### 1.3 | HIPÓTESIS

El objetivo principal de este trabajo es comprobar la siguiente hipótesis:

- Es posible generar nuevas funciones de escalarización de forma automática mediante una implementación de Programación Genética de tal forma que éstas tengan un comportamiento que equipare o mejore

al obtenido al utilizar funciones de escalarización comúnmente adoptadas en el área.

## 1.4 | ESTRUCTURA DE LA TESIS

Incluyendo esta introducción, este trabajo está compuesto por siete capítulos y cinco apéndices.

En el capítulo 2 presentamos un panorama general del área de cómputo evolutivo al explicar brevemente los tres paradigmas principales que la componen.

En el capítulo 3 proveemos definiciones básicas de problemas de optimización multi-objetivo y de los métodos de solución que existen para su solución.

En el capítulo 4 explicamos el concepto de función de escalarización así como su impacto en la solución de problemas de optimización multi-objetivo. Además, presentamos algunas de las funciones de escalarización más comúnmente utilizadas en el área.

En el capítulo 5 presentamos formalmente los algoritmos que componen nuestra implementación para generar funciones de escalarización, así como una descripción de ésta y de su funcionamiento. Adicionalmente, presentamos 2 nuevas funciones de escalarización obtenidas mediante dicha implementación.

En el capítulo 6 presentamos la metodología seguida para validar experimentalmente el desempeño de las 2 funciones obtenidas y los resultados obtenidos.

En el capítulo 7 presentamos las conclusiones obtenidas y el trabajo futuro que puede derivarse de esta tesis.

El apéndice A presenta una lista completa de los parámetros utilizados para la generación de nuevas funciones de escalarización. El apéndice B presenta los parámetros utilizados para la generación de las 2 funciones presentadas en este trabajo. El apéndice C muestra una lista de los puntos de referencia con los que se realizaron las pruebas de cálculo de hipervolumen para la validación experimental. El apéndice D contiene tablas con los resultados numéricos obtenidos usando los indicadores de desempeño de hipervolumen, energía-s y cobertura de conjuntos en la validación experimental. Finalmente, el apéndice E contiene los valores obtenidos al aplicar la prueba de suma de rangos de Wilcoxon para validar el desempeño de las funciones de escalarización.



# COMPUTACIÓN EVOLUTIVA

La computación evolutiva es una rama de la inteligencia artificial, en particular del aprendizaje automático, que estudia y desarrolla técnicas heurísticas bioinspiradas. Estas técnicas se conocen como algoritmos evolutivos, y se basan en modelos simplificados de la evolución biológica. Su principal uso es en la solución de problemas de optimización global, en donde el problema a resolver se puede ver como análogo al proceso evolutivo.

Los algoritmos evolutivos cuentan con una población inicial de individuos que representan soluciones potenciales al problema. Durante la ejecución del algoritmo, dicha población se actualiza de forma iterativa buscando remover estocásticamente a los individuos menos deseados e introduciendo pequeños cambios en ellos. El resultado final es una población que ha incrementado gradualmente el desempeño de la población en la solución del problema.

Al tratarse de técnicas heurísticas no es posible garantizar que la solución encontrada sea exacta o que sea la mejor, sino únicamente, y en el mejor de los casos, que se trate de una buena aproximación. No obstante, estas técnicas son capaces de producir soluciones óptimas en un amplio rango de problemas. Además, existen múltiples problemas de aplicación real cuya solución mediante técnicas clásicas es demasiado costoso, mientras que con algoritmos evolutivos es posible resolverlos.

## 2.1 | CONCEPTOS BÁSICOS

Los algoritmos evolutivos están inspirados en la síntesis evolutiva moderna, teoría propuesta a principios del siglo XX. A grandes rasgos, esta

teoría propone que la evolución ocurre debido a los cambios en el material genético de los individuos de una población, a través de las generaciones, mediante la selección natural. Los mecanismos más importantes que provocan estos cambios son dos: la mutación y la recombinación (también llamada cruza).

Biológicamente, el material genético de cada organismo vivo está compuesto por el ácido desoxirribonucleico (ADN). Su función principal es la de codificar toda la información necesaria para construir determinados componentes de las células, por lo que es necesario para el desarrollo y funcionamiento de cada organismo. El ADN se organiza en estructuras llamadas **cromosomas** que son segmentos largos del mismo, encontrados en el núcleo de las células. A su vez, un cromosoma se divide en múltiples **genes**, los cuales son segmentos cortos de ADN que codifican una función bioquímica definida. Además, en un par de cromosomas homólogos, el valor que puede adquirir una determinada posición genética se llama **alelo**. Un alelo puede ser dominante o recesivo.

El ADN comprende la información genética de un organismo. Sin embargo, esto no implica que toda la información contenida en el ADN se utilizará en algún punto de la vida del organismo. Al conjunto de todo el material genético se le conoce como el **genotipo** de un individuo. Mientras que a la información genética que sí es utilizada (o expresada) y se convierte en alguna característica física del individuo, se le conoce como **fenotipo**.

Los algoritmos evolutivos emulan varios de los conceptos mencionados, con el fin de resolver un problema específico. Dependiendo de la forma de emular ciertos mecanismos, o del nivel de abstracción utilizado, surgen una variedad de técnicas distintas. No obstante, a pesar de las diferencias, todos los algoritmos evolutivos hacen una analogía entre el proceso evolutivo y la solución de un problema determinado. Además, existen elementos comunes en la mayoría de estos algoritmos. A continuación se describen algunos de los elementos más comunes:

- **Individuo:** Dado un problema específico, un individuo es la representación de una posible solución. Generalmente se trata de una estructura de datos con múltiples parámetros que modelan una versión simplificada de un organismo biológico.
- **Población:** Conjunto de individuos que pueden interactuar entre ellos mismos. Dependiendo del algoritmo evolutivo, el tamaño de la población puede ser estático o dinámico durante la ejecución.

- **Generación:** Es una iteración del algoritmo principal. Se trata de la creación de una nueva población mediante operadores genéticos y un mecanismo de selección.
- **Aptitud:** Es un valor que cuantifica la calidad de un individuo como solución potencial al problema que se busca resolver. Es uno de los elementos más importantes de los algoritmos evolutivos, ya que permite determinar si un individuo es deseable o no. En el primer caso el individuo tendrá una mayor probabilidad de sobrevivir o de tener descendencia; en caso contrario tendrá una mayor probabilidad de ser eliminado de la población.
- **Operadores genéticos:** Son los mecanismos utilizados para modificar el material genético de los individuos. Es mediante estos operadores que se pueden explorar distintas alternativas del espacio de soluciones del problema.
  - **Cruza:** Es un operador que forma un nuevo individuo hijo a partir de la información de dos o más individuos padres.
  - **Mutación:** Es un operador que forma un nuevo individuo mediante alteraciones, (usualmente pequeñas) a la información de otro individuo.
- **Selección:** Es el proceso que determina cuáles individuos de la población se reproducirán mediante el operador de cruce. Generalmente utiliza el valor de aptitud de cada individuo y algún componente estocástico para realizar dicha selección.

Dependiendo de la técnica que se esté utilizando, se enfatiza el uso de un operador genético en mayor o menor medida.

## 2.2 | PARADIGMAS DE CÓMPUTO EVOLUTIVO

Existen tres paradigmas principales que conforman el área de computación evolutiva. Las diferencias principales entre éstos son los operadores genéticos utilizados, la forma en que éstos se aplican en cada generación y los parámetros que requieren. A continuación describimos las características principales de cada paradigma.

### 2.2.1 ESTRATEGIAS EVOLUTIVAS

Las Estrategias Evolutivas (EEs) son una técnica desarrollada en 1964 en la Universidad Técnica de Berlín [4]. En su versión original el operador genético principal es la mutación, mientras que la cruce es un operador secundario.

De forma general, una EE se denota mediante la siguiente notación

$$(\mu/\rho^+; \lambda) - EE,$$

donde  $\mu$ ,  $\rho$  y  $\lambda$  son valores enteros positivos. En cada generación se tiene una población de  $\mu$  individuos padres, de donde se eligen  $\rho$  individuos para aplicarles algún operador de recombinación. Dicho operador genera una población de  $\lambda$  individuos hijos, a los que se les calcula su valor de aptitud. Posteriormente se determina cuáles individuos serán considerados para pasar a la siguiente generación, para lo cual se utiliza algún operador de selección. Los dos operadores de selección comúnmente utilizados son:

- **Selección “+”** Indica que la población de padres de la siguiente generación se formará a partir de los mejores  $\mu$  individuos del conjunto conformado por la unión de los padres y los hijos. Esto permite que los padres que no son mejorados por sus descendientes pasen a la siguiente generación, ocupando el lugar de alguno de sus hijos. De esta forma si un individuo no es mejorado por sus descendientes puede existir en la población durante múltiples generaciones.
- **Selección “,”** Indica que la población de padres se formará a partir de los mejores  $\mu$  individuos del conjunto de  $\lambda$  hijos. En este caso se debe cumplir  $\lambda \geq \mu$ . Esto significa que únicamente los hijos son considerados para pasar a la siguiente generación. Por lo tanto, todos los individuos existirán en la población únicamente una generación.

En el algoritmo 1 se muestra el pseudocódigo de una EE genérica. Usualmente el valor  $\rho$  no es especificado, simplificando la notación a

$$(\mu^+; \lambda) - EE.$$

La primera EE que se desarrolló, la cual es también la más sencilla, se denota como  $(1 + 1) - EE$ . De acuerdo a la notación, dicha EE cuenta con una población de 2 individuos (1 padre y 1 hijo). En cada generación se genera un hijo aplicando un operador de mutación al padre. Después se calcula la aptitud del hijo resultante. Si la aptitud del hijo es mayor que

la del padre entonces el hijo reemplaza al padre. En caso contrario, el hijo es descartado. El operador de mutación aplicado es el siguiente: dado un padre  $\bar{x}^t = (x_1, x_2, \dots, x_n)^T$ , el hijo se genera usando

$$\bar{x}^{t+1} = \bar{x}^t + N(0, \bar{\sigma}),$$

donde  $t$  es el número de la generación actual, y  $N(0, \bar{\sigma})$  es un vector de números Gaussianos independientes con una media de 0 y desviación estándar  $\bar{\sigma}$ .

---

**Algoritmo 1** Estrategia Evolutiva  $(\mu/\rho^+, \lambda)$  genérica

---

- 1: **dados**  $n, \rho, \mu, \lambda \in \mathbb{N}_+$
  - 2: **inicializar**  $\mathcal{M} \leftarrow$  población de  $\mu$  individuos generados aleatoriamente
  - 3: **mientras** condición de paro no satisfecha **hacer**
  - 4:    $\mathcal{P} \leftarrow$  seleccionar\_padres( $\rho, \mathcal{M}$ )
  - 5:   **para**  $k \in \{1, \dots, \lambda\}$  **hacer**
  - 6:      $\mathbf{x}_k \leftarrow$  recombinar(seleccionar\_parejas( $\mathcal{P}$ ))  $\in \mathbb{R}^n$
  - 7:      $\mathbf{x}_k \leftarrow$  mutar( $\mathbf{x}_k$ )  $\in \mathbb{R}^n$
  - 8:      $f(\mathbf{x}_k) \leftarrow$  aptitud( $\mathbf{x}_k$ )
  - 9:    $\mathcal{M} \leftarrow \mathcal{M} \cup \{(\mathbf{x}_k, f(\mathbf{x}_k)) \mid 1 \leq k \leq \lambda\}$
  - 10:  $\mathcal{M} \leftarrow$  seleccionar\_μ\_mejores( $\mu, \mathcal{P}$ )
- 

Las EE se han utilizado para resolver distintos problemas en áreas como redes, bioquímica, óptica, diseño en ingeniería, magnetismo, entre otras.

### 2.2.2 ALGORITMOS GENÉTICOS

Los Algoritmos Genéticos (AGs) fueron propuestos por John H. Holland en 1962 [33]. Su principal uso es la solución de problemas de optimización, en donde existe una función matemática (llamada función objetivo) que se busca maximizar o minimizar. En estos algoritmos la analogía del proceso evolutivo se extiende aún más que en las EEs, ya que se intentan emular más componentes biológicos. Considerando que cada individuo emula a un organismo vivo, existen las siguientes analogías:

- **Fenotipo:** Son los valores del individuo que son evaluables en la función objetivo. Por ejemplo, dada una función que recibe 2 variables reales en el intervalo  $[0, 1]$ , el fenotipo de un individuo podría ser  $(0.25, 0.75)$ .

- **Genotipo:** Son los valores del fenotipo representados mediante alguna codificación, tradicionalmente binaria. Por ejemplo, los valores anteriores se podrían codificar con las cadenas (00100101, 01110101).
- **Cromosoma:** Es la cadena que codifica todas las variables del individuo. En el ejemplo mostrado, el cromosoma de dicho individuo sería la cadena binaria 0010010101110101.
- **Gene:** Es la cadena que codifica cada una de las variables del individuo. En el ejemplo, el cromosoma está compuesto de dos genes 00100101 y 01110101.
- **Alelo:** Son los posibles valores de cada posición. En una codificación binaria, cada posición únicamente puede tener los valores 0 ó 1.

El funcionamiento de los AG normalmente consiste en tres pasos:

1. Generar una población inicial de individuos
2. Seleccionar estocásticamente los valores que se reproducirán. A pesar de tener un elemento de aleatoriedad, el mecanismo utiliza el valor de optimalidad de cada solución (aptitud).
3. Generar una nueva población a partir de los individuos seleccionados en el punto anterior.

Los pasos 2 y 3 se repiten de forma iterativa hasta que se cumple algún criterio de paro. Los criterios más comunes son: haber realizado un número establecido de iteraciones o haber alcanzado determinado umbral de aptitud. En el algoritmo 2 se presenta el pseudocódigo de un AG genérico.

---

**Algoritmo 2** Algoritmo Genético genérico

---

- 1: **dados**  $n$  (*Tamaño de población*),  $g_{max}$  (*Número máximo de generaciones*)  $\in \mathbb{N}_+$
  - 2:  $t \leftarrow 0$
  - 3: **inicializar**  $\mathcal{G}_0 \leftarrow$  *población de  $n$  individuos generados aleatoriamente*
  - 4: **mientras** *condición de paro no satisfecha* o  $t < g_{max}$  **hacer**
  - 5:      $\bar{f} \leftarrow \text{aptitud}(\mathcal{G}_t)$
  - 6:      $\mathcal{G}_{selec} \leftarrow \text{seleccionar\_padres}(\bar{f}, \mathcal{G}_t)$
  - 7:      $\mathcal{G}_{t+1} \leftarrow \text{operadores\_genéticos}(\mathcal{G}_{selec})$
  - 8:      $t \leftarrow t + 1$
-

### 2.2.3 PROGRAMACIÓN EVOLUTIVA

La Programación Evolutiva (PE) fue propuesta por Fogel, Owens y Walsh en los 60s [22]. Originalmente se propuso para resolver problemas de predicción. Para ello se proporciona al algoritmo un conjunto de entrenamiento, compuesto por una serie de valores de entrada y de salida. Luego, la PE busca encontrar un programa que para cada valor de entrada, produzca el mismo valor de salida que el que se tiene en el conjunto de entrenamiento. Las principales diferencias con los paradigmas anteriores son dos:

1. Los individuos son representados como máquinas de estados finitos.
2. El operador genético principal es la mutación.

Una máquina de estados finitos es un modelo del comportamiento de un sistema, en donde dada una señal de entrada, se obtiene una señal de salida. Dicho modelo se compone de un alfabeto finito, un conjunto de estados finito, un estado inicial, una función de transición y un conjunto de estados finales.

La mutación realizada en la PE permite variar distintos aspectos de los individuos. Por ejemplo se puede cambiar el número de estados, el estado inicial, las transiciones de un estado a otro, o el símbolo de salida. Generalmente, se conserva a la mejor mitad de la población, en cada generación, y la otra mitad es reemplazada por mutaciones de la mejor mitad.

## 2.3 | PROGRAMACIÓN GENÉTICA

La Programación Genética (PG) es una técnica que busca resolver problemas sin requerir que el usuario conozca, o especifique de antemano, la estructura de la solución que se está buscando. A diferencia de la PE, la representación de las soluciones en PG suelen ser estructuras de datos que codifican un programa, o estructuras de algún lenguaje de programación.

Una de las primeras propuestas de evolucionar código ejecutable fue realizada en [13], quien utilizó una estructura de árbol para representar un programa, y propuso un operador de cruza entre árboles. Posteriormente se propusieron implementaciones que usan programas en LISP o PROLOG. No obstante, fue John R. Koza quien reconoció la importancia de la técnica en [47], donde presentó evidencia de su posible uso en distintas áreas. Finalmente la técnica se vio consolidada con la publicación de [48], a partir de la cual el crecimiento de la PG se disparó.

El funcionamiento básico de la PG es muy similar al de un algoritmo genético:

1. Generar de forma aleatoria una población inicial de individuos.
2. Utilizar una función de aptitud para medir el desempeño de cada individuo como una solución potencial del problema objetivo.
3. Seleccionar individuos de la población mediante algún mecanismo de selección. Después, realizar operaciones genéticas sobre los individuos seleccionados y almacenar el resultado en una nueva población.
4. Reemplazar la población existente con la nueva población y repetir los pasos 2-4 mientras no se cumpla algún criterio de paro.
5. Presentar al mejor individuo de la población como la salida del algoritmo.

Sin embargo, debido a la naturaleza de los individuos, existen diferencias significativas en los mecanismos y operadores utilizados por la PG con los de los AG o las EE. A continuación se describen algunos componentes y mecanismos particulares de la PG.

### 2.3.1 TERMINALES Y FUNCIONES

Los conjuntos de terminales y funciones definen el espacio de búsqueda de un algoritmo de PG. Las terminales proveen valores al sistema (generalmente numéricos), mientras que las funciones procesan valores ya almacenados en el sistema.

El **conjunto de terminales** está compuesto por tres tipos de valores:

1. Las entradas que recibirá el algoritmo, generalmente son variables.
2. Las constantes, generalmente definidas dentro de algún intervalo previamente especificado.
3. Funciones que no reciben argumentos, pero que tienen algún efecto secundario.

Se les llama valores terminales porque, en una representación de árbol, siempre terminan una rama.

El **conjunto de funciones** está compuesto por operadores y funciones que el algoritmo de PG puede ejecutar. Los elementos que componen a este conjunto pueden ser:

- Funciones lógicas (AND, OR, NOT, XOR), aritméticas (+, -, \*, /), trascendentes (*sen()*, *cos()*, *ln()*), etc.
- Estructuras condicionales (*if...then*, *switch...case*), de ciclo (*while...do*, *for...do*), de transferencia de control (*go to*, *call*), etc.
- Subrutinas de cualquier tipo, diseñadas para un problema específico.

Debido a que la PG explorará múltiples combinaciones entre los conjuntos de terminales y de funciones, es necesario considerar que cualquier combinación posible puede ser evaluada en algún punto de la ejecución del programa. Por ello, el conjunto de funciones debe cumplir con la propiedad de cerradura en los siguientes dos sentidos:

1. Consistencia de argumentos: Todas las funciones deben recibir el mismo número de argumentos, con el mismo tipo. En ocasiones basta con hacer una conversión de un tipo a otro, mientras que en otras hace falta usar alguna técnica más sofisticada.
2. Evaluación segura: Todas las funciones deben ser capaces de procesar todos los valores que puedan recibir. El ejemplo más común es el del operador aritmético de división, ya que no puede realizar una división entre el valor 0. Para manejar estos casos especiales es necesario modificar las funciones para que devuelvan valores específicos, en lugar de errores.

### 2.3.2 REPRESENTACIÓN

El aspecto más característico de la PG es el uso de programas computacionales como individuos de la población. Consecuentemente, la representación elegida para almacenar cada programa impacta directamente en la implementación del algoritmo de PG. El uso de una representación determinada afecta el uso de memoria que hará el programa, además de afectar los operadores genéticos que manipulan a los individuos. Existen múltiples representaciones de programas utilizadas en PG, siendo las principales las estructuras de árbol, lineales o grafos. En este trabajo describimos algunas características de las representaciones de árbol y lineales, ya que son las dos más comúnmente utilizadas.

La **representación de árbol** es la más común, ya que fue la estructura empleada en la década de los 90s, cuando el área estaba en proceso de consolidación. En esta representación se almacenan los elementos del conjunto de terminales y de funciones en nodos, los cuales se encuentran conectados en una estructura de árbol. En la figura 2.1 se muestra un ejemplo de un árbol que codifica la expresión  $[3 * (7 - b)] / \text{máx}(5, 1 + c)$ . En esta representación, cada una de las hojas del árbol contiene siempre un valor del conjunto de terminales, mientras que los nodos internos contienen siempre elementos del conjunto de funciones.

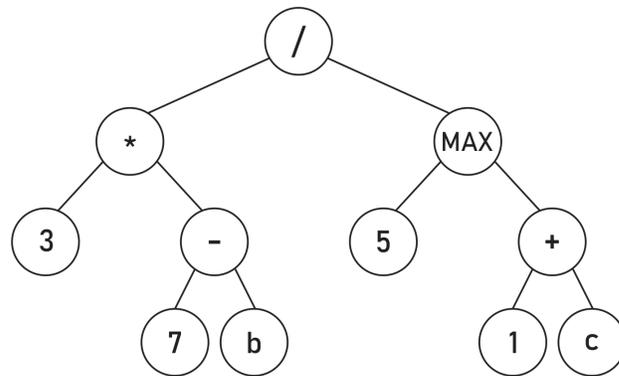


Figura 2.1: Programa almacenado en representación de árbol.

La **representación lineal** es una estructura que almacena una secuencia de instrucciones, las cuales se ejecutan una tras otra. Para su implementación, comúnmente se cuenta con un conjunto de registros a los que el algoritmo tiene acceso. En la figura 2.2 se muestra un ejemplo de un individuo que codifica la misma expresión presentada en la figura 2.1, pero utilizando una representación lineal con 4 registros.

La elección de determinada representación depende del programador, y de la naturaleza del problema a resolver. Algunos operadores son más fácilmente implementados en determinado lenguaje de programación, mientras que el manejo de memoria puede ser más eficiente en una estructura que en otra. De forma general, la estructura lineal es más rápida que la estructura de árbol, pero se encuentra más limitada en la capacidad de variación de los programas que puede representar.

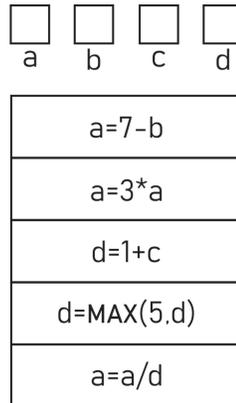


Figura 2.2: Programa almacenado en representación lineal.

### 2.3.3 INICIALIZACIÓN

La inicialización es el primer paso del algoritmo de PG. Consiste en crear de forma aleatoria una variedad de programas para después evolucionarlos. Generalmente se debe especificar un parámetro que indique el tamaño máximo de un individuo. Dependiendo de la representación utilizada se usan distintos algoritmos de inicialización y distintos parámetros de tamaño.

En la representación de árbol el parámetro puede ser el número máximo de nodos o la profundidad máxima. La profundidad de un nodo es el número mínimo de nodos que se deben atravesar desde el nodo raíz del árbol al nodo seleccionado. Los dos algoritmos más comunes para inicializar un programa en esta representación son los algoritmos *Grow* y *Full*:

- *Grow*: El valor de cada nodo se selecciona de forma aleatoria a partir de los conjuntos de funciones y de terminales establecidos. Las únicas excepciones son el nodo raíz (el cual es seleccionado exclusivamente del conjunto de funciones) y los nodos que alcanzan la profundidad máxima (los cuales son seleccionados exclusivamente del conjunto de terminales). Este método genera árboles con formas irregulares. En la figura 2.3 se muestra un programa inicializado usando esta técnica.
- *Full*: Para cada nodo interno se selecciona un elemento del conjunto de funciones. En el momento en que un nodo alcanza la profundidad máxima, se selecciona un elemento del conjunto de terminales. Este método genera árboles completos. En la figura 2.4 se muestra un programa inicializado usando esta técnica.

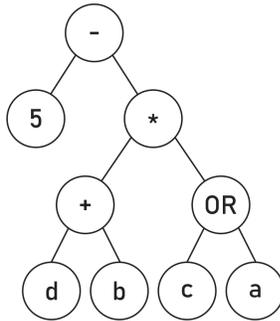


Figura 2.3: Árbol inicializado con el algoritmo Grow.

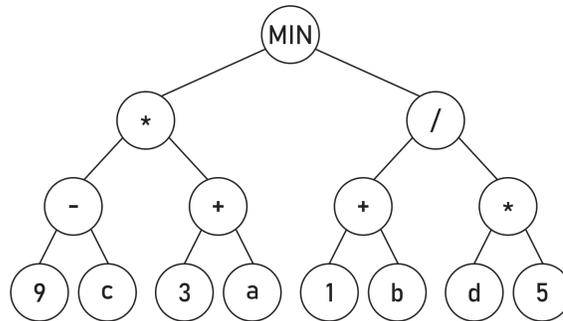


Figura 2.4: Árbol inicializado con el algoritmo Full.

Un elemento importante en todos los algoritmos evolutivos es la diversidad en las poblaciones de individuos. Al utilizar cualquiera de los dos métodos mencionados, la población tiende a contener individuos similares, o con poca diversidad. Buscando aumentar la diversidad de los individuos John Koza propuso el método *Ramped half-and-half*, el cual combina ambas técnicas: la mitad de la población se inicializa usando Full, y la otra mitad se inicializa usando Grow. Sin embargo, para garantizar una mayor diversidad, en cada mitad se utilizarán valores distintos del parámetro de tamaño máximo para cada individuo.

En el caso de la representación lineal, la inicialización suele ser más simple. El parámetro de tamaño puede ser el número máximo de instrucciones. Considerando que se cuenta con un conjunto de registros que representa los valores de entrada, un algoritmo simple para inicializar un individuo es el siguiente:

1. Elegir de forma aleatoria un valor menor o igual al número máximo de instrucciones.

2. Seleccionar un elemento del conjunto de funciones y completar sus argumentos eligiendo elementos del conjunto de registros o valores constantes.
3. Repetir el paso 2 hasta completar el número de instrucciones que se haya elegido en el paso 1.

#### 2.3.4 OPERADORES GENÉTICOS

Al igual que en el resto de los algoritmos evolutivos, los operadores genéticos son los mecanismos utilizados para transformar a los individuos de la población, explorando el espacio de soluciones del problema. En PG los tres operadores genéticos principales son tres: cruza, mutación y reproducción. Existen múltiples formas de implementar cada uno de estos operadores, aún usando una misma representación. A continuación se describen las versiones más comunes de cada operador.

##### 2.3.4.1 CRUZA

La cruza intercambia el material genético de dos individuos para crear uno o más descendientes. Es el operador genético principal en la PG. El paso previo a la cruza es la selección de los individuos padre. Para ello se utiliza algún mecanismo de selección, generalmente guiado por los valores de aptitud de los individuos de la población. Dependiendo de la representación utilizada, una vez seleccionados los dos individuos padres, se utilizan distintos algoritmos.

En la representación de árboles, se elige un punto de cruza de forma aleatoria en cada padre. Dicho punto de cruza es alguno de los nodos del árbol. Posteriormente, se intercambian los subárboles seleccionados entre los dos padres, generando dos individuos hijos. En la figura 2.5 se muestra un ejemplo de cruza entre dos árboles. Si bien se pueden almacenar los dos individuos generados por la cruza, se suele conservar únicamente a uno de ellos. Además, la selección de los puntos de cruza no se realiza con una probabilidad uniforme, ya que esto tiende a seleccionar hojas, o subárboles con pocos nodos, sesgando la capacidad de variación del operador. Por ello, se utiliza una mayor probabilidad de seleccionar nodos internos (funciones) y una menor probabilidad de seleccionar hojas (terminales). En [48], se propuso usar probabilidades de 90 % y 10 %, respectivamente.

En la representación lineal se elige, para cada padre, una secuencia de instrucciones de tamaño aleatorio. Posteriormente, se intercambian las se-

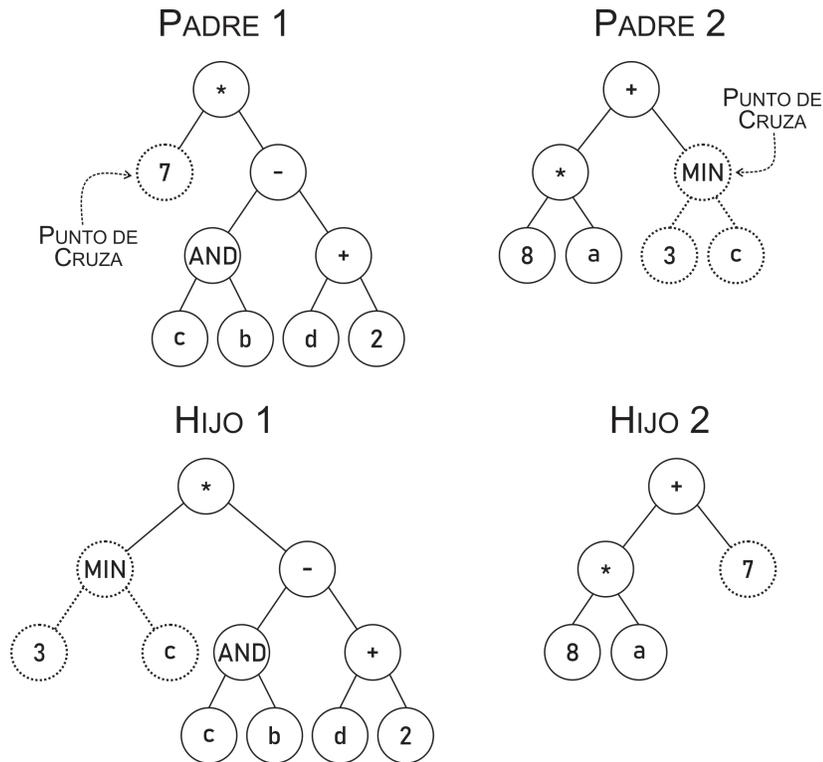


Figura 2.5: Cruza de dos programas en representación de árbol.

cuencias entre los dos padres, creando dos individuos hijos. En la figura 2.6 se muestra un ejemplo de cruza entre dos estructuras lineales.

### 2.3.4.2 MUTACIÓN

La mutación altera, de forma aleatoria, la composición del material genético de un individuo. Es un operador genético secundario en la PG.

En la representación de árboles, se elige un nodo del árbol de forma aleatoria. Posteriormente se reemplaza el subárbol enraizado en el nodo elegido, por un subárbol nuevo. Este subárbol es generado aleatoriamente, generalmente con el mismo mecanismo con el que se ha inicializado la población, y sujeto a las mismas restricciones de tamaño.

En estructuras lineales, se elige una instrucción del individuo de forma aleatoria. Después se generan uno o más cambios a dicha instrucción. Los cambios aplicables pueden ser de distintos tipos. Por ejemplo, se puede alterar cualquiera de los registros utilizados en la instrucción por algún otro, se puede alterar la función empleada en la instrucción, o el uso de una cons-

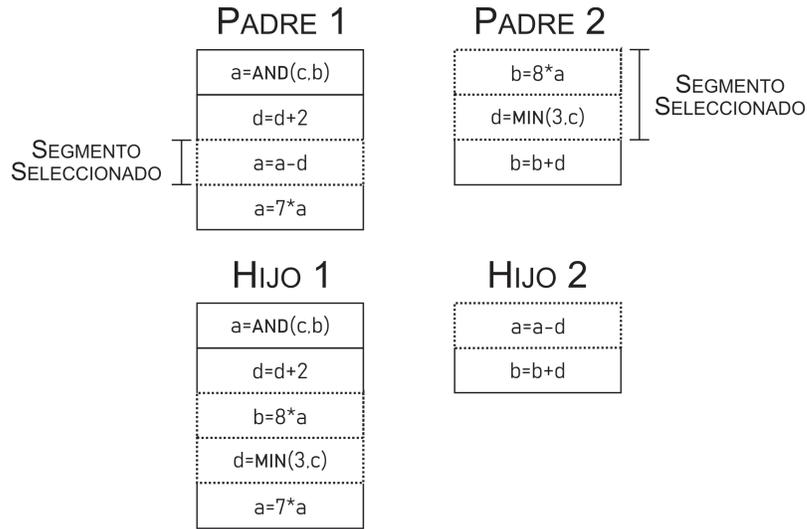


Figura 2.6: Cruza de dos programas en representación lineal.

tante por otra.

En las figuras 2.7 y 2.8 se muestra un ejemplo de mutación de un árbol y de una estructura lineal, respectivamente.

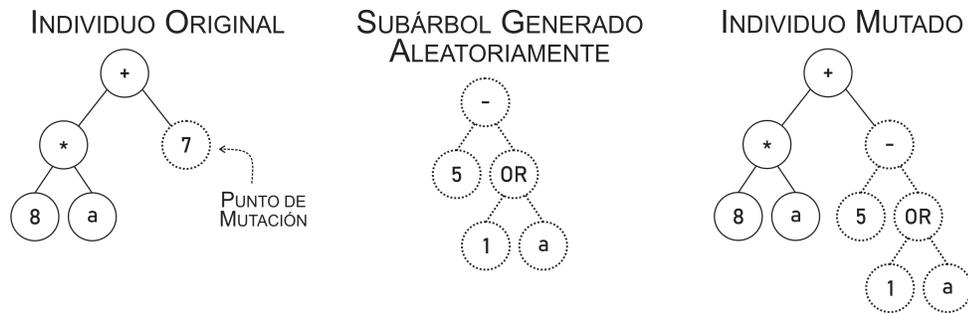


Figura 2.7: Mutación de un individuo en representación de árbol.

### 2.3.4.3 REPRODUCCIÓN

La reproducción es el operador genético más sencillo. Consiste en replicar un individuo de forma idéntica, sin alterar ninguna parte de su genoma.

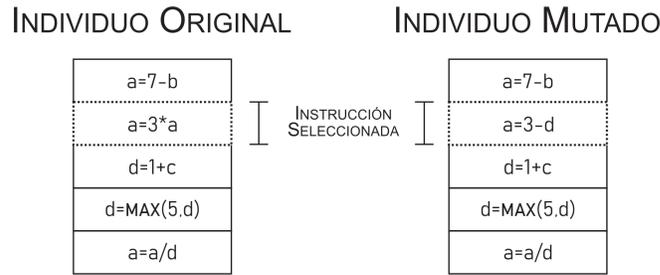


Figura 2.8: Mutación de un individuo en representación lineal.

La copia del individuo original se puede almacenar en la misma población, buscando aumentar sus probabilidades de reproducción. O se puede almacenar en la siguiente generación, garantizando su aparición en la descendencia.

## 2.4 | RESUMEN

En este capítulo presentamos un panorama general del área de cómputo evolutivo, así como los elementos comunes que comparten la mayoría de las técnicas desarrolladas en el área. Después, describimos las características principales de los 3 paradigmas más importantes del campo: Estrategias Evolutivas, Algoritmos Genéticos y Programación Evolutiva. Finalmente, presentamos de forma más detallada la técnica de Programación Genética, la cual es uno de los componentes principales de la implementación que hemos desarrollado como parte de este trabajo de tesis.



# PROBLEMAS DE OPTIMIZACIÓN MULTIOBJETIVO

En este capítulo se presenta un panorama general de los problemas de optimización, particularmente de su versión multi-objetivo. En la sección 3.1 se definen formalmente los problemas de optimización mono y multi-objetivo. En la sección 3.2 se introducen algunos conceptos básicos para definir la solución de dichos problemas así como otros elementos que se ocuparán más adelante. En la sección 3.3 se presentan algunas técnicas que existen para la solución de los problemas mencionados. En la sección 3.4 se mencionan algunas dificultades que pueden enfrentar dichas técnicas así como algunas de las posibles soluciones que se han propuesto para ellas. Finalmente, en la sección 3.5 se resume el contenido del capítulo.

## 3.1 | PROBLEMAS DE OPTIMIZACIÓN

Un problema de optimización es aquel en el que se busca encontrar la mejor solución (o el conjunto de las mejores soluciones), dado algún criterio, de una (o más) funciones objetivo. Los dos criterios comúnmente utilizados para determinar cuál solución es mejor son la maximización o minimización de uno (o más) valores de las funciones objetivo. En este trabajo, el criterio que utilizamos es el de minimización. Una forma de clasificar los problemas de optimización es en función del número de funciones objetivo que se busca minimizar. Usando este criterio se obtienen dos clases de problemas: aquellos que optimizan una función objetivo (mono-objetivo) y aquellos que optimizan dos o más funciones objetivos (multi-objetivo). A

continuación, se define formalmente los problemas de la primera clase.

**Definición 3.1.1 (Problema de optimización mono-objetivo).** *Un problema de optimización mono-objetivo con restricciones se define como:*

$$\begin{aligned} \min_{\vec{x}} \quad & f(\vec{x}), \\ \text{sujeto a} \quad & g_i(\vec{x}) \leq 0, \quad i = 1, \dots, m \\ & h_j(\vec{x}) = 0, \quad j = 1, \dots, p \end{aligned} \tag{3.1}$$

donde  $f : \mathbb{R}^n \mapsto \mathbb{R}$  es la función objetivo a minimizar,  $\vec{x} \in \mathbb{R}^n$  es el vector que contiene las variables de decisión,  $g_i(\vec{x}) \leq 0$  son las  $m$  restricciones de desigualdad y  $h_j(\vec{x}) = 0$  son las  $p$  restricciones de igualdad, con  $m \geq 0$  y  $p \geq 0$ .

La solución de (3.1) suele ser un único vector de variables  $\vec{x}$ , aunque también es posible obtener un conjunto de soluciones, las cuales optimicen a la función obteniendo el mismo valor objetivo  $f(\vec{x})$ , si el problema fuese multimodal.

Por otra parte, en el caso de que se busquen optimizar dos o más funciones objetivo simultáneamente se trata de un Problema de Optimización Multi-objetivo (POM). Este tipo de problemas se define a continuación.

**Definición 3.1.2 (Problema de optimización multi-objetivo).** *Un problema de optimización multi-objetivo con restricciones se define como:*

$$\begin{aligned} \min_{\vec{x}} \quad & \vec{F}(\vec{x}), \\ \text{sujeto a} \quad & g(\vec{x}) \leq 0 \\ & h(\vec{x}) = 0 \end{aligned} \tag{3.2}$$

donde  $\vec{F} : \mathbb{R}^n \mapsto \mathbb{R}^m$ ,  $\vec{F}(\vec{x}) = (f_1(\vec{x}), \dots, f_k(\vec{x}))^T$  es el vector de  $m$  funciones objetivo,  $f_i : \mathbb{R}^n \mapsto \mathbb{R}$  es la  $i$ -ésima función objetivo,  $\vec{x} \in \mathbb{R}^n$  es el vector que contiene las variables de decisión,  $g : \mathbb{R}^n \mapsto \mathbb{R}^q$  es el conjunto de  $q$  restricciones de desigualdad y  $h : \mathbb{R}^n \mapsto \mathbb{R}^p$  es el conjunto de  $p$  restricciones de igualdad, con  $m \geq 0$ ,  $p \geq 0$  y  $p < n$ .

Al vector  $\vec{x} = (x_1, \dots, x_n)^T$  se le conoce como *vector de decisión*, y debe pertenecer a la región definida por el conjunto de restricciones de igualdad y de desigualdad  $S \subset \mathbb{R}^n$ , llamada *región factible*. Dado un vector de decisión

$\vec{x}$ , se tiene un *vector objetivo*  $\vec{z}$ , el cual está compuesto por las imágenes del vector de decisión, es decir  $\vec{z} = \vec{F}(\vec{x}) = (f_1(\vec{x}), \dots, f_m(\vec{x}))$ . Además, la imagen de la región factible  $Z = \vec{F}(S)$  es llamada *región objetivo factible*.

A diferencia de la versión mono-objetivo, la solución de (3.2) suele ser un conjunto de soluciones. Esto se debe a que las funciones que componen al vector de funciones objetivo  $\vec{F}$  suelen estar en conflicto entre sí. Esto significa que la minimización de una de las funciones objetivo  $f_i$  causa un incremento en alguna otra función objetivo  $f_j, i \neq j$ .

Consecuentemente, es necesario definir un criterio para determinar cuándo un vector de decisión  $\vec{x}$  puede ser llamado una solución óptima. En la siguiente sección se define el criterio utilizado, llamado optimalidad de Pareto, así como algunos conceptos adicionales necesarios.

## 3.2 | CONCEPTOS BÁSICOS

La optimalidad de Pareto fue propuesta por Vilfredo Pareto en [60]. Este concepto permite calificar un vector  $\vec{x}$  como óptimo de Pareto si ninguno de los componentes  $z_i$  del vector objetivo correspondiente  $z = \vec{F}(\vec{x})$  se puede decrementar sin que se incremente al menos algún otro componente  $z_j, i \neq j$ . Formalmente se define como sigue:

**Definición 3.2.1 (Optimalidad de Pareto).** *Un vector de decisión  $\vec{x} \in S$  es un óptimo de Pareto si no existe otro vector de decisión  $\vec{y} \in S$  tal que*

$$\forall i \in \{1, \dots, n\}, f_i(y) \leq f_i(x) \quad \wedge \quad \exists j \in \{1, \dots, n\}, f_j(y) < f_j(x).$$

De forma estricta, decimos que un vector de decisión  $\vec{x}$  es una solución óptima de (3.2), si  $\vec{x}$  es un óptimo de Pareto. Para problemas no triviales, es decir, en los que las funciones objetivo se encuentren en conflicto, existe más de una solución óptima de (3.2), lo cual genera los siguientes dos conjuntos.

**Definición 3.2.2 (Conjunto de óptimos de Pareto).** *El conjunto de óptimos de Pareto  $\mathcal{P}_S$  se define como:*

$$\mathcal{P}_S := \{\vec{x} \in S \mid \vec{x} \text{ es óptimo de Pareto}\}.$$

**Definición 3.2.3 (Frente óptimo de Pareto).** *El frente óptimo de Pareto  $\mathcal{F}(\mathcal{P}_S)$  se define como:*

$$\mathcal{F}(\mathcal{P}_S) := \{F(\vec{x}) \mid \vec{x} \in \mathcal{P}_S\}.$$

Existe además otro tipo de optimalidad de Pareto que se considera menos estricto, definido a continuación.

**Definición 3.2.4 (Optimalidad débil de Pareto).** *Un vector de decisión  $\vec{x} \in S$  es un óptimo débil de Pareto si no existe otro vector de decisión  $\vec{y} \in S$  tal que*

$$\forall i \in \{1, \dots, n\}, f_i(y) < f_i(x).$$

A su vez, los óptimos débiles de Pareto definen los siguientes dos conjuntos en los espacios de decisión y objetivo.

**Definición 3.2.5 (Conjunto débilmente óptimo de Pareto).** *El conjunto débilmente óptimo de Pareto  $\mathcal{WP}_S$  se define como:*

$$\mathcal{WP}_S := \{\vec{x} \in S \mid \vec{x} \text{ es óptimo débil de Pareto}\}.$$

**Definición 3.2.6 (Frente débilmente óptimo de Pareto).** *El frente débilmente óptimo de Pareto  $\mathcal{F}(\mathcal{P}_S)$  se define como:*

$$\mathcal{F}(\mathcal{WP}_S) := \{F(\vec{x}) \mid \vec{x} \in \mathcal{WP}_S\}.$$

Decimos que la optimalidad débil de Pareto es menos estricta que la optimalidad de Pareto ya que el conjunto de óptimos de Pareto  $\mathcal{P}_S$  es un subconjunto de los óptimos débiles de Pareto  $\mathcal{WP}_S$ .

Una vez definidas las soluciones de un problema de optimización multiobjetivo es importante definir puntos de referencia en el espacio objetivo. Existen tres puntos de referencia principales, los cuales son usados por distintas técnicas de solución para acotar el espacio de búsqueda.

**Definición 3.2.7 (Vector objetivo ideal).** *El vector objetivo ideal  $\vec{z}^* \in Z$  está definido como:*

$$z_i^* = \min_{\vec{x} \in S} f_i(\vec{x}) \quad , \quad \forall i \in \{1, \dots, n\}.$$

**Definición 3.2.8 (Vector objetivo utópico).** *El vector objetivo utópico  $\vec{z}^{**} \in Z$  está definido como:*

$$z_i^{**} = \vec{z}_i^* - \epsilon \quad , \quad \forall i \in \{1, \dots, n\},$$

donde  $\epsilon \in \mathbb{R}^+$ .

**Definición 3.2.9 (Vector objetivo de nadir).** *El vector objetivo de nadir  $\vec{z}^{nad} \in Z$  está definido como:*

$$z_i^{nad} = \max_{\vec{x} \in S} f_i(\vec{x}) \quad , \quad \forall i \in \{1, \dots, n\}.$$

Los primeros dos puntos de referencia (ideal y utópico) sirven para acotar inferiormente el espacio de búsqueda. Si bien el punto de referencia ideal  $\vec{z}^*$  contiene los mínimos de cada función de forma independiente, en un problema no trivial siempre será una solución no factible. De forma similar el punto de referencia utópico  $\vec{z}^{**}$  siempre será no factible, ya que mejora al punto ideal por el factor elegido  $\epsilon$ . En la práctica se suele utilizar  $\epsilon$  como un valor escalar positivo pequeño.

Por otro lado, el punto de referencia (nadir) se utiliza para acotar superiormente el espacio de búsqueda, considerando los peores valores posibles de cada función objetivo. A pesar de su utilidad en distintos algoritmos, no existe un procedimiento para calcular de forma exacta dicho punto para problemas no lineales, por lo que, de ser necesario, se suelen usar aproximaciones [17].

### 3.3 | TÉCNICAS DE SOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN MULTI-OBJETIVO

Como se describe en [57], el proceso de solución de un problema de optimización multi-objetivo consiste de tres etapas:

1. Formular el problema de optimización real como un modelo matemático, particularmente como un POM.
2. Resolver el modelo obtenido usando alguna técnica de solución de POMs. El resultado de dicha técnica debe consistir en una aproximación del conjunto de Pareto  $\mathcal{P}_S$  del problema y el frente de Pareto  $\mathcal{F}(\mathcal{P})_S$  correspondiente.
3. Elegir la solución  $\vec{x} \in \mathcal{P}_S$  que mejor resuelva el problema, dado un determinado contexto.

Si bien cada una de estas etapas es importante y requiere múltiples elementos para su ejecución, en este trabajo nos enfocamos únicamente en la segunda etapa. Por lo tanto, suponemos que se cuenta ya con un modelo matemático del problema a resolver, y por lo tanto, que basta con entregar una aproximación tanto de  $\mathcal{P}_S$  como de  $\mathcal{F}(\mathcal{P})_S$ . Dados estos conjuntos, se presupone que un tomador de decisiones humano, que cuente con información detallada del problema, será el encargado de valorar las soluciones generadas y elegirá la más apropiada.

Existen distintas clasificaciones de las técnicas de solución de POMs. En [53] se distinguen dos tipos en función del rol del tomador de decisiones en el proceso de solución. En una clase están los métodos llamados “sin preferencia” porque no toman en cuenta al tomador de decisiones en ninguna parte del proceso de solución. En otra clase se encuentran los métodos que sí toman en cuenta al tomador de decisiones. Estos últimos se clasifican en métodos “a priori”, “interactivos” y “a posteriori” dependiendo de si toman en cuenta al tomador de decisiones al principio, durante, o al finalizar el proceso de solución, respectivamente. Por otro lado, en [10] las técnicas se clasifican en tres categorías principales: técnicas enumerativas, deterministas y estocásticas.

### 3.3.1 TÉCNICAS DE SOLUCIÓN ENUMERATIVAS

Las técnicas enumerativas son las estrategias de solución más sencillas, pero también las más limitadas en cuanto al número de problemas que pueden resolver. Consisten en enumerar de forma exhaustiva cada solución posible y determinar cuáles son óptimas.

Si bien estas técnicas garantizan la obtención de una solución exacta, únicamente son viables cuando el espacio de búsqueda es finito, enumerable y considerablemente pequeño. De otra forma, el costo computacional de generar cada solución hace que estas técnicas sea ineficiente (cuando el espacio es demasiado grande) o incluso no viable (cuando el espacio es no finito).

### 3.3.2 TÉCNICAS DE SOLUCIÓN DETERMINISTAS

Son estrategias que guían la búsqueda de soluciones óptimas utilizando información del problema.

- *Algoritmos glotones.* Funcionan eligiendo la mejor solución disponible en cada iteración. Su mecanismo se basa en la suposición de que una solución óptima local formará parte de la solución óptima global. Sin embargo, cuando esto no ocurre, el algoritmo falla, devolviendo como solución un óptimo local.
- *Algoritmos de escalada simple.* Conocidos comúnmente como algoritmos *hill climbing*, realizan una búsqueda en la dirección más prometedora, a partir de una solución actual. Al igual que los algoritmos glotones, terminará su ejecución al no poder encontrar una solución que mejore

a la actual, siendo susceptible a encontrar óptimos locales. Otra característica que comparten ambas técnicas es que no guardan un registro de las soluciones que ya han explorado, sino que únicamente buscan mejorar la solución actual.

- *Algoritmos de ramificación y poda.* Estos algoritmos buscan acotar el espacio de búsqueda para lo cual requieren el uso de heurísticas o algoritmos de decisión que proporcionen información del problema. El funcionamiento básico del algoritmo es buscar “ramificar” soluciones que parecen prometedoras y “podar” aquellas que no lo son. Esto significa que se generan posibles soluciones a partir de las mejores soluciones actuales y cuando alguna solución deja de ser prometedora se decide no expandir soluciones a partir de ella.
- *Algoritmos de búsqueda en profundidad.* Estos algoritmos eligen una solución y expanden posibles soluciones a partir de él. Posteriormente, eligen alguna de las nuevas soluciones y repiten el procedimiento. Esto se lleva a cabo de forma iterativa hasta que no pueden seguir expandiendo una solución. En ese momento se reanuda la expansión de soluciones eligiendo al hermano del último nodo procesado. Esto se repite de forma recursiva hasta explorar todo el árbol.
- *Algoritmos de búsqueda en amplitud.* Estos algoritmos son similares a los de búsqueda en profundidad en el sentido de ir eligiendo una solución, generar nuevas soluciones y realizar la exploración del árbol producido de forma recursiva. Sin embargo, la principal diferencia es el orden de exploración del árbol. Los algoritmos de búsqueda en amplitud exploran el árbol una capa a la vez, es decir, todos los nodos con la misma profundidad. Tanto los algoritmos de búsqueda en profundidad como en amplitud se pueden considerar como algoritmos no informados, ya que realizan la búsqueda siempre en el mismo orden, sin importar la ubicación de la solución en el árbol.
- *Algoritmos de búsqueda el mejor primero.* Son variantes de búsqueda en las que se asigna un valor numérico a las soluciones actuales usando información heurística. Posteriormente, se utiliza el valor para asignar el orden de exploración de las soluciones; la mejor solución es la primera en ser examinada. Este proceso se lleva a cabo de manera recursiva.
- *Algoritmos basados en cálculo.* Estos algoritmos requieren que el dominio de las variables sea continuo para poder aplicar técnicas de cálculo

y así encontrar un valor óptimo [3].

Los algoritmos mencionados se han usado en la solución de distintos problemas de optimización. Sin embargo, hay una amplia gama de problemas reales que tienen una serie de características que los hace particularmente difíciles de resolver. Algunas ejemplos son problemas en donde no existe una solución global única, sino un conjunto de óptimos globales (problemas multimodales), problemas discontinuos o con muchas dimensiones. En este tipo de problemas las técnicas deterministas suelen quedarse estancadas en soluciones locales, o requerir información del dominio del problema que no está disponible. Es por ello que se han desarrollado técnicas estocásticas, las cuales no pueden garantizar que encuentran una solución óptima, pero suelen proporcionar buenas soluciones a problemas que son particularmente difíciles para los métodos deterministas.

### 3.3.3 TÉCNICAS DE SOLUCIÓN ESTOCÁSTICAS

Estas técnicas utilizan estrategias de búsqueda con elementos estocásticos, con el fin de realizar una mejor exploración del espacio de búsqueda, en comparación con las técnicas deterministas y enumerativas:

- *Búsqueda aleatoria*. Es la estrategia de búsqueda estocástica más simple. Consiste en evaluar un conjunto de soluciones elegidas de forma aleatoria.
- *Paseo aleatorio*. Es similar a una búsqueda aleatoria, con la excepción de que la siguiente solución explorada es elegida aleatoriamente a partir de las soluciones que se pueden generar de la última solución evaluada.
- *Recocido simulado*. Es un algoritmo que modela el proceso de enfriamiento de materiales conocido como recocido. Este proceso consiste en calentar un material (metal o líquido generalmente) hasta una temperatura determinada, después mantener la temperatura por un tiempo dado, y luego dejar enfriar el material gradualmente. El funcionamiento del recocido simulado es como sigue: Dada una solución actual, se elige de forma aleatoria otra solución que pueda ser generada desde la solución actual. Si la nueva solución mejora a la anterior, el algoritmo se mueve a la nueva solución. En caso contrario, el movimiento se realiza con alguna probabilidad  $p < 1$ . Dicha probabilidad decrementa exponencialmente con el tiempo [45].

- *Monte Carlo*. Estos algoritmos involucran simulaciones con eventos estocásticos. Funcionan realizando una búsqueda completamente aleatoria en donde las soluciones son seleccionadas sin tomar en cuenta ninguna solución elegida previamente [59].
- *Búsqueda Tabú*. Es una estrategia de búsqueda desarrollada para evitar el principal problema de los métodos deterministas: encontrar únicamente óptimos locales. Para ello almacena en un registro las soluciones previamente visitadas y las soluciones que se pueden alcanzar desde dichas soluciones. Esta información restringe la elección de las soluciones que se evaluarán después [25]. La búsqueda tabú es a menudo integrada con otros métodos de optimización.
- *Cómputo Evolutivo*. Es un término genérico para referirse a distintos métodos de búsqueda estocásticos que simulan el proceso evolutivo natural, tal como se ha descrito en el capítulo 2. De los tres paradigmas principales del cómputo evolutivo, los Algoritmos Genéticos son los que se han utilizado más ampliamente en la solución de POMs.

Existen además otras técnicas que se han desarrollado para la solución de POMs, tales como la Optimización mediante Cúmulos de Partículas [44], Algoritmos Culturales [63], Búsqueda Cooperativa [55, 65] o híbridos de dos o más técnicas.

### 3.4 | PROBLEMAS DE OPTIMIZACIÓN CON MUCHAS FUNCIONES OBJETIVO

Existen múltiples algoritmos para la solución de problemas de optimización que usan las técnicas mencionadas en la sección anterior. En este trabajo nos enfocamos particularmente en los algoritmos de Cómputo Evolutivo que resuelven POMs, los cuales se engloban en un área llamada Optimización Evolutiva Multiobjetivo (OEM). A este tipo de algoritmos se les conoce como Algoritmos Evolutivos Multi-objetivo (AEMOs). En las últimas décadas se han usado diversos AEMOs en la solución de una amplia gama de problemas reales [10, 16, 9] de forma exitosa. Cabe hacer notar que muchos de estos algoritmos funcionan con un enfoque basado en optimalidad de Pareto, es decir que durante la ejecución del algoritmo, la técnica de búsqueda trata siempre de encontrar soluciones que sean óptimos de Pareto. Si bien esto genera buenas soluciones en POMs con dos o tres funciones objetivo, se ha demostrado que su uso no es satisfactorio en problemas con

más de tres funciones objetivo [20]. A esta subárea que estudia la solución de problemas de optimización con cuatro o más funciones objetivo se le conoce como *many-objective optimization* (optimización de muchas funciones objetivo).

En [42] se plantean los siguientes tres problemas que enfrentan los AEMOs en la solución de problemas con muchas funciones objetivo:

1. **Incremento exponencial del número de soluciones requeridas para la aproximación del frente de Pareto.** La dimensión del espacio objetivo es directamente proporcional al número de objetivos que se busca optimizar. Por otro lado, el número de puntos contenidos en el espacio aumenta de forma exponencial conforme se aumenta el número de dimensiones. El objetivo de los AEMOs es entregar un conjunto de soluciones que aproxime el frente de Pareto, el cual es una hipersuperficie en el espacio objetivo. Por lo tanto, el número de soluciones que se requieren para aproximar esta hipersuperficie se incrementa de forma exponencial conforme aumenta el número de dimensiones.
2. **Deterioro de la capacidad de búsqueda.** En un AEMO basado en optimalidad de Pareto la presión de selección es determinada por el número de soluciones que dominan a otras. De esta forma, si hay un mayor número de soluciones que sean óptimos de Pareto, la presión de selección hacia el verdadero frente será mayor que si no hay tantas soluciones óptimas en la población. Por la misma razón mencionada en el punto anterior, en un problema con muchas funciones objetivo el número de soluciones no dominadas es mayor, provocando que la presión de selección no aumente de la misma forma que lo haría en un problema con dos o tres objetivos. Esto se ha discutido en [37, 36] donde se realizaron pruebas con el algoritmo NSGA-II y se demostró que existe un deterioro en la capacidad de búsqueda conforme se aumenta el número de dimensiones.
3. **Dificultad en la visualización de las soluciones.** Una vez devuelto el conjunto de soluciones, se espera que sea un tomador de decisiones quien elija la solución adecuada basado en su preferencia. Sin embargo, al incrementar el número de dimensiones se vuelve complicado visualizar las soluciones de manera intuitiva.

Se han propuesto distintas alternativas para solucionar cada uno de estos problemas. En el caso del incremento de soluciones requeridas, se ha

propuesto implementar el uso de información de preferencia en los AEMOs [18, 68]. Dicha información se utiliza para concentrarse en una región más pequeña del frente de Pareto en donde el AEMO pueda encontrar soluciones óptimas. Por otro lado, para resolver el deterioro de la capacidad de búsqueda, se ha propuesto modificar el enfoque de dominancia de Pareto, para lograr encontrar soluciones no dominadas en cada iteración [66]. Otra propuesta es el uso de rangos de soluciones no dominadas [69, 49, 12]. Finalmente, se han desarrollado técnicas de visualización de soluciones no dominadas para facilitar la toma de decisiones en este tipo de problemas [58, 2, 46].

Otra alternativa de solución para el segundo problema es utilizar AEMOs basados en descomposición. Estos algoritmos descomponen un POM en un conjunto de subproblemas de optimización mono-objetivo. Para llevar a cabo esta descomposición, generalmente se usa un tipo de funciones llamadas funciones de escalarización y un vector de pesos el cual determina distintos niveles de preferencia en cada subproblema generado. El algoritmo más conocido dentro de este tipo de AEMOs es el MOEA/D [74], cuyo código fuente se encuentra disponible para distintas plataformas en <https://cswwww.essex.ac.uk/staff/zhang/IntrotoResearch/MOEAd.htm>.

### 3.5 | RESUMEN

En este capítulo presentamos las definiciones y conceptos básicos de la optimización multi-objetivo. En primer lugar definimos formalmente los problemas de optimización mono-objetivo y multi-objetivo. Posteriormente definimos los conceptos de optimalidad de Pareto, optimalidad débil de Pareto y los conjuntos que cada uno de ellos genera respectivamente. Estas definiciones nos permiten enunciar formalmente cómo es la solución de un problema de optimización multi-objetivo. Además, se definieron tres puntos de referencia importantes: el vector objetivo ideal, el vector objetivo utópico y el vector objetivo de nadir. Estos vectores son utilizados por distintas técnicas de solución, pero cobrarán particular importancia en el siguiente capítulo. Después presentamos el proceso de solución de un POM así como dos clasificaciones de las técnicas existentes. La primera clasificación se basa en el modo en el que el tomador de decisiones es involucrado en el proceso de optimización, mientras que la segunda se enfoca más en el proceso en que se guía la búsqueda. A partir de esta clasificación presentamos una breve descripción de algunas técnicas más comúnmente utilizadas. Posteriormente, presentamos algunos problemas que enfrentan las técnicas

que utilizan un enfoque basado en la optimalidad de Pareto, particularmente en la solución de POMs con más de tres funciones objetivo. Finalmente explicamos algunas alternativas de solución para este tipo de POMs y mencionamos la técnica de descomposición como una alternativa posible. En el siguiente capítulo hacemos un análisis de dicha técnica, y de las funciones de escalarización que permiten su implementación.



---

## FUNCIONES DE ESCALARIZACIÓN

En este capítulo se presenta un panorama general del uso de la técnica de solución de Problemas de Optimización Multi-objetivo (POM) basada en funciones de escalarización. En la sección 4.1 se describe el mecanismo mediante el cual se puede resolver un POM utilizando este tipo de funciones. Posteriormente, en la sección 4.2 se presenta una clasificación de las distintas funciones de escalarización existentes basada en el tipo de información requerida. Después, en la sección 4.3 se definen formalmente las funciones de escalarización más comúnmente utilizadas en el área. Finalmente, en la sección 4.4 se presenta un resumen del capítulo.

### 4.1 | SOLUCIÓN DE POMS MEDIANTE DESCOMPOSICIÓN

Una de las formas más naturales de intentar resolver un POM es la de transformar un POM en varios problemas de optimización mono-objetivo cuya solución sea equivalente a resolver el POM original. A esta técnica se le conoce como descomposición, ya que consiste en descomponer el problema multi-objetivo en uno (o más) problemas de optimización mono-objetivo. Para llevar a cabo dicha descomposición se utilizan distintas funciones matemáticas, a las cuales se les conoce como funciones de escalarización, de agregación, o de utilidad.

Formalmente, una función de escalarización,  $f : \mathbb{R}^m \mapsto \mathbb{R}$  es un modelo que, tomando en cuenta determinada información de preferencia, combina las  $m$  funciones objetivo del POM en una única función. Dicha combinación se lleva a cabo manipulando las funciones objetivo con operadores matemáticos, normalmente aritméticos. Generalmente, la información de preferen-

cia permite especificar la importancia de cada función objetivo, asignándole un peso distinto en la función de escalarización. De esta forma, es posible implementar las preferencias del tomador de decisiones en el proceso de solución.

Una de las principales ventajas que presenta esta técnica es que, una vez descompuesto el problema multi-objetivo, se puede hacer uso de la extensa literatura existente sobre problemas de optimización mono-objetivo y los múltiples algoritmos de solución que existen para ellos.

Al aplicar funciones de escalarización se reduce la dimensión del espacio de búsqueda transformando el problema original en un subproblema que busca reflejar las preferencias del tomador de decisiones. Esto genera dos metodologías comúnmente utilizadas para la solución de un POM:

1. **Método *a priori*:** Suponiendo que el tomador de decisiones tiene algún tipo de información acerca del POM y su respectivo espacio objetivo, se le solicita que especifique la información de preferencia necesaria antes de efectuar la búsqueda. Una vez que se cuenta con esta información se simplifica el POM a una versión mono-objetivo, la cual es resuelta con alguna técnica previamente establecida. El resultado único se presenta al tomador de decisiones como la solución, esperando que refleje adecuadamente la información de preferencia proporcionada.
2. **Método *a posteriori*:** No se solicita información de preferencia al tomador de decisiones. En lugar de ello, se generan múltiples parámetros, generalmente en forma de vectores, que proporcionen distintas configuraciones de preferencias para cada objetivo. Posteriormente, se aplica el proceso de descomponer el POM tantas veces como vectores se hayan generado. Después, se resuelve cada subproblema mono-objetivo y se presenta al tomador de decisiones el conjunto de soluciones obtenidas para que sea él quien elija la que considere más conveniente.

Se puede debatir sobre cuál de los dos enfoques es el mejor, ya que se podría afirmar que el hecho de considerar la información sobre las preferencias proporcionada por el tomador de decisiones que cuenta con información específica del problema es mejor que generar determinísticamente un conjunto de valores independientemente del problema. Sin embargo, existe evidencia de que aún utilizando la misma información de preferencia en dos técnicas de descomposición distintas se obtienen soluciones distintas [6]. Por lo tanto, la información de referencia permite únicamente guiar

la dirección de búsqueda, pero no existe garantía de que se reflejará de forma precisa dicha información. Es por ello que la mayoría de los Algoritmos Evolutivos Multi-objetivo (AEMOs) basados en descomposición utilizan el segundo enfoque, proveyendo al tomador de decisiones un conjunto de soluciones.

## 4.2 | CLASIFICACIÓN DE FUNCIONES DE ESCALARIZACIÓN

Existen distintos tipos de funciones de escalarización. De igual forma, existen distintas clasificaciones de ellas. A continuación se presenta una clasificación en función de los dos tipos de información de preferencia que se requieren comúnmente [52].

### 4.2.1 FUNCIONES DE ESCALARIZACIÓN BASADAS EN CLASIFICACIÓN

Por su naturaleza, este tipo de funciones se aplican en métodos “interactivos”, en donde se resuelve el problema de forma iterativa. En cada una de las iteraciones se solicita al usuario, generalmente el tomador de decisiones, que especifique qué tipo de cambios son deseables realizar a la solución actual.

El proceso general consiste en tres pasos que se repiten de forma iterativa:

1. Se utiliza la función de escalarización para descomponer el POM y se resuelve el subproblema generado.
2. Se muestran al usuario los valores obtenidos en cada una de las funciones objetivo con la solución actual.
3. Se solicita al usuario que clasifique cualitativamente cada uno de estos valores, y por consecuencia, que clasifique cuáles funciones objetivo son más importantes.

Las clases más utilizadas comúnmente son tres: valores objetivo que se desean mejorar, valores que son aceptables, y valores que pueden empeorar. Adicionalmente, es posible que el usuario especifique cuantitativamente el nivel de mejora que busca en los valores de la primera clase, o el nivel máximo al que pueden llegar a empeorar los valores de la tercera clase. Finalmente, puede existir una clase adicional para valores que no se busquen clasificar, los cuales podrán variar libremente y son ignorados por la función de escalarización.

#### 4.2.2 FUNCIONES DE ESCALARIZACIÓN BASADAS EN PUNTOS DE REFERENCIA

Los puntos de referencia permiten a la función de escalarización jerarquizar la importancia de las funciones objetivo mediante puntos en el espacio de las funciones objetivo. Este tipo de funciones no requieren necesariamente que se proporcione información en cada paso del proceso de solución; basta con especificar los valores del punto de referencia *a priori*. Los puntos de referencia se pueden clasificar en tres tipos.

1. Puntos de aspiración: expresan los valores objetivo que se desean alcanzar. Generalmente no son valores factibles.
2. Puntos de reserva: expresan valores objetivo que son tolerables para el tomador de decisiones, y que mínimamente se deberían alcanzar.
3. Puntos de peor resultado: expresan valores objetivo que se deberían evitar.

De estos tres tipos, los puntos de referencia más comúnmente utilizados son los de aspiración. Por lo tanto, la diferencia principal respecto a las funciones de escalarización basadas en clasificación, es que en estas últimas se deben especificar algunos valores objetivo que puedan empeorarse, mientras que en las funciones basadas en puntos de referencia se pueden utilizar valores no factibles para cada función objetivo.

Adicionalmente, independientemente del tipo de función utilizada, es común proporcionar al tomador de decisiones los valores del vector objetivo ideal y una aproximación del vector objetivo de nadir, presentados en el capítulo anterior, los cuales se consideran puntos de referencia. Esto con el fin de dar al tomador de decisiones una perspectiva de las posibilidades y limitaciones de los valores de cada función objetivo.

En el presente trabajo se presentan funciones basadas en puntos de referencia. Sin embargo, estos puntos no son necesariamente los vectores objetivo ideal o de nadir, sino vectores de pesos que permitan indicar la dirección de búsqueda deseada.

### 4.3 | REVISIÓN DE FUNCIONES DE ESCALARIZACIÓN

A continuación se presenta una definición formal y algunas características de seis de las funciones de escalarización más utilizadas en el área y

algunas de sus variaciones. El tipo de información de preferencia que requieren todas las funciones presentadas es un vector de pesos  $\vec{w} \in \mathbb{R}^m$ , además del *vector de Nadir* en el caso de la función PBI.

- WEIGHTED SUM FUNCTION (WS)

Es una de las primeras técnicas propuestas para la solución de POMs [11], y por lo tanto una de las más utilizadas. Esta función permite combinar linealmente los objetivos, generando un problema de optimización escalar. Formalmente se define como:

$$WS(\vec{f}', \vec{w}) := \sum_i w_i f'_i. \quad (4.1)$$

El tipo de soluciones que se pueden obtener mediante WS depende del vector de pesos  $\vec{w}$ . Para  $w_i > 0$  y  $\sum_i w_i = 1$ ,  $i \in 1, \dots, m$ , esta función genera soluciones que son óptimos de Pareto. Sin embargo, para  $w_i \geq 0$ ,  $i \in 1, \dots, m$  únicamente se puede garantizar que las soluciones obtenidas sean óptimos débiles de Pareto [56].

Debido a su relativa simplicidad, requiere menor poder de cómputo que otras funciones de escalarización más complejas, siendo ésta su principal ventaja.

No obstante, la principal desventaja que presenta es que únicamente tiene un buen desempeño en POMs cuyos frentes de Pareto tienen una geometría convexa. En el caso de que la geometría sea cóncava la función únicamente puede encontrar los puntos extremos del frente, es decir los mínimos de cada función objetivo por separado.

Esta función se ha implementado en múltiples AEMOs [74, 38].

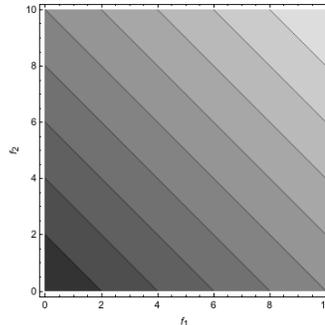


Figura 4.1: Curvas de nivel de la función WS con  $\vec{w} = (0.5 \ 0.5)^T$ .

■ CHEBYSHEV FUNCTION (CH)

Esta función minimiza el máximo valor encontrado por una combinación lineal de las funciones objetivo. Formalmente se define como:

$$CH(\vec{f}', \vec{w}) := \max_i (w_i |f'_i|). \quad (4.2)$$

A diferencia de WS, con esta función es posible obtener soluciones en porciones cóncavas del frente de Pareto. Sin embargo, únicamente se puede garantizar que las soluciones obtenidas sean óptimos débiles de Pareto. Además, debido a la presencia de la función  $\max$ , no es posible utilizar métodos de optimización basados en gradiente para minimizar esta función, ya que no es diferenciable [56].

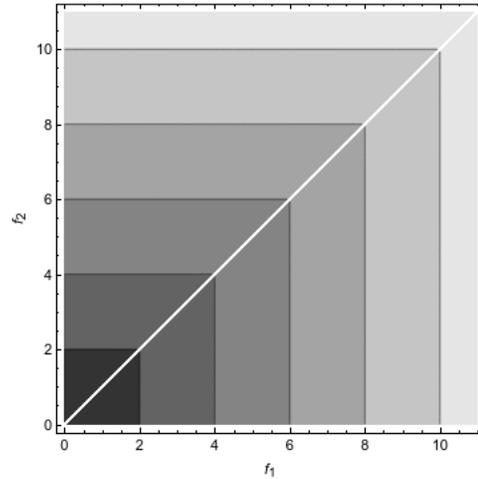


Figura 4.2: Curvas de nivel de la función CH con  $\vec{w} = (0.5 \quad 0.5)^T$ .

■ AUGMENTED CHEBYSHEV FUNCTION (ACH)

Esta función es una modificación de CH, en la cual se agrega un término adicional. Formalmente se define como:

$$ACH(\vec{f}', \vec{w}) := \max_i (w_i |f'_i|) + \alpha \sum_i |f'_i|. \quad (4.3)$$

El impacto del nuevo término se ajusta con un parámetro escalar  $\alpha$ , y su propósito principal es evitar las soluciones débiles de Pareto y garantizar la obtención de óptimos de Pareto. Esto se logra generando diferentes pendientes para cada función objetivo [56].

Un valor muy pequeño de  $\alpha$  puede provocar que el término adicional pierda su efecto y se generen óptimos débiles de Pareto, al igual que CH. Por otro lado, un valor demasiado grande de  $\alpha$  puede provocar que algunos puntos se vuelvan inalcanzables [62]. Esto indica que existe una alta sensibilidad a dicho parámetro [40], sin embargo, en la práctica se suelen utilizar valores relativamente pequeños (p.ej.  $\alpha = 10^{-6}$ ).

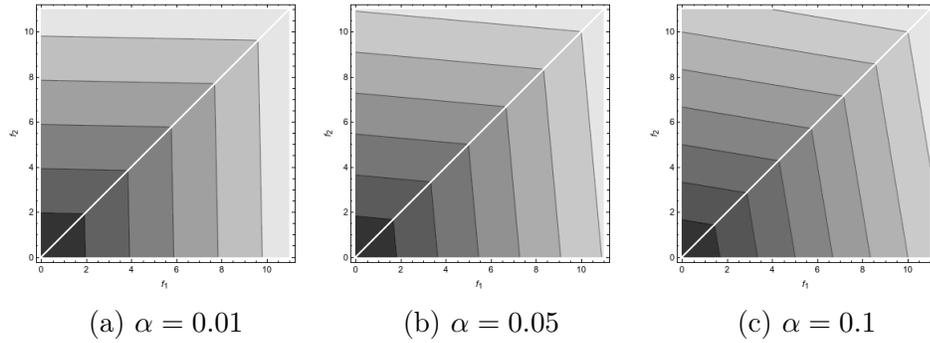


Figura 4.3: Curvas de nivel de la función ACH con  $\vec{w} = (0.5 \quad 0.5)^T$ .

■ ACHIEVEMENT SCALARIZING FUNCTION (ASF)

Similar a la función CH, se busca minimizar el valor máximo de una combinación lineal de los objetivos [24]. Sin embargo, la forma en la que se realiza la combinación lineal es distinta, ya que no se multiplica directamente por los pesos  $w_i$ , sino por sus valores recíprocos. Formalmente se define como:

$$ASF(\vec{f}^l, \vec{w}) := \max_i \left( \frac{f_i^l}{w_i} \right). \quad (4.4)$$

Esta función se ha utilizado en la solución de distintos AEMOs basados en descomposición como MOMBI-II, NSGA-III y MOEA/D-GRA.

En cuanto a las soluciones que se obtienen, ASF genera óptimos débiles de Pareto. Además, existe evidencia de que esta función tiende a tener una mejor dispersión del frente de Pareto que la función CH, la cual tiende a preferir soluciones en la rodilla y los extremos del frente [61].

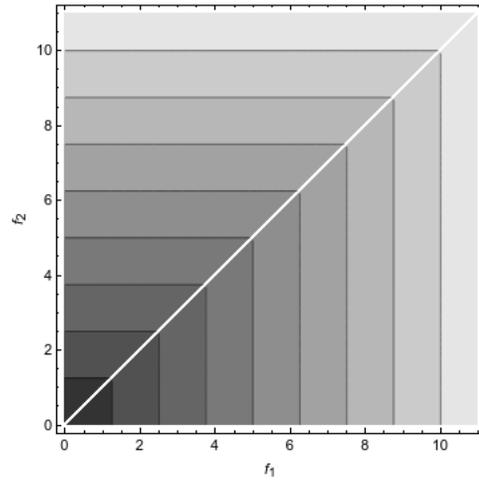


Figura 4.4: *Curvas de nivel de la función ASF con  $\vec{w} = (0.5 \ 0.5)^T$ .*

■ AUGMENTED ACHIEVEMENT SCALARIZING FUNCTION (AASF)

Similar a ACH, esta función utiliza un término adicional para garantizar que se obtienen óptimos de Pareto [53]. Formalmente se define como:

$$AASF(\vec{f}', \vec{w}) := \max_i \left( \frac{f'_i}{w_i} \right) + \alpha \sum_i \frac{f'_i}{w_i}. \quad (4.5)$$

En la práctica,  $\alpha$  toma valores relativamente pequeños (*p.ej.*  $\alpha = 10^{-4}$ ).

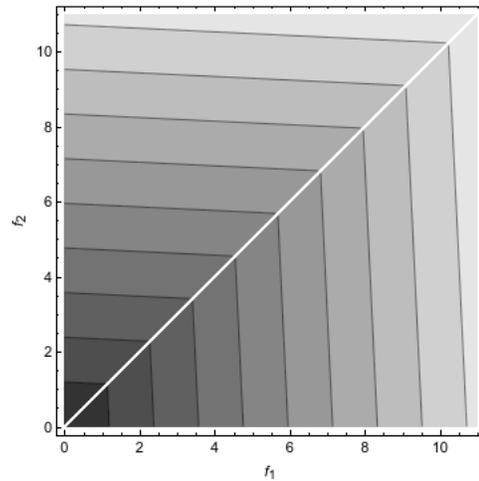


Figura 4.5: *Curvas de nivel de la función AASF con  $\vec{w} = (0.5 \ 0.5)^T$ .*

■ PENALTY BOUNDARY INTERSECTION FUNCTION (PBI)

A diferencia de las funciones anteriores, PBI utiliza el vector de Nadir como punto de referencia. Su funcionamiento es mediante un valor escalar de penalización  $\theta$  que permite balancear dos parámetros: la convergencia (medida por  $d_1$ ) y la diversidad (medida por  $d_2$ ). Se busca que ambos parámetros sean minimizados [64]. Formalmente se define como:

$$PBI(\vec{f}', \vec{w}) := \theta d_2 - d_1 \quad (4.6)$$

donde:

$$d_1 := \left| f' \cdot \frac{w}{\|w\|} \right|, \quad d_2 := \|f'' - d_1 \frac{w}{\|w\|}\|$$

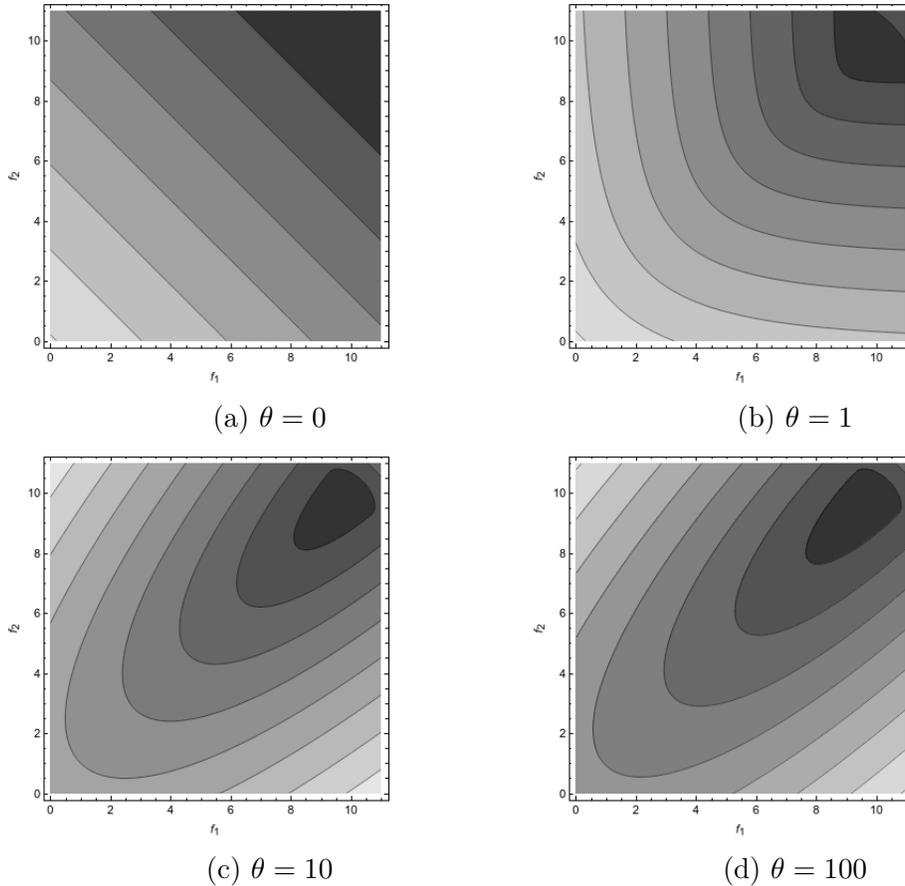


Figura 4.6: Curvas de nivel de la función PBI con  $\vec{w} = (0.5 \quad 0.5)^T$ .

## 4.4 | RESUMEN

En este capítulo presentamos el modo de funcionamiento de las funciones de escalarización en la solución de POMs, así como una taxonomía de éstas. Si bien existen múltiples tipos de funciones, con distintos tipos de información de preferencia, en este trabajo nos enfocamos en la generación de funciones de escalarización que utilizan vectores de pesos. Esto es debido a que son el tipo de funciones más comunes en los AEMOs basados en descomposición. Una vez establecidas las propiedades de las funciones de escalarización más conocidas, es posible conocer las características deseables para el diseño de una nueva función de escalarización. Y es precisamente este proceso de generar una nueva función el que se describe en los capítulos siguientes.



---

## IMPLEMENTACIÓN PROPUESTA

En este capítulo se describen las herramientas utilizadas para generar un programa que permita la creación de nuevas funciones de escalarización utilizando programación genética. Para ello se utilizaron dos programas los cuales fueron resultado de tesis doctorales. En la sección 5.1 se describen las características y funcionamiento del primer programa, que es una implementación de Programación Genética. Esta implementación fue la base del programa utilizado para diseñar la nueva función de escalarización. En la sección 5.2 se describen los componentes principales del AEMO utilizado para realizar la validación de las funciones generadas. En la sección 5.3 se explican las modificaciones realizadas a los programas mencionados para obtener la implementación propuesta en este trabajo, así como la información más relevante para poder realizar su ejecución. Finalmente en la sección 5.4 se detallan los parámetros utilizados en dicha implementación para generar dos nuevas funciones de escalarización de manera automatizada.

### 5.1 | PROGRAMACIÓN GENÉTICA

Existen diversas implementaciones de Programación Genética disponibles en distintos lenguajes de programación [23, 50, 51, 67, 71]. Los criterios seguidos para la elección de alguno de estos programas fueron principalmente dos: licencia de uso y lenguaje de implementación. En el primer caso se buscó software libre, ya que es necesario realizar modificaciones al código original para realizar la búsqueda de la función de escalarización. Por otro lado, se decidió priorizar una implementación en algún lenguaje de programación de bajo nivel, buscando obtener el menor tiempo de ejecu-

ción posible. Bajo estos dos criterios la implementación que se utilizó en este trabajo fue el algoritmo de programación genética lineal epigenética (ELGP, por sus siglas en inglés) reportado en [50].

Como se explicó en el capítulo 2, la Programación Genética es un algoritmo iterativo que utiliza una población de individuos, los cuales son manipulados mediante operadores genéticos para generar una nueva población. Cada individuo puede ser un programa o una función, codificados en estructuras de datos. Las características principales de ELGP son dos:

1. Utiliza una representación lineal de los individuos, la cual es procesada mediante estructuras de datos de tipo pila.
2. Implementa un algoritmo de búsqueda local inspirado en mecanismos biológicos llamados mecanismos epigenéticos.

A continuación explicamos con más detalle estas características y proporcionamos algunos ejemplos para ilustrarlas.

### 5.1.1 REPRESENTACIÓN DE LOS INDIVIDUOS

En ELGP se utiliza una representación lineal en notación posfija. La principal ventaja de esta representación es que se puede garantizar una validez sintáctica para secuencias arbitrarias de instrucciones. En contraste, las representaciones basadas en árbol pueden volverse sintácticamente inválidas al alterar nodos del árbol. Para realizar la ejecución de los programas se utiliza una estructura de tipo pila, la cual ayuda a garantizar la validez sintáctica mencionada. Para lograrlo, simplemente se ignora la ejecución de aquellos operadores con una aridad superior al tamaño actual de la pila. Además, se considera únicamente el último elemento en la pila al final de la ejecución. Estas dos reglas permiten que la ejecución de dos o más individuos con distintas instrucciones se ejecuten de la misma forma. Por ejemplo, dados los siguientes tres individuos  $i_1$ ,  $i_2$  y  $i_3$  con los genotipos mostrados en (5.1), se obtiene en los tres casos la misma ejecución, en este caso la instrucción  $(x_1 + x_2)$ .

$$\begin{aligned}
 i_1 &= [ x_1 \ x_2 \ + \ ] \Rightarrow (x_1 + x_2) \\
 i_2 &= [ x_1 \ x_2 \ + \ - \ * \ / \ ] \Rightarrow (x_1 + x_2) \\
 i_3 &= [ u \ + \ x_1 \ / \ x_1 \ x_2 \ + \ ] \Rightarrow (x_1 + x_2)
 \end{aligned} \tag{5.1}$$

En la ejecución de  $i_1$  se utilizan todos los elementos del genotipo. En el caso de  $i_2$  se ignoran los operadores  $-$ ,  $*$ , y  $/$  debido a que el tamaño de la pila es insuficiente. Finalmente, en el caso de  $i_3$  existen dos elementos en la pila al final de la ejecución, pero únicamente se considera el último. A continuación se presenta un ejemplo de la ejecución del individuo  $i_3$  para ilustrar el procedimiento.

$i_3 = [ u + x_1 / x_1 x_2 + ]$	
Procesamiento de elementos	Pila
1. $(u)$ : push $(u)$	$[u]$
2. $(+)$ : ignorar (argumentos insuficientes)	$[u]$
3. $(x_1)$ : push $(x_1)$	$[u x_1]$
4. $(/)$ : top $(x_1)$ ; top $(u)$ ; push $(u/x_1)$	$[(u/x_1)]$
5. $(x_1)$ : push $(x_1)$	$[(u/x_1) x_1]$
6. $(x_2)$ : push $(x_2)$	$[(u/x_1) x_1 x_2]$
7. $(+)$ : top $(x_2)$ ; top $(x_1)$ ; push $(x_1 + x_2)$	$[(u/x_1) (x_1 + x_2)]$
→ devolver último elemento de la pila	$(x_1 + x_2)$

### 5.1.2 BÚSQUEDA LOCAL INSPIRADA EN MECANISMOS EPIGENÉTICOS

La epigenética estudia los cambios hereditarios en la expresión genética que no se deben a alteraciones en la secuencia del ADN [34]. De forma análoga, ELGP busca mejorar el desempeño de la Programación Genética tradicional implementando mecanismos que permitan modificar la expresión de los individuos (es decir la secuencia de instrucciones ejecutada) sin modificar su genotipo.

Esto se logra agregando a cada elemento del genotipo un marcador que indica si dicho elemento será procesado o no. Nos referimos a esta secuencia de marcadores como el epigenoma del individuo. Por lo tanto, el procesamiento de cada individuo se lleva a cabo procesando únicamente los elementos indicados en el epigenoma e ignorando el resto. Esto es posible debido a las restricciones establecidas previamente en el proceso de ejecución de las instrucciones. De esta forma, alterando el epigenoma de un individuo es posible obtener distintas ejecuciones a partir de un mismo genotipo. ELGP utiliza estas variaciones del epigenoma para explorar localmente el espacio de búsqueda. Por ejemplo, el individuo  $i_3$  puede generar distintas ejecuciones mediante transformaciones epigenéticas, tales como:

$$\begin{aligned}
 i_3 \rightarrow i'_3 &= \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ u & + & x_1 & / & x_1 & x_2 & + \end{bmatrix} & \Rightarrow & (x_1 + x_2), \\
 i_3 \rightarrow i''_3 &= \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ u & + & x_1 & / & x_1 & x_2 & + \end{bmatrix} & \Rightarrow & (u/x_1), \\
 i_3 \rightarrow i'''_3 &= \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ u & + & x_1 & / & x_1 & x_2 & + \end{bmatrix} & \Rightarrow & (u/x_1 + x_2).
 \end{aligned} \tag{5.2}$$

Durante la inicialización de ELGP, los marcadores epigenéticos son inicializados de forma aleatoria en la población inicial con una probabilidad del 50% de estar activos. La mutación del epigenoma en las generaciones siguientes puede ser de distintas formas, de acuerdo a los parámetros que se establezcan en el programa.

ELGP permite configurar cuatro formas distintas de utilizar los marcadores epigenéticos:

1. Ep1M: El epigenoma de cada individuo es mutado junto con el genotipo en cada generación. Por lo tanto, las mutaciones epigenéticas forman parte del proceso normal de la Programación Genética.
2. EHC1: Se genera una mutación adicional del epigenoma del individuo, se procesa el genotipo considerando los nuevos marcadores epigenéticos, y se compara su aptitud con la del individuo con el epigenoma original. Finalmente, se conserva el mejor epigenoma.
3. EHC5: Se realiza el mismo procedimiento que EHC1 pero con 5 mutaciones del epigenoma.
4. EHC10: Se realiza el mismo procedimiento que EHC1 pero con 10 mutaciones del epigenoma.

Al comparar la aptitud de dos o más epigenomas se busca siempre que el programa resultante tenga un mejor valor de aptitud en la función objetivo. En caso de existir un empate en los valores de aptitud se elige el programa con menor complejidad. Ésta puede ser medida de distintas formas, pero en esta implementación se calcula asignando valores a los elementos

del genotipo en función del tipo de instrucción. La complejidad  $C_i$  de un programa  $i$  con un genotipo activo (es decir, una vez considerado el epigenoma),  $g_a = [g_{a1} \dots g_{al}]$  se define como:  $C_i = \sum_{q=1}^l c(g_{a_q})$ , donde el valor de cada elemento se determina de la siguiente forma:

$$c(g_a) = \begin{cases} 4 & : (g_a = \log) \vee (g_a = \exp) \\ 3 & : (g_a = \text{sen}) \vee (g_a = \text{cos}) \\ 2 & : (g_a = /) \vee (g_a = \sqrt{\phantom{x}}) \\ 1 & : g_a = \text{otro} \end{cases} \quad (5.3)$$

Por lo tanto, en caso de existir un empate en los valores de aptitud de dos epigenomas, se conserva siempre aquel con menor valor  $C_i$ , buscando priorizar programas con menor complejidad computacional.

## 5.2 | ALGORITMO EVOLUTIVO MULTI-OBJETIVO

Existe una gran variedad de Algoritmos Evolutivos Multi-objetivo (AEMOs) en la literatura. Sin embargo, para este trabajo se requiere utilizar específicamente un AEMO que utilice funciones de escalarización en el proceso de solución, dado que el propósito de este trabajo es encontrar alguna función de escalarización que sea diferente de las usadas tradicionalmente y que mejore el desempeño de los AEMOs. Por lo tanto, no se pretende realizar una comparación exhaustiva entre AEMOs, sino únicamente entre funciones de escalarización. Consecuentemente, basta con elegir algún AEMO de este tipo e implementarlo en el cálculo de aptitud de la Programación Genética. No obstante, se buscó utilizar el AEMO que mejor se adecuara a las necesidades del proyecto.

Dos de los AEMOs que utilizan funciones de escalarización más utilizados en la literatura especializada son el algoritmo evolutivo multi-objetivo basado en descomposición (MOEA/D, por sus siglas en inglés) [74] y el algoritmo genético basado en un ordenamiento no dominado en su tercera versión (NSGA-III, por sus siglas en inglés) [15, 43].

Existen múltiples variaciones del MOEA/D que implementan distintas funciones de escalarización y mecanismos de búsqueda, algunas de las cuales han logrado mejorar el desempeño de la propuesta original en determinados tipos de problemas [75, 72, 70, 39, 8].

Por otro lado, el NSGA-III es más reciente y también ampliamente conocido en la comunidad. Sin embargo, no existe un código disponible de parte

de sus autores que permita replicar los resultados obtenidos en los artículos donde el algoritmo fue presentado originalmente, ya que no se trata de software libre.

Otros dos AEMOs que utilizan funciones de escalarización son la metaheurística basada en el indicador R2 para muchos objetivos en su primera y segunda versión (MOMBI y MOMBI-II por sus siglas en inglés, respectivamente). La principal ventaja de estos algoritmos es que están basados en el indicador R2, compitiendo en desempeño con algoritmos que están basados en indicadores como el hipervolumen pero con un menor costo computacional. Además, se ha demostrado que logran mejorar los resultados de otros AEMOs, entre ellos el MOEA/D, en diversos problemas de optimización multi-objetivo [29, 31].

MOMBI-II fue el algoritmo elegido para este trabajo, debido a tres razones principales: tiene un desempeño competitivo con respecto a AEMOs basados tanto en descomposición como en indicadores, tiene un costo computacional bajo comparado con AEMOs que utilizan el hipervolumen y el código fuente del algoritmo está disponible para su uso (<https://www.cs.cinvestav.mx/~EVOCINV/software/MOMBI-II/MOMBI-II.html>).

En la figura 5.1 se muestra el funcionamiento general de MOMBI-II. El algoritmo utiliza dos puntos de referencia para realizar una normalización de los valores objetivo obtenidos. Además, se utiliza un enfoque de ordenamiento en rangos de la población basado en el indicador R2, a diferencia de los algoritmos basados en dominancia de Pareto. Una vez seleccionados los padres, utilizando el criterio basado en rangos para determinar la calidad de los individuos, se genera una nueva población. Cada individuo de esta nueva población es evaluado, normalizado y posteriormente clasificado en rangos como se ha mencionado previamente. Después se reduce la población eligiendo a los mejores individuos de la unión de la población anterior y la población actual. Finalmente, se actualizan los puntos de referencia utilizando información estadística de las poblaciones anteriores.

La función de escalarización utilizada de forma predeterminada por MOMBI-II es la función ASF, definida como:

$$ASF(\vec{f}', \vec{w}) := \max_i \left( \frac{f'_i}{w_i} \right). \quad (5.4)$$

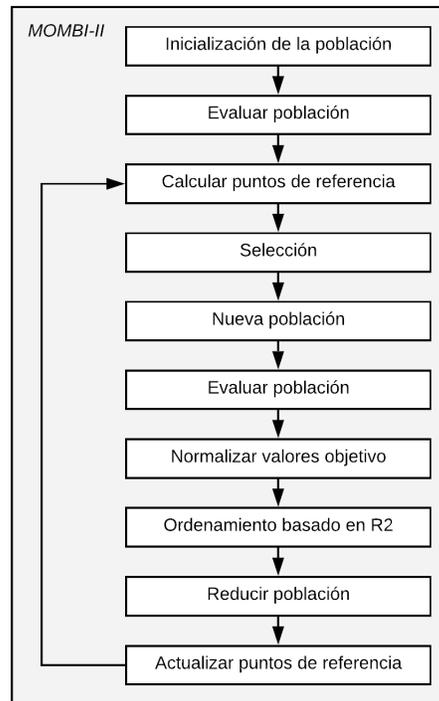


Figura 5.1: Diagrama del funcionamiento de MOMBI-II.

Sin embargo, en la implementación se encuentran además programadas las funciones *WSF*, *CHF*, *PBI*, *AASF* y *ACHF*, definidas en el capítulo anterior, por lo que es posible utilizar cualquiera de estas funciones en la ejecución del programa.

### 5.3 | IMPLEMENTACIÓN PROPUESTA

La implementación utilizada en este trabajo se compone de la unión de los dos algoritmos previamente descritos: ELGP y MOMBI-II. En la figura 5.2 se muestra el modo en que se combinan ambos algoritmos: el algoritmo base es una versión modificada de ELGP el cual ejecuta MOMBI-II en el cálculo de aptitud de los individuos.

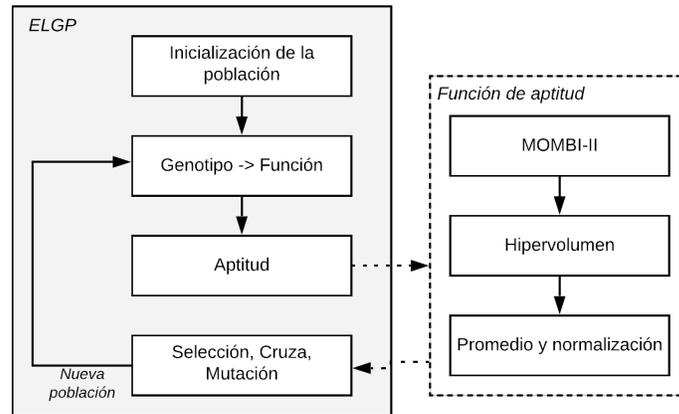


Figura 5.2: Diagrama del funcionamiento de la implementación propuesta para generar funciones de escalarización.

La modificación principal que se realizó al algoritmo de PG es la forma de calcular la aptitud de los programas. En el algoritmo original se busca encontrar la función matemática que mejor ajuste un conjunto de datos de entrenamiento, es decir que tenga menor error como regla de correspondencia entre dos o más conjuntos de valores. Sin embargo, en la versión modificada no se busca ajustar valores, sino encontrar una función de escalarización que resuelva un Problema de Optimización Multi-objetivo (POM). Para ello se programó una versión modificada del cálculo de aptitud, en la cual se ejecuta MOMBI-II pero proporcionándole la función de escalarización cuyo desempeño se quiere evaluar.

El pseudocódigo de la función utilizada para el cálculo de la aptitud se muestra en el algoritmo 3. Dada una función de escalarización, el algoritmo ejecuta una versión modificada de MOMBI-II, la cual utiliza dicha función en el proceso de solución de un POM. Una vez ejecutado MOMBI-II se calcula el hipervolumen del frente de Pareto obtenido y se almacena dicho valor. Cabe recordar que MOMBI-II es una metaheurística, por lo que su comportamiento no es determinista y el resultado que produce varía en cada ejecución. Por lo tanto, el procedimiento descrito se repite un número  $n$  de veces, especificado por el usuario, buscando que al promediar los  $n$  hipervolumenes obtenidos se obtenga una estimación del comportamiento de la función de escalarización. Este promedio es el que se utiliza como valor de aptitud de la función de escalarización; sin embargo, es necesario realizar un último ajuste. ELGP funciona minimizando el valor de aptitud, por

lo tanto, si utilizamos el valor de aptitud como el promedio de los hipervolumenes estaríamos dando preferencia a funciones que generan frentes con un menor valor de hipervolumen. Por lo tanto, se asigna como valor final de aptitud el valor  $HV_{max} - apt$ , donde  $HV_{max}$  equivale a una cota superior del hipervolumen establecida por el usuario.

---

**Algoritmo 3** Algoritmo para cálculo de aptitud

---

**Entrada:** *Función de escalarización  $f_i$ , Número de iteraciones  $n$ , Archivo de parámetros*

- 1:  $apt \leftarrow 0$
  - 2: **para**  $k \leftarrow 1$  hasta  $n$  **hacer**
  - 3:    $PF_i \leftarrow \text{MOMBI2}(f)$
  - 4:    $HV \leftarrow$  Calcular hipervolumen de  $PF_i$
  - 5:    $apt \leftarrow apt + HV$
  - 6:  $apt \leftarrow HV_{max} - apt/n$
  - 7: Devolver  $apt$
- 

El pseudocódigo del algoritmo principal se muestra en el algoritmo 4. El algoritmo comienza inicializando una población de individuos de forma aleatoria. Posteriormente se decodifican los genotipos de dicha población para obtener los fenotipos, es decir, la función de escalarización codificada en cada individuo. Después se utiliza la función descrita en el algoritmo 3 para calcular la aptitud de cada individuo de la población. Posteriormente se inicia el bucle principal del algoritmo: se seleccionan los pares de padres a cruzar, se genera una nueva población, se decodifican los genotipos de la nueva población, se calcula su aptitud y se actualizan los valores utilizados para fines estadísticos. Después, la nueva población sustituye a la anterior. Este bucle es ejecutado mientras no se satisfaga alguna de las siguientes condiciones de paro:

1. Aptitud mínima alcanzada por algún individuo.
2. Número máximo de generaciones alcanzado.

Si bien MOMBI-II es más rápido que los AEMOs que utilizan el cálculo del hipervolumen durante el proceso de solución, el costo computacional del algoritmo propuesto puede ser considerable, ya que en cada iteración de la programación genética se calcula el hipervolumen de cada una de las  $n$  soluciones obtenidas para cada individuo de la población. Por

---

**Algoritmo 4** Algoritmo principal

---

**Entrada:** *Archivo de parámetros*

- 1:  $i \leftarrow 1$
  - 2: Inicializar aleatoriamente población  $P_i$
  - 3: Decodificar genotipos de  $P_i$
  - 4: Calcular aptitud de  $P_i$
  - 5: **mientras** *condición de paro no satisfecha* **hacer**
  - 6:    Seleccionar padres de  $P_i$
  - 7:     $P'_i \leftarrow$  Generar genotipos de la descendencia de  $P_i$
  - 8:    Decodificar genotipos de  $P'_i$
  - 9:    Calcular aptitud de  $P'_i$
  - 10:    Actualizar estadísticas
  - 11:     $P_i = P'_i$
  - 12:     $i \leftarrow i + 1$
  - 13: Devolver  $P_i$
- 

lo tanto, se utilizó el algoritmo incremental para el cálculo del hipervolumen propuesto por el *Walking Fish Group*, el cual es el más veloz conocido en la literatura. Sin embargo, se utilizó la versión corregida presentada en [32]. El código fuente de la implementación utilizada se encuentra disponible en [https://computacion.cs.cinvestav.mx/~abernabe/tesis/software/ELGP\\_MOMBI2.zip](https://computacion.cs.cinvestav.mx/~abernabe/tesis/software/ELGP_MOMBI2.zip).

### 5.3.1 PARÁMETROS NECESARIOS

En su versión original, ELGP requiere de dos archivos para su ejecución: uno con los parámetros de configuración y otro con los datos que se buscan ajustar mediante las funciones generadas. No obstante, como ya se mencionó, en nuestra implementación no requerimos del conjunto de datos a ajustar, por lo que únicamente se necesita de los parámetros de configuración. La lista completa de parámetros se muestra en el apéndice A.1.

De igual forma, MOMBI-II requiere de dos archivos: uno con los parámetros de configuración y otro con los valores de los pesos que permitirán definir las direcciones de búsqueda en el POM. Debido a que la única modificación realizada a MOMBI-II es el uso de la función de escalarización utilizada se requiere de ambos archivos de configuración. La lista completa de parámetros del archivo de configuración se muestra en el apéndice A.2. Por otro lado, el archivo de pesos almacena las coordenadas de los vectores

de pesos, con el número de vectores y el tamaño de la dimensionalidad indicados en la primera línea del archivo. En el apéndice A.3 se muestra un ejemplo.

Consecuentemente, la implementación final requiere de todos estos parámetros para su ejecución.

### 5.3.2 ARCHIVOS GENERADOS

El resultado final de la ejecución del algoritmo es un conjunto de archivos, cuyo contenido se describe a continuación:

- `best_ind`. Contiene estadísticas del mejor individuo encontrado durante la ejecución del algoritmo.
- `data`. Contiene estadísticas de la población en cada generación en forma resumida.
- `last_pop`. Contiene estadísticas de cada individuo que forma parte de la última población.
- `log`. Contiene un resumen de los parámetros de configuración utilizados y estadísticas de la población en cada generación en forma detallada.

### 5.3.3 REQUERIMIENTOS DE SOFTWARE

ELGP está implementado en el lenguaje de programación C++, mientras que MOMBI-II está implementado en C, por lo que fue necesario realizar algunas modificaciones menores para poder integrarlos completamente. El código final utiliza ambos lenguajes de programación y fue compilado con g++ en su versión 5.4.

Adicionalmente, ELGP requiere de la biblioteca Eigen [1] para realizar cálculos de álgebra lineal. En la implementación utilizada se instaló la versión 3.3.7 disponible en <http://bitbucket.org/eigen/eigen/get/3.3.7.tar.bz2>.

## 5.4 | GENERACIÓN DE NUEVAS FUNCIONES

Utilizando la implementación descrita en la sección anterior, realizamos múltiples ejecuciones con distintos parámetros de configuración. Las primeras pruebas realizadas consideraban una generación completamente a

cargo de la programación genética. Es decir, que MOMBI-II utilizaba la función generada por ELGP de forma directa. Sin embargo, observamos que al utilizar el operador de valor absoluto a la función generada, el desempeño de las funciones incrementaba considerablemente. Por lo tanto, en los experimentos subsecuentes utilizamos dicho operador a todas las funciones generadas.

En el proceso de solución de MOMBI-II utilizamos distintos problemas de prueba de la familia DTLZ, definidos formalmente en el siguiente capítulo, específicamente DTLZ1, DTLZ4 y DTLZ7. El propósito de esto fue utilizar POMs cuyos frentes de Pareto tuvieran distintas geometrías (lineal, cóncavo y disconexo). Si bien las funciones obtenidas lograban un buen desempeño en el POM utilizado, esperábamos encontrar funciones que fueran capaces de “generalizar” su buen desempeño en POMs con distintas características. Por lo tanto, utilizando cada una de las 20 mejores funciones que devuelve el algoritmo, resolvimos 7 POMs de prueba de la familia DTLZ. Estos fueron resueltos considerando 2 y 3 objetivos para poder visualizar los frentes de Pareto de forma gráfica y, posteriormente, calculamos el hipervolumen para poder cuantificar el desempeño de las soluciones obtenidas.

Por otro lado, consideramos el uso de operadores aritméticos y trigonométricos. Sin embargo, observamos que el uso de operadores trigonométricos alentaba la ejecución del algoritmo y decrementaba la calidad de las funciones de escalarización generadas.

Dichos experimentos fueron realizados en una computadora con un procesador Intel Core i5-5200U, con una memoria RAM de 8 GB DDR3 y utilizando el sistema operativo Ubuntu 16.04. Cabe aclarar que el tiempo de ejecución del algoritmo fue considerablemente alto, ya que utilizando un número de generaciones entre 35 y 50, el algoritmo requiere entre 5 y 7 días para finalizar su ejecución.

Finalmente, el conjunto de funciones que obtuvieron un mejor desempeño y lograron generalizar mejor los POMs de prueba fueron las funciones obtenidas en el experimento que resolvió el problema DTLZ4 con dos objetivos. El punto de referencia utilizado para el cálculo del hipervolumen fue el punto (1,1) y  $HV_{max} = 1$ .

Las dos mejores funciones de escalarización obtenidas fueron las siguientes:

$$F4A(\vec{f}', \vec{w}) := \max_i \left( |3f'_i + w_i + \frac{f'_i}{w_i} + \frac{f'_i}{w_i^2}| \right),$$

$$F4B(\vec{f}', \vec{w}) := \max_i \left( \left| w_i - \frac{f'_i}{w_i} - f'_i \right| \right).$$

En el apéndice B se muestran los archivos de configuración completos utilizados en este experimento.

## 5.5 | RESUMEN

En este capítulo presentamos una descripción de los dos algoritmos principales que fueron combinados para la generación de la implementación utilizada para realizar la experimentación de este trabajo. También se presenta una descripción del funcionamiento general de dicha implementación y de los parámetros y requerimientos de software necesarios para su ejecución. Este algoritmo fue utilizado para realizar diversos experimentos, sin embargo, se proporcionan los parámetros del experimento principal, ya que a partir de éste se obtuvieron dos funciones de escalarización cuyo desempeño se evalúa en el siguiente capítulo.





---

## VALIDACIÓN EXPERIMENTAL

En este capítulo se describen las herramientas que utilizamos para validar el desempeño de las funciones de escalarización generadas mediante la implementación descrita en el capítulo anterior. Para ello se resolvieron distintos Problemas de Optimización Multi-objetivo (POMs) de prueba, cada uno con las funciones cuyo desempeño buscamos evaluar. Posteriormente se calcularon tres indicadores de las soluciones obtenidas para poder comparar los resultados. En la sección 6.1 se describen los problemas de prueba resueltos en este proceso de validación. En la sección 6.2 se describen los indicadores utilizados para medir la calidad de las soluciones obtenidas. En la sección 6.3 se muestran los resultados obtenidos. Finalmente en la sección 6.4 se presenta un resumen del capítulo.

### 6.1 | PROBLEMAS DE PRUEBA UTILIZADOS

Existen distintas familias de POMs de prueba que son utilizados en el área para comparar el desempeño de diferentes Algoritmos Evolutivos Multi-objetivo (AEMOs). Dos conjuntos de problemas de prueba comúnmente utilizados son los denominados Deb-Thiele-Laumanns-Zitzler (DTLZ) [19] y Walking Fish Group (WFG) [35]. Ambas familias de problemas son artificiales, ya que no modelan problemas del mundo real. No obstante, esto proporciona algunas ventajas como son: escalabilidad, conocimiento de la forma y ubicación exacta del frente de Pareto, menor tiempo de ejecución y facilidad de comprensión, implementación y visualización [30].

Decidimos utilizar estos problemas principalmente porque son escalables a cualquier número de objetivos y variables, lo cual resulta importante

para observar el comportamiento de las nuevas funciones de escalarización en problemas con distintas dimensionalidades.

Por otro lado, cabe aclarar que la mayoría de los problemas contenidos en los conjuntos DTLZ y WFG tienen un frente de Pareto similar a un simplex. Esto puede favorecer el desempeño de un AEMO que utiliza funciones de escalarización, ya que a menudo se utilizan vectores de pesos uniformemente distribuidos sobre un simplex para guiar la búsqueda en este tipo de algoritmos. Por lo tanto decidimos trabajar con la familia de problemas de prueba I-DTLZ [43], los cuales consisten en una variación de los problemas DTLZ en los que el frente de Pareto se encuentra invertido, por lo que un AEMO que utiliza funciones de escalarización suele tener una degradación en la calidad de las soluciones obtenidas en estos problemas [7].

La heurística MOMBI-II implementa dos de las familias de problemas de prueba utilizadas (DTLZ y WFG). Por otro lado la familia de problemas I-DTLZ son una variante de los problemas DTLZ, por lo que su implementación en el código requirió únicamente modificar la definición de dichas funciones.

A continuación se define formalmente cada uno de los problemas de prueba utilizados, así como una gráfica de su verdadero frente de Pareto en su versión con 3 funciones objetivo.

### 6.1.1 CONJUNTO DEB-THIELE-LAUMANN-S-ZITZLER (DTLZ)

La familia de problemas DTLZ consta de nueve problemas de optimización multi-objetivo (DTLZ1-DTLZ9), cuyos frentes de Pareto son conocidos e incluyen geometrías unifrontales, multifrontales y degenerados. En este trabajo utilizamos los primeros siete problemas, los cuales carecen de restricciones explícitas e incluyen frentes de Pareto con geometrías multifrontales y degeneradas.

#### DTLZ1

Este problema de prueba es separable, multifrontal y su frente de Pareto es lineal. Está definido por la siguiente expresión:

$$\text{Dado } \vec{x} = \{x_1, \dots, x_{m-1}, x_m, \dots, x_n\}.$$

Minimizar

$$f_1(\vec{x}) = 0.5(1 + g(\vec{y})) \prod_{i=1}^{m-1} x_i,$$

$$f_{j=1:m-1}(\vec{x}) = 0.5(1 + g(\vec{y}))(1 - x_{m-j+1}) \prod_{i=1}^{m-j} x_i,$$

$$f_m(\vec{x}) = 0.5(1 + g(\vec{y}))(1 - x_1),$$

donde:

$$y_{i=1:k} = \{x_m, x_{m+1}, \dots, x_n\},$$

$$g(\vec{y}) = 100 \left[ k + \sum_{i=1}^k (y_i - 0.5)^2 - \cos(20\pi(y_i - 0.5)) \right],$$

sujeto a  $n = m + k - 1$  y  $x_i \in [0, 1], \forall i \in \{1, \dots, n\}$ . Se sugiere utilizar  $k = 5$ .

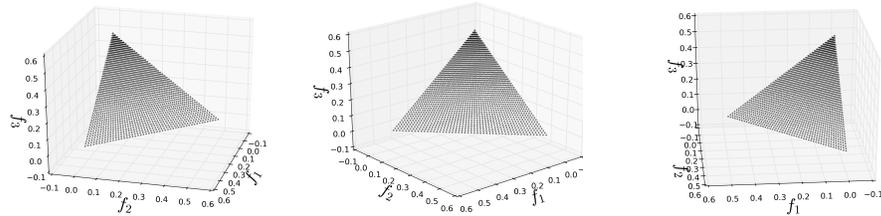


Figura 6.1: Frente de Pareto de DTLZ1 con 3 funciones objetivo.

## DTLZ2

Este problema de prueba es separable, unimodal y su frente de Pareto es cóncavo. Está definido por la siguiente expresión:

$$\text{Dado } \vec{x} = \{x_1, \dots, x_{m-1}, x_m, \dots, x_n\}.$$

Minimizar

$$f_1(\vec{x}) = (1 + g(\vec{y})) \prod_{i=1}^{m-1} \cos(x_i\pi/2),$$

$$f_{j=2:m-1}(\vec{x}) = (1 + g(\vec{y})) \left( \prod_{i=1}^{m-j} \cos(x_i\pi/2) \right) \text{sen}(x_{m-j+1}\pi/2),$$

$$f_m(\vec{x}) = (1 + g(\vec{y})) \text{sen}(x_1\pi/2),$$

donde:

$$y_{i=1:k} = \{x_m, x_{m+1}, \dots, x_n\},$$

$$g(\vec{y}) = \sum_{i=1}^k (y_i - 0.5)^2,$$

sujeto a  $x_i \in [0, 1], \forall i \in \{1, \dots, n\}$ . Se sugiere utilizar  $k = 10$ .

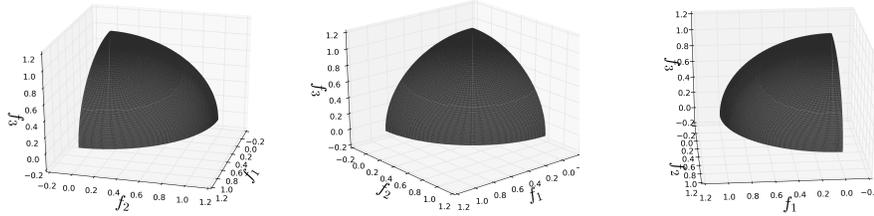


Figura 6.2: Frente de Pareto de DTLZ2 con 3 funciones objetivo.

### DTLZ3

Este problema de prueba es igual a DTLZ2 exceptuando por la función  $g$ , la cual lo hace multifrontal. Está definido por la siguiente expresión:

Dado  $\vec{x} = \{x_1, \dots, x_{m-1}, x_m, \dots, x_n\}$ .

Minimizar

$$f_1(\vec{x}) = (1 + g(\vec{y})) \prod_{i=1}^{m-1} \cos(x_i \pi / 2),$$

$$f_{j=2:m-1}(\vec{x}) = (1 + g(\vec{y})) \left( \prod_{i=1}^{m-j} \cos(x_i \pi / 2) \right) \text{sen}(x_{m-j+1} \pi / 2),$$

$$f_m(\vec{x}) = (1 + g(\vec{y})) \text{sen}(x_1 \pi / 2),$$

donde:

$$y_{i=1:k} = \{x_m, x_{m+1}, \dots, x_n\},$$

$$g(\vec{y}) = 100 \left[ k + \sum_{i=1}^k (y_i - 0.5)^2 - \cos(20\pi(y_i - 0.5)) \right],$$

sujeto a  $x_i \in [0, 1], \forall i \in \{1, \dots, n\}$ . Se sugiere utilizar  $k = 10$ .

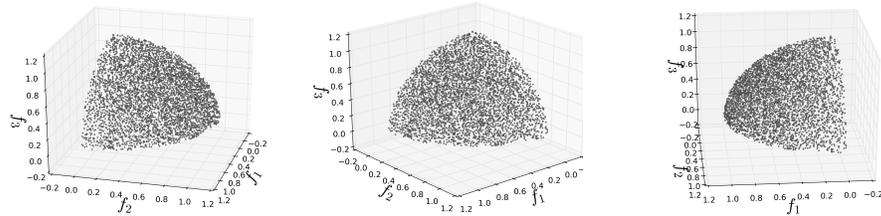


Figura 6.3: Frente de Pareto de DTLZ3 con 3 funciones objetivo.

#### DTLZ4

Este problema de prueba es separable, unifrontal y con un frente cóncavo. Está definido por la siguiente expresión:

$$\text{Dado } \vec{x} = \{x_1, \dots, x_{m-1}, x_m, \dots, x_n\}.$$

Minimizar

$$f_1(\vec{x}) = (1 + g(\vec{y})) \prod_{i=1}^{m-1} \cos(x_i^\alpha \pi/2),$$

$$f_{j=2:m-1}(\vec{x}) = (1 + g(\vec{y})) \left( \prod_{i=1}^{m-j} \cos(x_i^\alpha \pi/2) \right) \text{sen}(x_{m-j+1}^\alpha \pi/2),$$

$$f_m(\vec{x}) = (1 + g(\vec{y})) \text{sen}(x_1^\alpha \pi/2),$$

donde:

$$y_{i=1:k} = \{x_m, x_{m+1}, \dots, x_n\},$$

$$g(\vec{y}) = \sum_{i=1}^k (y_i - 0.5)^2,$$

sujeto a  $x_i \in [0, 1], \forall i \in \{1, \dots, n\}$ . Se sugiere utilizar los parámetros  $\alpha = 100$  y  $k = 10$ .

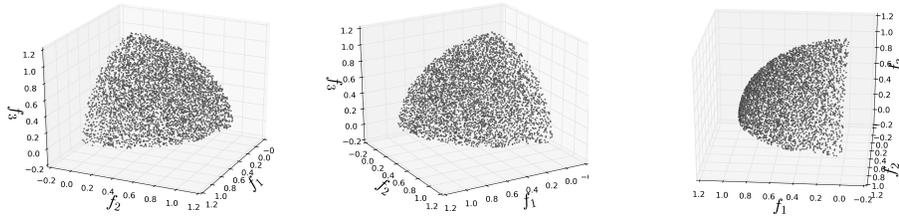


Figura 6.4: Frente de Pareto de DTLZ4 con 3 funciones objetivo.

### DTLZ5

Este problema de prueba es unifrontal y degenerado con un frente cóncavo. Está definido por la siguiente expresión:

Dado  $\vec{x} = \{x_1, \dots, x_{m-1}, x_m, \dots, x_n\}$ .

Minimizar

$$f_1(\vec{x}) = (1 + g(\vec{y})) \prod_{i=1}^{m-1} \cos(\theta_i \pi/2),$$

$$f_{j=2:m-1}(\vec{x}) = (1 + g(\vec{y})) \left( \prod_{i=1}^{m-j} \cos(\theta_i^\alpha \pi/2) \right) \text{sen}(\theta_{m-j+1}^\alpha \pi/2),$$

$$f_m(\vec{x}) = (1 + g(\vec{y})) \text{sen}(\theta_1^\alpha \pi/2),$$

donde:

$$y_{i=1:k} = \{x_m, x_{m+1}, \dots, x_n\},$$

$$\theta_i = \begin{cases} x_i, & i = 1 \\ \frac{1+2g(\vec{y})}{2(1+g(\vec{y}))} x_i, & \forall i \in \{2, 3, \dots, m-1\} \end{cases},$$

$$g(\vec{y}) = \sum_{i=1}^k (y_i - 0.5)^2,$$

sujeto a  $x_i \in [0, 1], \forall i \in \{1, \dots, n\}$ . Se sugiere utilizar  $k = 10$ .

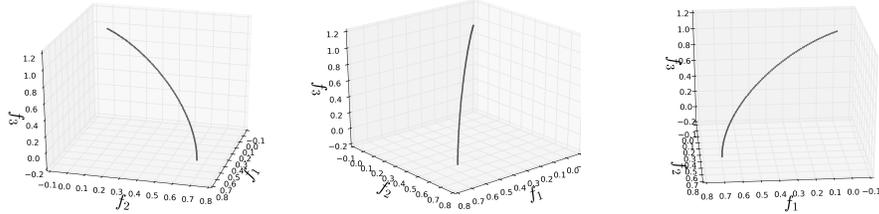


Figura 6.5: Frente de Pareto de DTLZ5 con 3 funciones objetivo.

### DTLZ6

Este problema de prueba es una modificación de DTLZ5, variando la función  $g$ . El problema es unifrontal y degenerado, definido por la siguiente expresión:

$$\text{Dado } \vec{x} = \{x_1, \dots, x_{m-1}, x_m, \dots, x_n\}.$$

Minimizar

$$f_1(\vec{x}) = (1 + g(\vec{y})) \prod_{i=1}^{m-1} \cos(\theta_i \pi/2),$$

$$f_{j=2:m-1}(\vec{x}) = (1 + g(\vec{y})) \left( \prod_{i=1}^{m-j} \cos(\theta_i^\alpha \pi/2) \right) \text{sen}(\theta_{m-j+1}^\alpha \pi/2),$$

$$f_m(\vec{x}) = (1 + g(\vec{y})) \text{sen}(\theta_1^\alpha \pi/2),$$

donde:

$$y_{i=1:k} = \{x_m, x_{m+1}, \dots, x_n\},$$

$$\theta_i = \begin{cases} x_i, & i = 1 \\ \frac{1+2g(\vec{y})}{2(1+g(\vec{y}))} x_i, & \forall i \in \{2, 3, \dots, m-1\} \end{cases},$$

$$g(\vec{y}) = \sum_{i=1}^k (y_i - 0.5)^2,$$

sujeto a  $x_i \in [0, 1], \forall i \in \{1, \dots, n\}$ . Se sugiere utilizar  $k = 10$ .

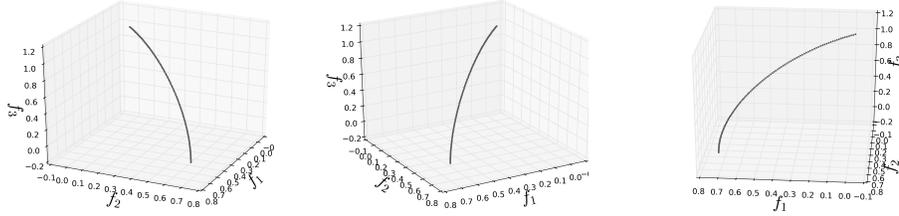


Figura 6.6: Frente de Pareto de DTLZ6 con 3 funciones objetivo.

### DTLZ7

Este problema tiene un frente de Pareto disconexo con un total de  $2^{m-1}$  regiones de óptimos de Pareto en el espacio de búsqueda. Está definido por la siguiente expresión:

$$\text{Dado } \vec{x} = \{x_1, \dots, x_{m-1}, x_m, \dots, x_n\}.$$

Minimizar

$$f_{j=1:m-1}(\vec{x}) = x_j,$$

$$f_m(\vec{x}) = (1 + g(\vec{y})) \left( m - \sum_{i=1}^{m-1} \left[ \frac{f_i}{1 + g(\vec{y})} (1 + \text{sen}(3\pi f_i)) \right] \right),$$

donde:

$$y_{i=1:k} = \{x_m, x_{m+1}, \dots, x_n\},$$

$$g(\vec{y}) = 1 + \frac{9}{k} \sum_{i=1}^k y_i,$$

sujeto a  $x_i \in [0, 1], \forall i \in \{1, \dots, n\}$ . Se sugiere utilizar  $k = 20$ .

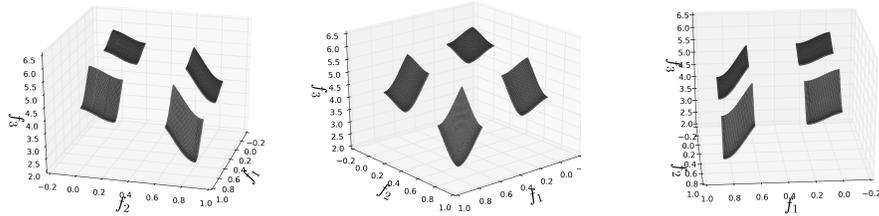


Figura 6.7: Frente de Pareto de DTLZ7 con 3 funciones objetivo.

### 6.1.2 WALKING FISH GROUP (WFG)

La familia de problemas WFG está compuesta de nueve problemas de optimización multi-objetivo (WFG1-WFG9). En este trabajo utilizamos todos estos problemas para realizar pruebas.

#### WFG1

Este problema de prueba es separable y unifrontal. Está definido por la siguiente expresión:

$$\text{Dado } \vec{x} = \{x_1, \dots, x_{m-1}, x_m, \dots, x_n\}.$$

Minimizar

$$f_1(\vec{x}) = x_m + 2 \prod_{i=1}^{m-1} (1 - \cos(x_i \pi / 2)),$$

$$f_{j=2:m-1}(\vec{x}) = x_m + 2j \left( \prod_{i=1}^{m-j} (1 - \cos(x_i \pi / 2)) \right) (1 - \sin(x_{m-j+1} \pi / 2)),$$

$$f_m(\vec{x}) = x_m + 2m \left( 1 - x_1 - \frac{\cos(10\pi x_1 + \pi/2)}{10\pi} \right),$$

donde:

$$x_{i=1:m-1} = r\_sum(\{y_{(i-1)k/(m-1)+1}, \dots, y_{ik/(m-1)}\},$$

$$\{2(i-1)k/(m-1) + 1, \dots, 2ik/(m-1)\}),$$

$$x_m = r\_sum(\{y_{k+1}, \dots, y_n\}, \{2(k+1), \dots, 2n\}),$$

$$y_{i=1:n} = b\_poly(y'_i, 0.02),$$

$$y'_{i=1:k} = y''_i,$$

$$y'_{i=k+1:n} = b\_flat(y''_i, 0.8, 0.75, 0.85),$$

$$y''_{i=1:k} = z_i / (2i),$$

$$y''_{i=k+1:n} = s\_linear(z_i / (2i), 0.35).$$

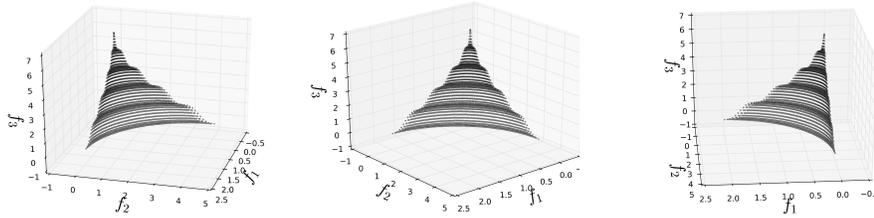


Figura 6.8: Frente de Pareto de WFG1 con 3 funciones objetivo.

## WFG2

Este problema de prueba es no separable y multifrontal. Está definido por la siguiente expresión:

Dado  $\vec{x} = \{x_1, \dots, x_{m-1}, x_m, \dots, x_n\}$ .  
 Minimizar

$$f_1(\vec{x}) = x_m + 2 \prod_{i=1}^{m-1} (1 - \cos(x_i \pi / 2)),$$

$$f_{j=2:m-1}(\vec{x}) = x_m + 2j \left( \prod_{i=1}^{m-j} (1 - \cos(x_i \pi / 2)) \right) (1 - \sin(x_{m-j+1} \pi / 2)),$$

$$f_m(\vec{x}) = x_m + 2m (1 - x_1 \cos^2(5x_1 \pi)),$$

donde:

$$x_{i=1:m-1} = r\_sum \left( \{y_{(i-1)k/(m-1)+1}, \dots, y_{ik/(m-1)}\}, \{1, \dots, 1\} \right),$$

$$x_m = r\_sum \left( \{y_{k+1}, \dots, y_{k+l/2}\}, \{1, \dots, 1\} \right),$$

$$y_{i=1:k} = y'_i,$$

$$y_{i=k+1:k+l/2} = r\_nonsep \left( \{y'_{k+2(i-k)-1}, y'_{k+2(i-k)}\}, 2 \right),$$

$$y'_{i=1:k} = z_i / (2i),$$

$$y'_{i=k+1:n} = s\_linear(z_i / (2i), 0.35).$$

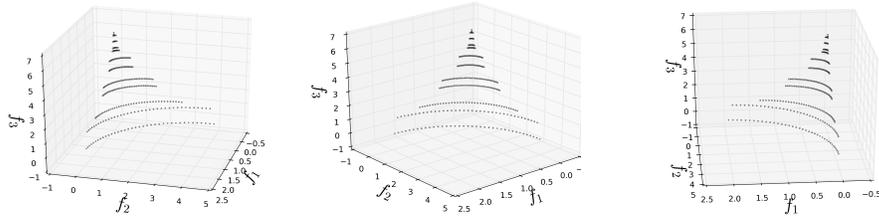


Figura 6.9: Frente de Pareto de WFG2 con 3 funciones objetivo.

### WFG3

Este problema de prueba es no separable y unifrontal. Está definido por la siguiente expresión:

Dado  $\vec{x} = \{x_1, \dots, x_{m-1}, x_m, \dots, x_n\}$ .

Minimizar

$$f_1(\vec{x}) = x_m + 2 \prod_{i=1}^{m-1} x_i,$$

$$f_{j=2:m-1}(\vec{x}) = x_m + 2j \left( \prod_{i=1}^{m-j} (x_i) \right) (1 - x_{m-j+1}),$$

$$f_m(\vec{x}) = x_m + 2m(1 - x_1),$$

donde:

$$x_{i=1} = u_i,$$

$$x_{i=2:m-1} = x_m(u_i - 0.5) + 0.5,$$

$$x_m = r\_sum(\{y_{k+1}, \dots, y_{k+l/2}\}, \{1, \dots, 1\}),$$

$$u_i = r\_sum(\{y_{(i-1)k/(m-1)+1}, \dots, y_{ik/(m-1)}\}, \{1, \dots, 1\}),$$

$$y_{i=1:k} = y'_i,$$

$$y_{i=k+1:k+l/2} = r\_nonsep(\{y'_{k+2(i-k)-1}, y'_{k+2(i-k)}\}, 2),$$

$$y'_{i=1:k} = z_i/(2i),$$

$$y'_{i=k+1:n} = s\_linear(z_i/(2i), 0.35).$$

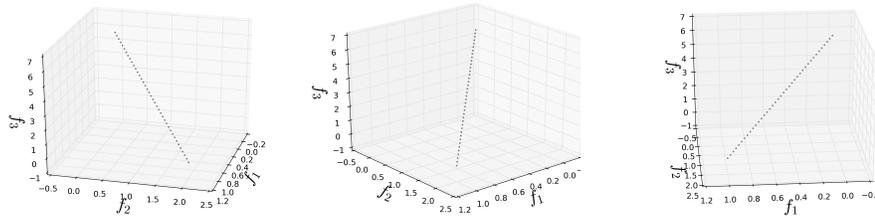


Figura 6.10: Frente de Pareto de WFG3 con 3 funciones objetivo.

#### WFG4

Este problema de prueba es separable y multifrontal. Está definido por la siguiente expresión:

$$\text{Dado } \vec{x} = \{x_1, \dots, x_{m-1}, x_m, \dots, x_n\}.$$

Minimizar

$$f_1(\vec{x}) = x_m + 2 \prod_{i=1}^{m-1} \text{sen}(x_i\pi/2),$$

$$f_{j=2:m-1}(\vec{x}) = x_m + 2j \left( \prod_{i=1}^{m-j} \text{sen}(x_i\pi/2) \right) (\cos(x_{m-j+1}\pi/2)),$$

$$f_m(\vec{x}) = x_m + 2m \cos(x_1\pi/2),$$

donde:

$$x_{i=1:m-1} = r\_sum(\{y_{(i-1)k/(m-1)+1}, \dots, y_{ik/(m-1)}\}, \{1, \dots, 1\}),$$

$$x_m = r\_sum(\{y_{k+1}, \dots, y_n\}, \{1, \dots, 1\}),$$

$$y_{i=1:n} = s\_multi(z_i/(2i), 30, 10, 0.35).$$

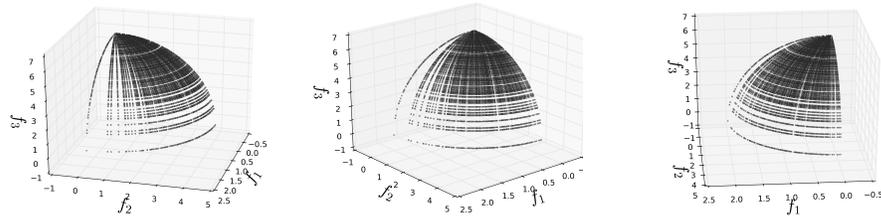


Figura 6.11: Frente de Pareto de WFG4 con 3 funciones objetivo.

### WFG5

Este problema de prueba es separable y deceptivo. Está definido por la siguiente expresión:

Dado  $\vec{x} = \{x_1, \dots, x_{m-1}, x_m, \dots, x_n\}$ .

Minimizar

$$f_1(\vec{x}) = x_m + 2 \prod_{i=1}^{m-1} \text{sen}(x_i\pi/2),$$

$$f_{j=2:m-1}(\vec{x}) = x_m + 2j \left( \prod_{i=1}^{m-j} \text{sen}(x_i\pi/2) \right) (\cos(x_{m-j+1}\pi/2)),$$

$$f_m(\vec{x}) = x_m + 2m \cos(x_1\pi/2),$$

donde:

$$\begin{aligned} x_{i=1:m-1} &= r\_sum(\{y_{(i-1)k/(m-1)+1}, \dots, y_{ik/(m-1)}\}, \{1, \dots, 1\}), \\ x_m &= r\_sum(\{y_{k+1}, \dots, y_n\}, \{1, \dots, 1\}), \\ y_{i=1:n} &= s\_multi(z_i/(2i), 30, 10, 0.35). \end{aligned}$$

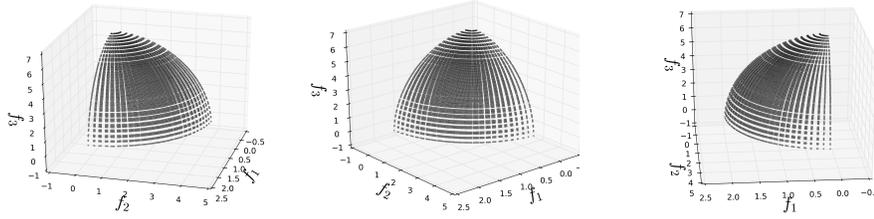


Figura 6.12: Frente de Pareto de WFG5 con 3 funciones objetivo.

### WFG6

Este problema de prueba es no separable y unifrontal. Está definido por la siguiente expresión:

Dado  $\vec{x} = \{x_1, \dots, x_{m-1}, x_m, \dots, x_n\}$ .

Minimizar

$$\begin{aligned} f_1(\vec{x}) &= x_m + 2 \prod_{i=1}^{m-1} \text{sen}(x_i\pi/2), \\ f_{j=2:m-1}(\vec{x}) &= x_m + 2j \left( \prod_{i=1}^{m-j} \text{sen}(x_i\pi/2) \right) (\cos(x_{m-j+1}\pi/2)), \\ f_m(\vec{x}) &= x_m + 2m \cos(x_1\pi/2), \end{aligned}$$

donde:

$$\begin{aligned} x_{i=1:m-1} &= r\_nonsep(\{y_{(i-1)k/(m-1)+1}, \dots, y_{ik/(m-1)}\}, k/(m-1)) \\ x_m &= r\_nonsep(\{y_{k+1}, \dots, y_n\}, l), \\ y_{i=1:k} &= z_i/(2i), \\ y_{i=k+1:n} &= s\_linear(z_i/(2i), 0.35). \end{aligned}$$

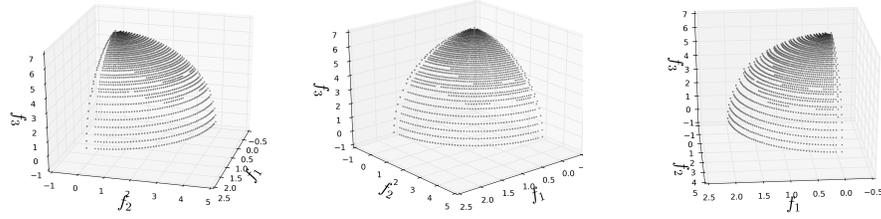


Figura 6.13: Frente de Pareto de WFG6 con 3 funciones objetivo.

### WFG7

Este problema de prueba es separable y unifrontal. Está definido por la siguiente expresión:

Dado  $\vec{x} = \{x_1, \dots, x_{m-1}, x_m, \dots, x_n\}$  y  $\vec{z} = \{z_1, \dots, z_k, x_{k+1}, \dots, z_n\}$ .  
Minimizar

$$f_1(\vec{x}) = x_m + 2 \prod_{i=1}^{m-1} \text{sen}(x_i \pi / 2),$$

$$f_{j=2:m-1}(\vec{x}) = x_m + 2j \left( \prod_{i=1}^{m-j} \text{sen}(x_i \pi / 2) \right) (\cos(x_{m-j+1} \pi / 2)),$$

$$f_m(\vec{x}) = x_m + 2m \cos(x_1 \pi / 2),$$

donde:

$$x_{i=1:m-1} = r\_sum(\{y_{(i-1)k/(m-1)+1}, \dots, y_{ik/(m-1)}\}, k/(m-1)),$$

$$x_m = r\_sum(\{y_{k+1}, \dots, y_n\}, \{1, \dots, 1\}),$$

$$y_{i=1:k} = y'_i,$$

$$y_{i=k+1:n} = s\_linear(y'_i, 0.35),$$

$$y'_{i=1:k} = b\_param(z_i/(2i), r\_sum(\{z_{i+1}/2(i+1), \dots, z_n(2n)\}, \{1, \dots, 1\}), \frac{0.98}{49.98}, 0.02, 50),$$

$$y'_{i=k+1:n} = z_i/(2i).$$

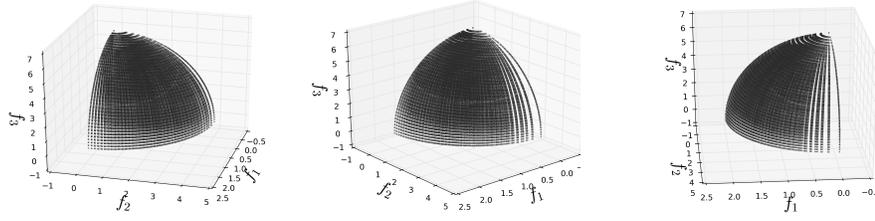


Figura 6.14: Frente de Pareto de WFG7 con 3 funciones objetivo.

### WFG8

Este problema de prueba es no separable y unifrontal. Está definido por la siguiente expresión:

Dado  $\vec{x} = \{x_1, \dots, x_{m-1}, x_m, \dots, x_n\}$  y  $\vec{z} = \{z_1, \dots, z_k, x_{k+1}, \dots, z_n\}$ .  
Minimizar

$$f_1(\vec{x}) = x_m + 2 \prod_{i=1}^{m-1} \text{sen}(x_i \pi / 2),$$

$$f_{j=2:m-1}(\vec{x}) = x_m + 2j \left( \prod_{i=1}^{m-j} \text{sen}(x_i \pi / 2) \right) (\cos(x_{m-j+1} \pi / 2)),$$

$$f_m(\vec{x}) = x_m + 2m \cos(x_1 \pi / 2),$$

donde:

$$x_{i=1:m-1} = r\_sum(\{y_{(i-1)k/(m-1)+1}, \dots, y_{ik/(m-1)}\}, \{1, \dots, 1\}),$$

$$x_m = r\_sum(\{y_{k+1}, \dots, y_n\}, \{1, \dots, 1\}),$$

$$y_{i=1:k} = y'_i,$$

$$y_{i=k+1:n} = s\_linear(y'_i, 0.35),$$

$$y'_{i=1:k} = z_i / (2i),$$

$$y'_{i=k+1:n} = b\_param(z_i / (2i), r\_sum(\{z_1/2, \dots, z_{i-1}/(2(i-1))\}, \{1, \dots, 1\}), \frac{0.98}{49.98}, 0.02, 50).$$

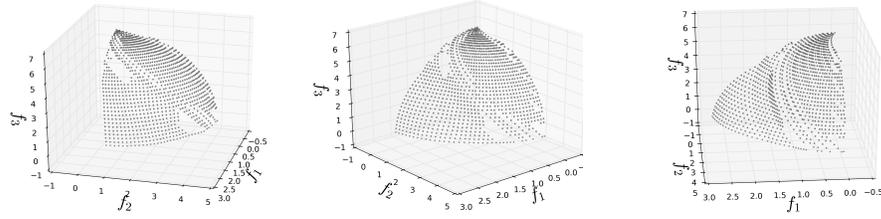


Figura 6.15: Frente de Pareto de WFG8 con 3 funciones objetivo.

### WFG9

Este problema de prueba es no separable, multifrontal y deceptivo. Está definido por la siguiente expresión:

Dado  $\vec{x} = \{x_1, \dots, x_{m-1}, x_m, \dots, x_n\}$  y  $\vec{z} = \{z_1, \dots, z_k, x_{k+1}, \dots, z_n\}$ .  
Minimizar

$$f_1(\vec{x}) = x_m + 2 \prod_{i=1}^{m-1} \text{sen}(x_i \pi / 2),$$

$$f_{j=2:m-1}(\vec{x}) = x_m + 2j \left( \prod_{i=1}^{m-j} \text{sen}(x_i \pi / 2) \right) (\cos(x_{m-j+1} \pi / 2)),$$

$$f_m(\vec{x}) = x_m + 2m \cos(x_1 \pi / 2),$$

donde:

$$x_{i=1:m-1} = r\_nonsep(\{y_{(i-1)k/(m-1)+1}, \dots, y_{ik/(m-1)}\}, k/(m-1)),$$

$$x_m = r\_nonsep(\{y_{k+1}, \dots, y_n\}, l),$$

$$y_{i=1:k} = s\_decept(y'_i, 0.35, 0.001, 0.05),$$

$$y_{i=k+1:n} = s\_multi(y'_i, 30, 95, 0.35),$$

$$y'_{i=1:n-1} = b\_param(z_i/(2i), r\_sum(\{z_{i+1}/(2(i+1)), \dots, z_n/(2n)\}, \{1, \dots, 1\}), \frac{0.98}{49.98}, 0.02, 50),$$

$$y'_n = z_n/(2n).$$

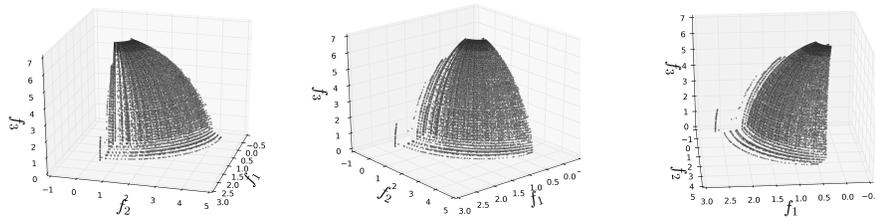


Figura 6.16: Frente de Pareto de WFG9 con 3 funciones objetivo.

### 6.1.3 I-DTLZ

La familia de problemas I-DTLZ es una modificación de los problemas DTLZ. La idea de proponer esta modificación es generar formas distintas del frente de Pareto de cada problema modificando su forma general como sigue:

$$\begin{aligned} &\text{Minimizar } \{-f_1(\vec{x}), -f_2(\vec{x}), \dots, f_m(\vec{x})\}. \\ &\text{sujeto a } \vec{x} \in \mathcal{X} \end{aligned}$$

El resultado que se obtiene es un frente de Pareto invertido, aunque algunas otras propiedades de los problemas originales pueden cambiar.

Como se señala en [41], los AEMOs que utilizan vectores de pesos como puntos de referencia tienen un deterioro en su desempeño al cambiar la forma de los frentes de Pareto de los conjuntos DTLZ y WFG. Esto es evidencia de que estos algoritmos funcionan mejor con frentes de Pareto regulares y simétricos.

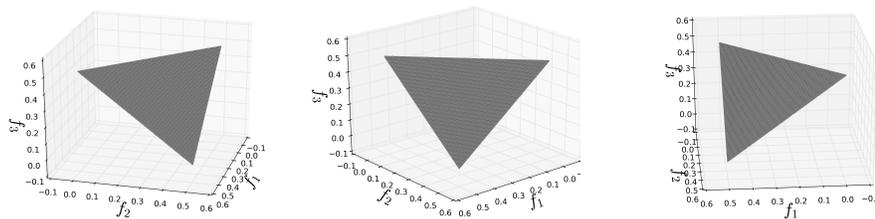


Figura 6.17: Frente de Pareto de IDTLZ1 con 3 funciones objetivo.

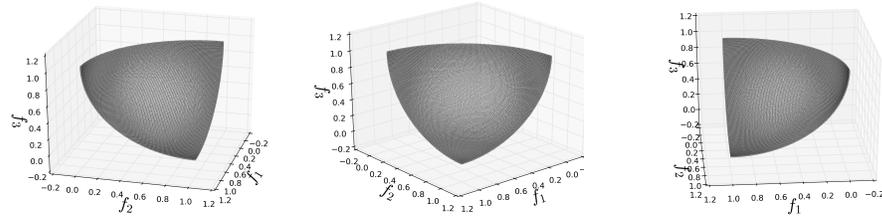


Figura 6.18: Frente de Pareto de IDTLZ2 con 3 funciones objetivo.

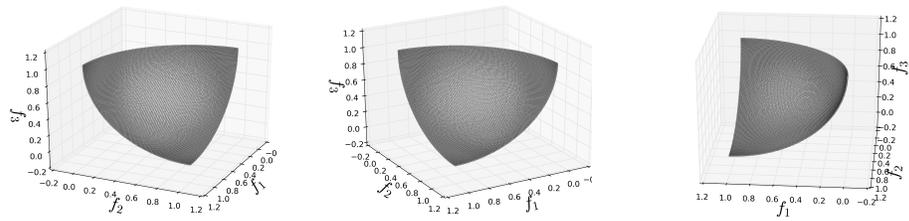


Figura 6.19: Frente de Pareto de IDTLZ3 con 3 funciones objetivo.

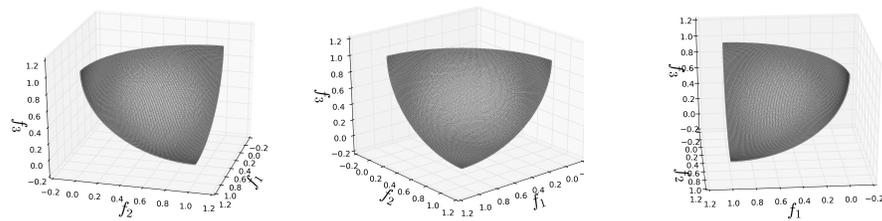


Figura 6.20: Frente de Pareto de IDTLZ4 con 3 funciones objetivo.

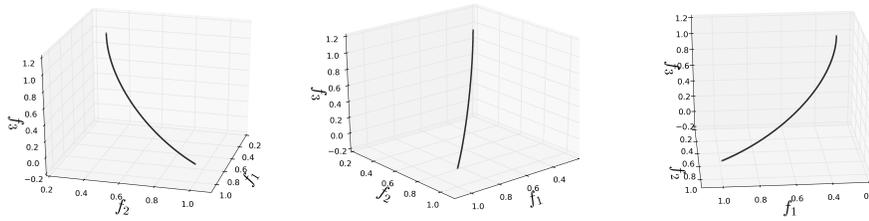


Figura 6.21: *Frente de Pareto de IDTLZ5 con 3 funciones objetivo.*

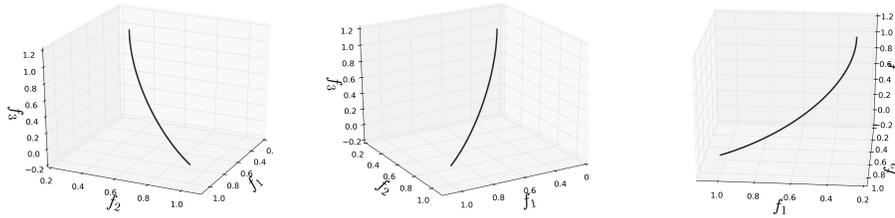


Figura 6.22: *Frente de Pareto de IDTLZ6 con 3 funciones objetivo.*

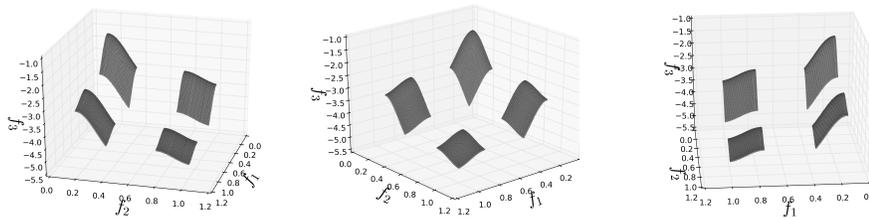


Figura 6.23: *Frente de Pareto de IDTLZ7 con 3 funciones objetivo.*

## 6.2 | INDICADORES DE DESEMPEÑO

Un indicador de desempeño evalúa la calidad de una aproximación de un frente de Pareto cuantificando aspectos como convergencia o diversidad

de las soluciones obtenidas. Uno de los principales usos de los indicadores de desempeño es el de comparar el comportamiento de distintos optimizadores. No obstante, existe una tendencia en el área a incorporar indicadores en el proceso de búsqueda de estos algoritmos [32]. En este trabajo utilizamos tres indicadores de desempeño para comparar la calidad de las soluciones obtenidas con las distintas funciones de escalarización: hipervolumen, energía-s y cobertura de conjuntos.

### 6.2.1 HIPERVOLUMEN

El indicador de hipervolumen ( $HV$ ), también llamado métrica  $S$  fue propuesto por Zitzler en [76]. Este indicador determina el tamaño de la porción del espacio objetivo que es dominada por las soluciones de un conjunto dado, acotado por un punto de referencia  $\vec{z} \in \mathcal{R}^m$ .

Este indicador es estrictamente monótono con respecto a la dominancia de Pareto. Esto implica que el frente de Pareto de un POM tiene el máximo valor posible de hipervolumen en comparación con cualquier aproximación de dicho frente, las cuales tendrán un valor menor del indicador. Dado un conjunto de aproximación  $A$  y un punto de referencia  $\vec{z}$ , el hipervolumen se define formalmente como sigue

$$HV(A, \vec{z}) = \mathcal{L}(\cup_{\vec{a} \in A} \{\vec{a}' \mid \vec{a} \prec \vec{a}' \prec \vec{z}\})$$

donde  $\mathcal{L}(\cdot)$  denota la medida de Lebesgue de un conjunto [5]. Para poder realizar la comparación entre dos conjuntos de aproximación es necesario que se realice el cálculo del hipervolumen utilizando el mismo punto de referencia  $\vec{z}$ . A su vez, dicho punto de referencia debe ser dominado por todos los puntos del conjunto cuyo hipervolumen se calcula. Por lo tanto, es común construir el vector  $\vec{z}$  con los peores valores esperados en cada función objetivo, o con los peores valores de cada función obtenidos en los conjuntos a comparar.

### 6.2.2 ENERGÍA-S

El indicador de energía-s ( $(E)_s$ ) propuesto en [28] permite cuantificar la distribución de las soluciones de un conjunto de aproximación  $A$  mediante la siguiente expresión

$$\mathcal{E}_s(A) := \sum_{i \neq j} \|a_i - a_j\|^{-s},$$

donde  $s > 0$  es un parámetro fijo. Este indicador se utiliza para discretizar una variedad en distintas dimensiones, ya que su minimización conduce a una distribución uniforme de los puntos en  $A$  cuando  $s \geq m - 1$ .

### 6.2.3 COBERTURA DE CONJUNTOS

Este indicador fue propuesto por Zitzler en [77] y permite comparar dos conjuntos utilizando como métrica el número de soluciones dominadas en cada uno de ellos. Dados dos conjuntos de aproximación distintos  $A$  y  $B$ , el indicador de cobertura  $\mathcal{C}(A, B)$  calcula el porcentaje de soluciones en  $B$  que son débilmente dominados por las soluciones de  $A$ . Formalmente, se define como sigue

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

El valor  $\mathcal{C}(A, B) = 1$  implica que todos los elementos de  $B$  son débilmente dominados por elementos de  $A$ , lo cual se puede interpretar como que el conjunto de aproximación  $B$  es “cubierto” por  $A$ . Por otro lado, el valor  $\mathcal{C}(A, B) = 0$  implica que ninguno de los elementos de  $B$  es dominado por elementos de  $A$ . En general, al utilizar este indicador de desempeño es necesario calcular tanto  $\mathcal{C}(A, B)$  como  $\mathcal{C}(B, A)$ , ya que las intersecciones de ambos conjuntos pueden no ser vacías.

## 6.3 | RESULTADOS

La validación de las funciones de escalarización generadas se llevó a cabo resolviendo los 23 problemas de prueba previamente descritos (DTLZ1-DTLZ7, WFG1-WFG9 y IDTLZ1-IDTLZ7). Cada uno de estos se resolvió para 2, 3, 4, 5, 6 y 10 funciones objetivo, generando un total de 138 problemas resueltos. Esto con el fin de observar la escalabilidad de las funciones de escalarización. Para resolver cada problema se utilizó la heurística MOMBI-II, variando únicamente la función de escalarización utilizada. Esto es debido a que no se pretende comparar el comportamiento de dos o más AEMOs, sino únicamente el impacto que tiene la función de escalarización en la calidad de las soluciones obtenidas. Las pruebas que realizamos consideraron tres funciones de escalarización:  $ASF$ ,  $F4A$  y  $F4B$ . La función utilizada por defecto en MOMBI-II es  $ASF$ , por lo que fue el punto de comparación que utilizamos para medir el desempeño de las otras dos funciones, las cuales

fueron generadas mediante la implementación descrita en el capítulo anterior. Para cada uno de los problemas mencionados se realizaron 30 ejecuciones de MOMBI-II para obtener 30 conjuntos de aproximación del frente de Pareto de cada problema. Para cada una de éstas realizamos el cálculo de los tres indicadores de desempeño previamente descritos y obtuvimos el valor promedio y varianza de cada indicador. Adicionalmente, utilizamos la prueba de suma de rangos de Wilcoxon para determinar si las distribuciones de cada función probada son idénticas o distintas con un intervalo de confianza de 95 %.

En esta sección presentamos los resultados obtenidos para cada uno de estos indicadores en los problemas de prueba. Las tablas completas de cada indicador en función del número de objetivos se muestran en el apéndice D, mientras que las tablas con los valores-p de la prueba de suma de rangos de Wilcoxon se encuentran en el apéndice E.

### 6.3.1 HIPERVOLUMEN

Para realizar el cálculo del hipervolumen se utilizó, para cada problema, el punto de referencia generado por las peores coordenadas obtenidas de la unión de todos los conjuntos de aproximación obtenidos. En el apéndice C se muestran los puntos de referencia utilizados para cada problema. En las figuras 6.24 y 6.25 se muestra gráficamente el número de problemas mejorados por cada función en términos del número de objetivos y la familia de problemas de prueba. Decimos que una función de escalarización mejora el desempeño de otra cuando mejora tanto el valor promedio de hipervolumen como el valor dado por la prueba de suma de rangos de Wilcoxon usando un nivel de significancia del 5 %. Por otro lado, los problemas que se muestran con un comportamiento similar son aquellos en los que a pesar de haber una función con un mejor desempeño en el promedio, dicho valor no es respaldado por la prueba de Wilcoxon. Considerando los 23 problemas de prueba en las 6 dimensiones que se probaron, la función *ASF* mejoró el desempeño de 29.71 % de los problemas con respecto a *F4A*. De forma análoga, *F4A* mejoró el desempeño de 36.23 % de los problemas con respecto a *ASF*. El 34.06 % restante son problemas en los que el comportamiento de ambas funciones es similar. En el caso de la función *F4B*, ésta mejoró 55.07 % de los problemas con respecto a *ASF*. Análogamente *ASF* mejoró 7.25 % de los problemas con respecto a *F4B*. Los problemas en los que las funciones se comportaron de forma similar representan el 37.68 %. A partir de estos datos podemos observar que el comportamiento de *F4A* en los problemas *DTLZ* y *WFG* es similar al de *ASF*, aunque logra mejo-

rar considerablemente en los problemas IDTLZ. Por otro lado, la función *F4B* logra mejorar el desempeño de *ASF* en la mayoría de los problemas utilizados. No obstante, en ambas comparaciones se puede observar que el desempeño de las funciones se deteriora conforme aumenta el número de objetivos en los problemas de prueba.

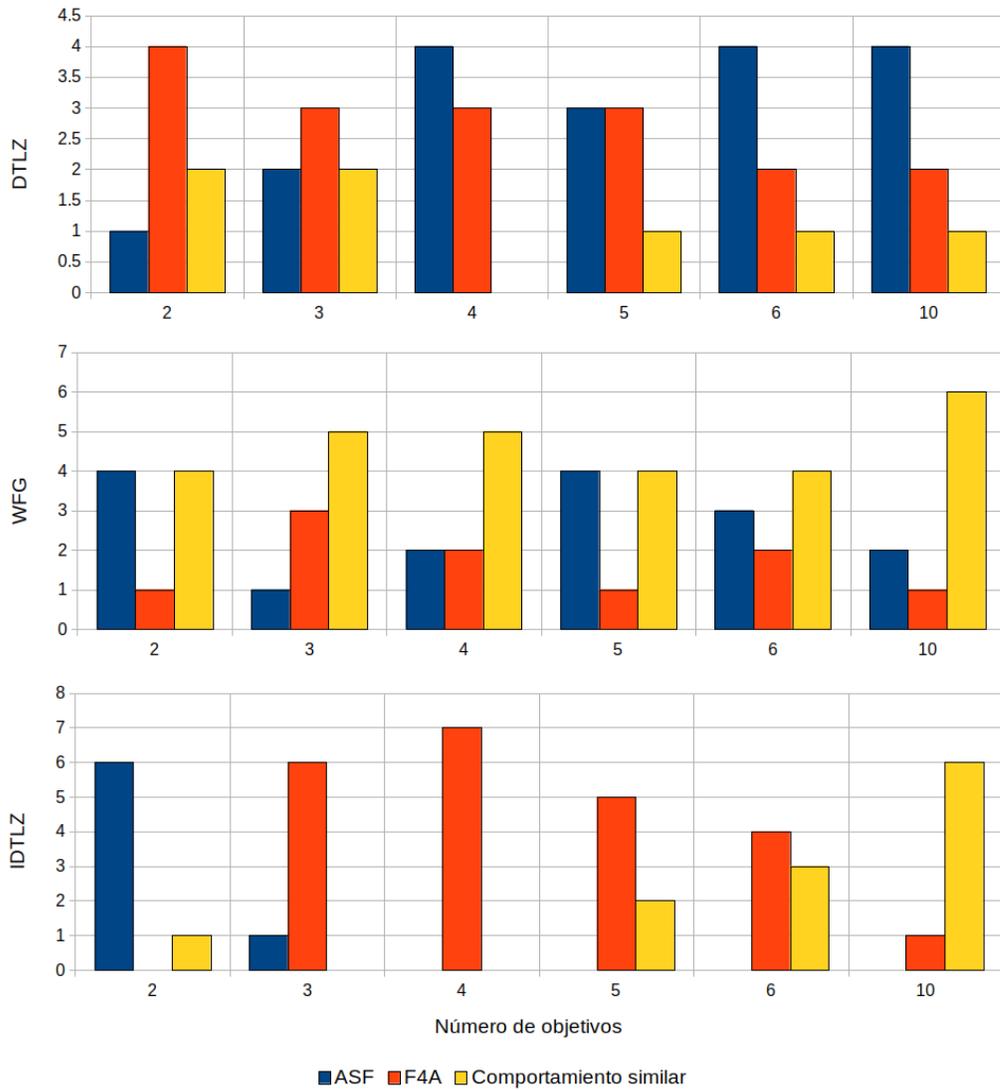


Figura 6.24: Comparación de problemas mejorados por las funciones *F4A* y *ASF* en relación a la familia de problemas de prueba y el número de objetivos usando el indicador de hipervolumen.

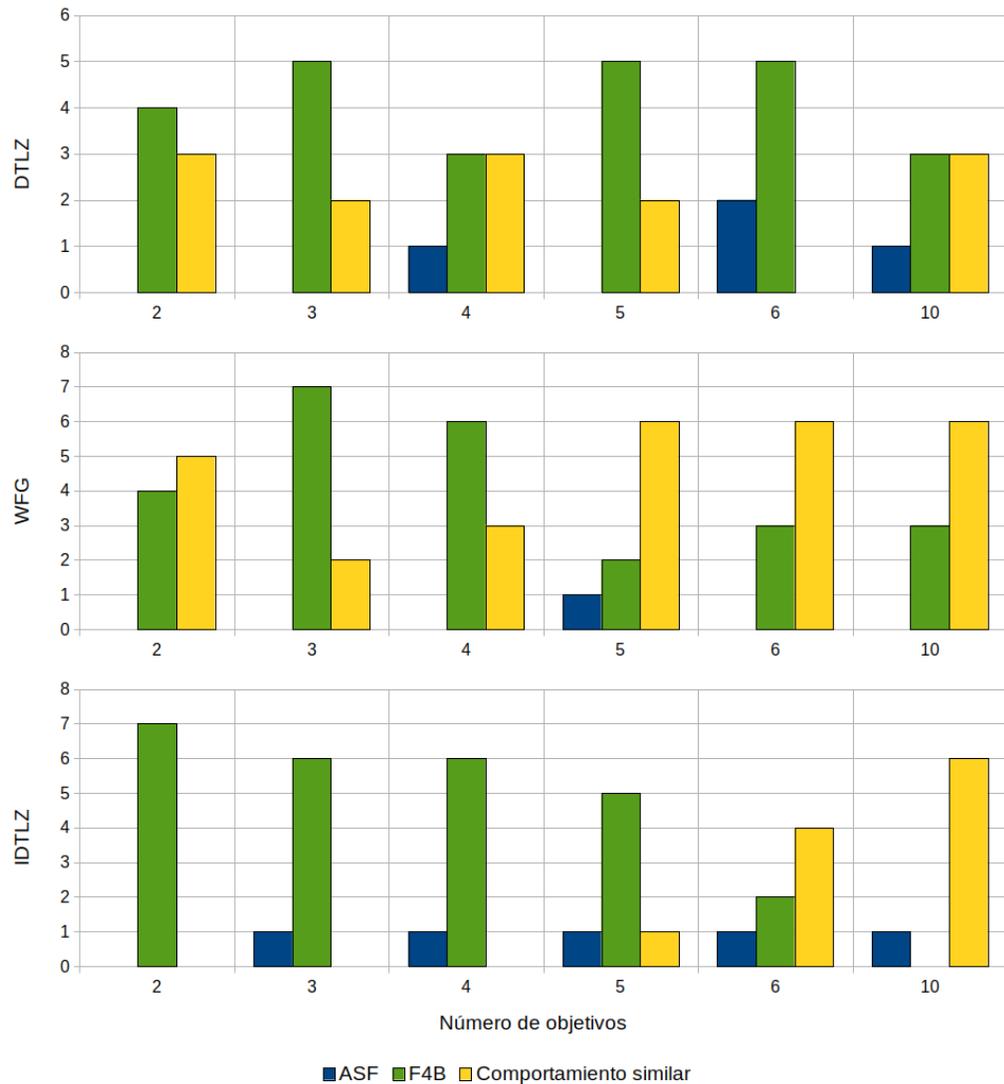


Figura 6.25: Comparación de problemas mejorados por las funciones  $F4B$  y  $ASF$  en relación a la familia de problemas de prueba y el número de objetivos usando el indicador de hipervolumen.

### 6.3.2 ENERGÍA-S

En las figuras 6.26 y 6.27 se muestra gráficamente el número de problemas mejorados por cada función en términos del número de objetivos y la familia de problemas de prueba. El criterio para determinar si una función de escalarización mejora el desempeño de otra es el mismo que el utilizado

para el indicador de hipervolumen. La única diferencia al respecto es que en este indicador se considera como una mejora al valor más pequeño, ya que la minimización de este indicador implica una mejor distribución de los puntos en el espacio objetivo. Considerando los 23 problemas de prueba en las 6 dimensiones que se probaron, la función *ASF* mejoró el desempeño de 12.32 % problemas respecto a *F4A*. De forma análoga, *F4A* mejoró el desempeño de 21.74 % problemas respecto a *ASF*. El 65.94 % restante son problemas en los que el comportamiento de ambas funciones es similar. En el caso de la función *F4B*, ésta mejoró 18.84 % de los problemas con respecto a *ASF*. Análogamente, *ASF* mejoró 4.35 % de los problemas con respecto a *F4B*. Los problemas en los que las funciones se comportaron de forma similar representan el 76.81 %. Usando este indicador podemos observar que las mejoras son menos significativas que usando el indicador de hipervolumen. Además, en este caso la función *F4A* obtiene un porcentaje de mejora mayor al de la función *F4B*. Por otro lado, podemos observar la misma tendencia a deteriorar su desempeño conforme aumenta el número de funciones objetivo.

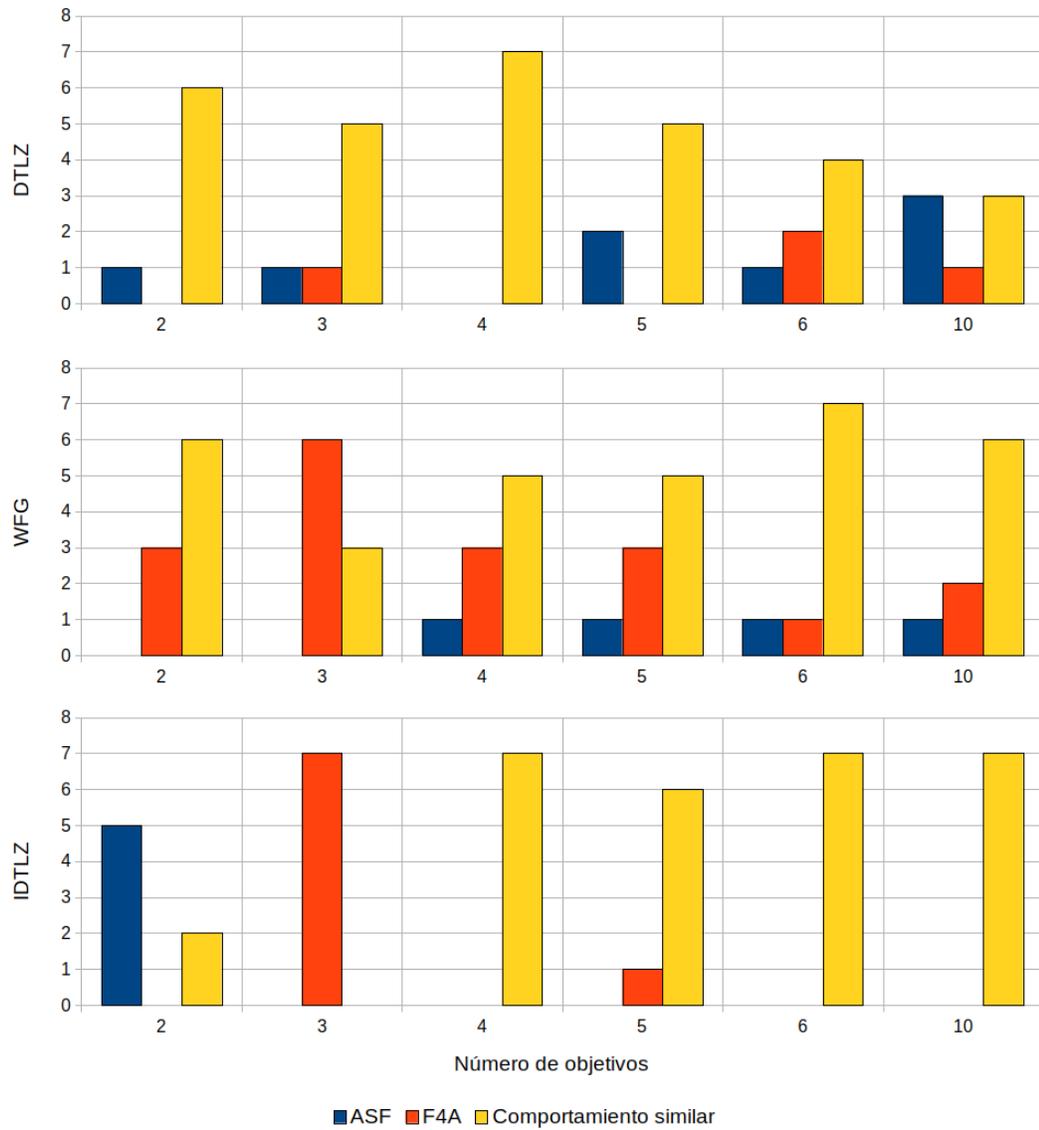


Figura 6.26: Comparación de problemas mejorados por las funciones  $F4A$  y  $ASF$  en relación a la familia de problemas de prueba y el número de objetivos usando el indicador de energía-s.

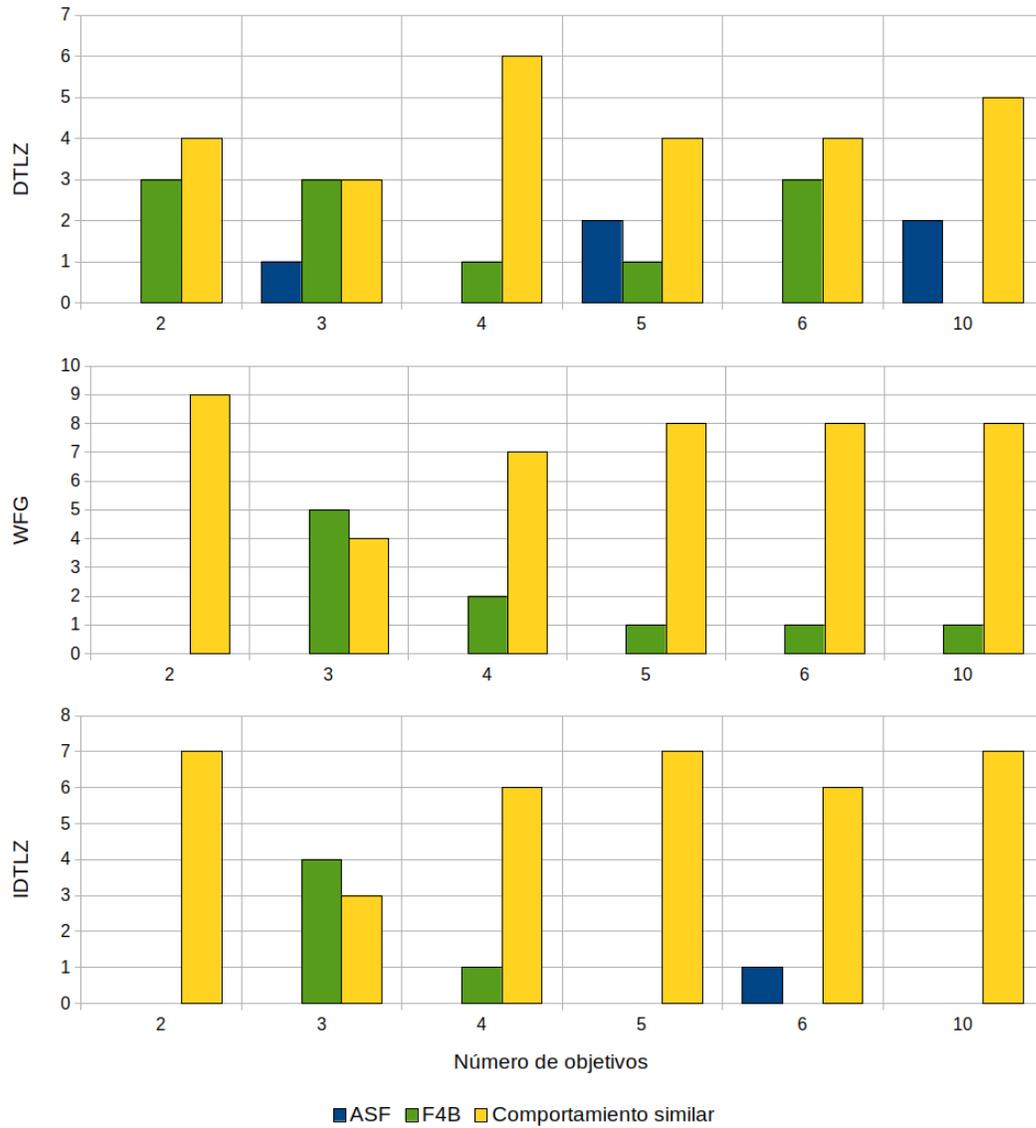


Figura 6.27: Comparación de problemas mejorados por las funciones F4B y ASF en relación a la familia de problemas de prueba y el número de objetivos usando el indicador de energía-s.

### 6.3.3 COBERTURA DE CONJUNTOS

En la figura 6.28 se muestra el promedio de los porcentajes de mejora obtenidos por cada función de escalarización. Como se puede observar, el desempeño de las tres funciones es bastante similar, ya que aunque existe

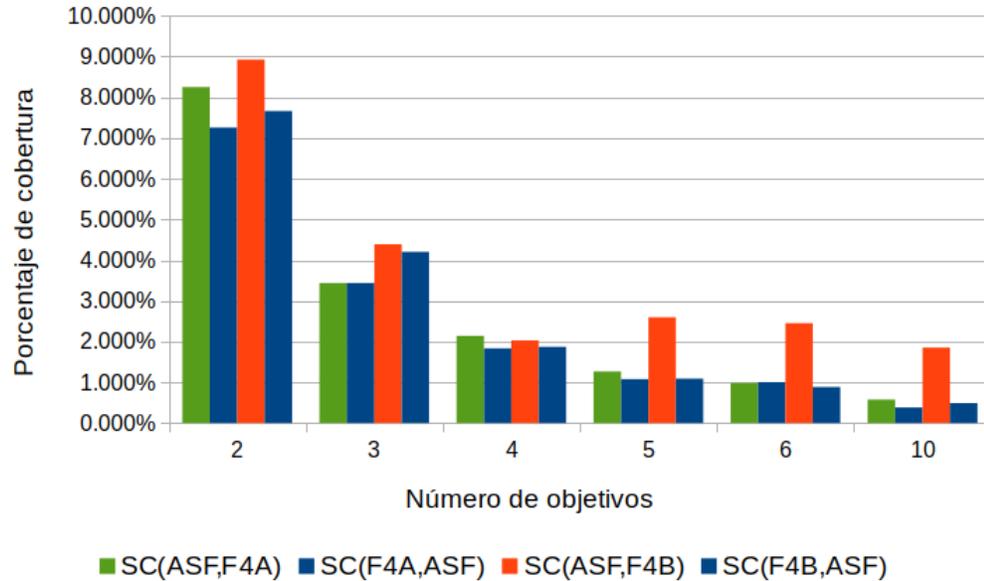


Figura 6.28: Comparación de funciones  $F4A$ ,  $F4B$  y  $ASF$  en función del número de objetivos usando el indicador de cobertura de conjuntos.

una ligera mejora en el porcentaje de puntos de  $ASF$  dominados por  $F4A$  y  $F4B$ , la fracción de puntos dominados en el sentido contrario es bastante similar. Por lo tanto, no es posible asegurar que la convergencia de alguna de las funciones mejore de forma evidente al de las demás. Por otro lado, una vez más es claro que existe una tendencia al aumentar el número de objetivos. Sin embargo, este comportamiento es esperable utilizando este indicador, ya que, conforme se aumenta la dimensionalidad del espacio objetivo probabilísticamente se vuelve menos común que un punto domine a otro.

## 6.4 | RESUMEN

En este capítulo se describieron los elementos necesarios para llevar a cabo la validación experimental de las nuevas funciones  $F4A$  y  $F4B$ . Adicionalmente, se describió la metodología llevada a cabo para realizar dicha validación, así como los resultados obtenidos al utilizar los indicadores de hipervolumen, energía-s y cobertura de conjuntos para medir la calidad de convergencia y distribución de puntos logrados por las funciones evaluadas. El análisis de los datos muestran que la función  $F4A$  tiene un desem-

peño cuantitativamente similar al de  $ASF$  en la solución de los problemas de prueba utilizados, a pesar de la clara diferencia en su definición. Por otro lado, la función  $F4B$  logra mejorar la mayoría de problemas de prueba con respecto a  $ASF$ . Cabe hacer notar que ambas funciones logran una mayor mejora en el indicador de hipervolumen que en el de energía-s o en el de cobertura de conjuntos. Esto sugiere que la mejora de desempeño observada se debe más a una mejor convergencia en los frentes de Pareto obtenidos que a una mejor distribución de las soluciones (aunque ésta también se ve mejorada ligeramente).



## CONCLUSIONES Y TRABAJO FUTURO

En este trabajo utilizamos distintas técnicas para lograr la generación automática de funciones de escalarización, utilizadas en la solución de problemas de optimización objetivo.

Uno de los productos de este trabajo es la implementación de software que permite dicha generación. Esta implementación utiliza un algoritmo de Programación Genética (ELGP) para la creación de nuevas funciones y un algoritmo evolutivo multi-objetivo (MOMBI-II) para la validación de las funciones creadas. El resultado es un algoritmo que permite la generación de distintas funciones de escalarización de forma automática. Una vez obtenida la implementación de dicho algoritmo realizamos distintos experimentos para obtener funciones de escalarización que tuvieran un comportamiento similar al de las funciones de escalarización comúnmente usadas en la literatura. En estas ejecuciones se utilizaron distintos parámetros para observar el comportamiento de las funciones de escalarización obtenidas. Algunos de los parámetros probados más relevantes fueron los siguientes: características del problema de optimización multi-objetivo de prueba resuelto por MOMBI-II (número de objetivos y geometría del frente de Pareto del problema de prueba) y el tipo de operadores permitidos en el proceso de generación de las funciones (aritméticos o trigonométricos). A partir de las múltiples pruebas realizadas obtuvimos distintas funciones de escalarización. Posteriormente comparamos su desempeño usando una serie de problemas de prueba para observar si lograban generalizar el desempeño obtenido durante el proceso de generación a distintos problemas. De las pruebas obtenidas observamos que el mejor desempeño se obtuvo con los

parámetros mostrados en el apéndice B. Además, observamos que la aplicación del operador de valor absoluto a las funciones generadas es fundamental para mejorar su desempeño.

Un segundo resultado de este trabajo son las dos funciones de escalarización ( $F4A$  y  $F4B$ ) obtenidas en una de las múltiples ejecuciones realizadas, específicamente la ejecución utilizando los parámetros mencionados. Dichas funciones fueron evaluadas experimentalmente para comparar su comportamiento en un conjunto de problemas de optimización multi-objetivo de prueba en comparación con la función de escalarización  $ASF$ .

Uno de los comportamientos observados en dicha validación es que el desempeño de las funciones de escalarización se deteriora conforme aumenta el número de objetivos en los problemas de optimización multi-objetivo. Sin embargo, este comportamiento puede explicarse con el parámetro de número de evaluaciones utilizado para resolver los problemas, el cual permaneció fijo en todos los problemas de prueba.

Con los datos obtenidos en la validación experimental realizada podemos observar que la función  $F4A$  logra un desempeño similar al de  $ASF$ , mientras que  $F4B$  logra mejorar el desempeño de  $ASF$ . A partir de esto, podemos afirmar que la hipótesis inicial es correcta: es posible utilizar Programación Genética para la generación automática de funciones de escalarización que sean competitivas respecto a funciones comúnmente utilizadas en la literatura.

Como trabajo futuro existen diversas posibilidades ya que, hasta donde tenemos conocimiento y de acuerdo a la revisión de la literatura realizada, no existen trabajos en el área que hayan utilizado un enfoque similar para la generación de nuevas funciones de escalarización.

En primer lugar se pueden realizar ejecuciones de la implementación propuesta utilizando un mayor número de iteraciones, ya que en los experimentos realizados se realizaron un máximo de 50 iteraciones. La razón principal de esto es el alto costo computacional del programa, pero consideramos probable que aumentando el número de iteraciones puedan obtenerse mejores funciones de escalarización.

Por otro lado, se puede explorar el uso de distintos operadores matemáticos para formar las funciones, ya que en los experimentos realizados utilizamos principalmente operadores aritméticos. Adicionalmente, realizamos un experimento con operadores trigonométricos. Sin embargo, debido al mayor costo computacional de este tipo de operadores utilizamos un menor número de iteraciones, y la aptitud de las funciones generadas resultó menor que la obtenida con operadores aritméticos.

Otra posibilidad es utilizar la solución de un problema de optimización

---

multi-objetivo distinto en el proceso de validar las funciones generadas automáticamente. En la mayoría de los experimentos realizados utilizamos el problema DTLZ4 con 3 objetivos para verificar la calidad de cada función. También realizamos experimentos con el problema DTLZ7. Sin embargo, observamos que las funciones obtenidas tenían una sobre-especificación, por lo que no generalizaban correctamente a problemas con frentes de Pareto con distintas geometrías. No obstante, queda abierta la posibilidad de utilizar un problema distinto o un número distinto de objetivos para intentar mejorar el decremento de desempeño al aumentar el número de objetivos.

Finalmente, es posible modificar la implementación para la generación de otros elementos utilizados por distintos algoritmos evolutivos multi-objetivo. Por ejemplo, se podría utilizar para la búsqueda de nuevos indicadores de desempeño que permitan cuantificar la calidad de los frentes de Pareto obtenidos.



# A



## PARÁMETROS DE CONFIGURACIÓN

### A.1 | PARÁMETROS DE CONFIGURACIÓN DE ELGP

Parámetro	Valor predeterminado	Descripción
<i>Parámetros generales</i>		
g	100	Número de generaciones
popsiz	500	Tamaño de población
limit_evals	0	Límite de evaluaciones en lugar de número de generaciones
max_evals	0	Número máximo de evaluaciones
<i>Parámetros de generación</i>		
sel	1	Tipo de selección: (1) torneo, (2) crowding determinístico, (3) Selección lexicográfica 4: algoritmo de edad-aptitud

PS_sel	1	Criterios para supervivencia: (1) edad+aptitud, (2) edad+aptitud+generalidad, (3) edad+aptitud+complejidad, (4) aptitudes de clases, (5) aptitudes de clases + edad
tourn_size	2	Número de individuos por torneo
rt_rep	0	Porcentaje de reproducción
rt_cross	0.8	Porcentaje de cruza

APÉNDICE A. PARÁMETROS DE CONFIGURACIÓN

rt_mut	0.2	Porcentaje de mutación
cross	3	Tipo de cruza: (1) ultra, (2) un punto, (3) subárbol
mutate	2	Tipo de mutación: (1) mutación de un punto, (2) mutación de subárbol
cross_ar	0.025	Porcentaje de alternación en cruza (únicamente para ultra)
mut_ar	0.025	Porcentaje de alternación de mutación
align_dev	0	Agregar desviación de alineación gaussiana a la cruza
elitism	0	Conservar al mejor individuo en cada generación
stop_condition	1	Terminar la ejecución cuando aptitud <1e-6
init_validate_on	0	Validación inicial de aptitud de la población inicial
<i>Parámetros de datos</i>		
train	0	Dividir conjunto de datos en subconjuntos de entrenamiento y validación
train_pct	0.5	Porcentaje de datos usados para entrenamiento
shuffle_data	0	Barajar conjunto de datos antes de seleccionar subconjuntos de entrenamiento y validación
pop_restart	0	Reiniciar ejecución a partir de una población previa
pop_restart_path	""	Ruta del archivo con la población usada para reiniciar ejecución
<i>Parámetros de resultados</i>		
resultspath	""	Ruta en la que se almacenan los resultados
print_every_pop	0	Almacenar en un archivo la población de cada generación
print_genome	0	Imprimir genoma para visualización en paraview
print_novelty	0	Imprimir número de vectores de salida únicos

A.1. PARÁMETROS DE CONFIGURACIÓN DE ELGP

print_homology	0	Imprimir homología genética en los programas
num_log_pts	0	Número de puntos de registro a imprimir (0 imprime cada generación)
<i>Parámetros de clasificación</i>		
classification	0	Define un problema de clasificación en lugar de uno de regresión
class_bool	0	Interpreta etiquetas de clases como la conversión bit-string de una pila booleana
class_m3gp	0	Utiliza aptitud de clasificación basada en distancia mahalanobis
class_prune	0	Poda las dimensiones del mejor individuo en cada generación
<i>Información del problema</i>		
intvars	none	Variables en el archivo de datos a utilizar en los programas
cvals	none	Semilla para la población inicial con determinados valores constantes
seeds	none	Semilla para soluciones parciales p.e. (x+y)
AR	0	Incluir variables de salida auto-regresivas
AR_n	1	Orden de auto-regresión
AR_lookahead	0	Predice únicamente una salida adelantada
ERC	1	Constantes aleatorias efímeras
ERCints	0	Hace que las variables de ERC sean enteros en lugar de floats
maxERC	1	
minERC	-1	
numERC	1	
<i>Parámetros de aptitud</i>		
fit_type	1	Tipo de aptitud: (1) error absoluto medio, (2) corr, (3) combo, (4) VAF
norm_error	0	Normalizar error considerando la desviación estandar de los datos objetivo

APÉNDICE A. PARÁMETROS DE CONFIGURACIÓN

max_fit	1.00E+20	Aptitud máxima posible
min_fit	1.00E-20	Aptitud mínima posible
estimate_fitness	0	Coevolucionar estimadores de aptitud
FE_pop_size	0	Tamaño de población del estimador de aptitud
FE_ind_size	0	Número de casos de aptitud para el estimador de aptitud
FE_train_size	0	Tamaño de la población de entrenamiento
FE_train_gens	0	Número de generaciones entre selecciones de entrenamiento
FE_rank	0	Usar rango para estimador de aptitud en lugar de error
estimate_generality	0	Estimar qué tan bien generalizan las soluciones utilizando la porción de validación del estimador de aptitud
G_sel	0	Tipo de aptitud a utilizar en el test de generalidad
G_shuffle	0	Barajar datos cada generación
op_list	n v + - * /	Operadores disponibles: n v + - * / sin cos log exp sqrt = ! <= >= if-then if-then-else &
weight_ops_on	0	Ponderar operadores
op_weight	empty	Ponderación de cada operador especificado en op_list
min_len	3	Longitud mínima del programa
max_len	20	Longitud máxima del programa
max_len_init	max_len	Especificar una longitud máxima distinta para la población inicial
init_trees	0	Inicializar genotipos como árboles sintácticamente válidos en lugar de pilas aleatorias
complex_measure	2	Medida de complejidad: (1) tamaño de genotipo, (2) tamaño simbólico, (3) tamaño efectivo de genotipo
<i>Parámetros de Hill-climbing</i>		

A.1. PARÁMETROS DE CONFIGURACIÓN DE ELGP

pHC_on	0	Utilizar hill-climbing en cada generación
pHC_its	1	Número de iteraciones
<i>Parámetros epigenéticos</i>		
eHC_on	0	Hill-climbing epigenético
eHC_its	1	Número de iteraciones
eHC_prob	0.1	Probabilidad de cambiar un gene
eHC_init	0.5	Porcentaje de genes expresados en genotipos iniciales
eHC_slim	0	Minimizar evaluaciones de punto tanto como sea posible
eHC_mut	0	Realizar mutación en lugar de hill-climbing
<i>Parámetros del archivo de Pareto</i>		
prto_arch_on	0	
prto_arch_size	20	
<i>Modelo de islas</i>		
islands	0	Usar múltiples poblaciones de islas, una por núcleo
island_gens	100	Número de generaciones antes de revolver las poblaciones de las islas
<i>Parámetros de casos léxicos</i>		
lexpool	1	Fracción de población a usar en eventos de selección por casos léxicos
lex_class	0	Usar aptitudes de clases distintas como casos en un problema de clasificación
lex_metacases	none	Especificar casos adicionales para selección. Opciones: edad, complejidad
lex_eps_std	0	Utilizar casos léxicos épsilon con eps = desviación estandar del error
lex_eps_error	0	Utilizar casos léxicos épsilon con épsilons basados en error
lex_eps_target	0	Utilizar casos léxicos épsilon con épsilons basados en objetivo
lex_eps_target_mad	0	Utilizar casos léxicos épsilon con desviación media absoluta (basados en objetivo)

lex_eps_error_mad	0	Utilizar casos léxicos épsilon con desviación media absoluta (basados en error)
lex_epsilon	0.1	Valor de épsilon (ignorado para versiones mad y std)

## A.2 | PARÁMETROS DE CONFIGURACIÓN DE MOMBI-II

Parámetro	Descripción
seed	Ruta de archivo con semillas para el generador de números
nobj	Número de funciones objetivo
nvar	Número de variables de decisión
nvar_wfg	Número de variables de decisión para funciones WFG
npos	Número de parámetros relacionados a posición
psize	Tamaño de población
pc	Probabilidad de cruce
pm	Probabilidad de mutación
nc	Índice de distribución de cruce (cruce SBX)
nm	Índice de distribución de mutación (mutación real polinomial)
feval	Número máximo de evaluaciones de función
time	Tiempo máximo de ejecución
fitness	Aptitud máxima
output	Ruta de directorio de salida
plot	Gráfica para 2 y 3 objetivos: (0) deshabilitado, (1) iterativo, (2) al final
debug	Habilitar depuración
term	Tipo de terminal en Gnuplot (wxt, x11, etc.)
wfile	Ruta de archivo con vectores de pesos
utility	Función de escalarización para MOEA/D

## A.3 | EJEMPLO DE ARCHIVO DE PESOS PARA MOMBI-II

```

1 # 105 3
2 1.000000e+00 0.000000e+00 0.000000e+00
3 9.230769e-01 7.692308e-02 0.000000e+00
4 9.230769e-01 0.000000e+00 7.692308e-02

```

### A.3. EJEMPLO DE ARCHIVO DE PESOS PARA MOMBI-II

---

```
5 8.461538e-01 1.538462e-01 0.000000e+00
6 8.461538e-01 7.692308e-02 7.692308e-02
7 8.461538e-01 0.000000e+00 1.538462e-01
8 7.692308e-01 2.307692e-01 0.000000e+00
9 7.692308e-01 1.538462e-01 7.692308e-02
10 7.692308e-01 7.692308e-02 1.538462e-01
11 7.692308e-01 0.000000e+00 2.307692e-01
12 6.923077e-01 3.076923e-01 0.000000e+00
13 6.923077e-01 2.307692e-01 7.692308e-02
14 6.923077e-01 1.538462e-01 1.538462e-01
15 6.923077e-01 7.692308e-02 2.307692e-01
16 6.923077e-01 0.000000e+00 3.076923e-01
17 6.153846e-01 3.846154e-01 0.000000e+00
18 6.153846e-01 3.076923e-01 7.692308e-02
19 6.153846e-01 2.307692e-01 1.538462e-01
20 6.153846e-01 1.538462e-01 2.307692e-01
21 6.153846e-01 7.692308e-02 3.076923e-01
22 6.153846e-01 0.000000e+00 3.846154e-01
23 5.384615e-01 4.615385e-01 0.000000e+00
24 5.384615e-01 3.846154e-01 7.692308e-02
25 5.384615e-01 3.076923e-01 1.538462e-01
26 5.384615e-01 2.307692e-01 2.307692e-01
27 5.384615e-01 1.538462e-01 3.076923e-01
28 5.384615e-01 7.692308e-02 3.846154e-01
29 5.384615e-01 0.000000e+00 4.615385e-01
30 4.615385e-01 5.384615e-01 0.000000e+00
31 4.615385e-01 4.615385e-01 7.692308e-02
32 4.615385e-01 3.846154e-01 1.538462e-01
33 4.615385e-01 3.076923e-01 2.307692e-01
34 4.615385e-01 2.307692e-01 3.076923e-01
35 4.615385e-01 1.538462e-01 3.846154e-01
36 4.615385e-01 7.692308e-02 4.615385e-01
37 4.615385e-01 0.000000e+00 5.384615e-01
38 3.846154e-01 6.153846e-01 0.000000e+00
39 3.846154e-01 5.384615e-01 7.692308e-02
40 3.846154e-01 4.615385e-01 1.538462e-01
41 3.846154e-01 3.846154e-01 2.307692e-01
42 3.846154e-01 3.076923e-01 3.076923e-01
43 3.846154e-01 2.307692e-01 3.846154e-01
44 3.846154e-01 1.538462e-01 4.615385e-01
45 3.846154e-01 7.692308e-02 5.384615e-01
46 3.846154e-01 0.000000e+00 6.153846e-01
47 3.076923e-01 6.923077e-01 0.000000e+00
48 3.076923e-01 6.153846e-01 7.692308e-02
49 3.076923e-01 5.384615e-01 1.538462e-01
50 3.076923e-01 4.615385e-01 2.307692e-01
51 3.076923e-01 3.846154e-01 3.076923e-01
52 3.076923e-01 3.076923e-01 3.846154e-01
```

APÉNDICE A. PARÁMETROS DE CONFIGURACIÓN

---

53	3.076923e-01	2.307692e-01	4.615385e-01
54	3.076923e-01	1.538462e-01	5.384615e-01
55	3.076923e-01	7.692308e-02	6.153846e-01
56	3.076923e-01	0.000000e+00	6.923077e-01
57	2.307692e-01	7.692308e-01	0.000000e+00
58	2.307692e-01	6.923077e-01	7.692308e-02
59	2.307692e-01	6.153846e-01	1.538462e-01
60	2.307692e-01	5.384615e-01	2.307692e-01
61	2.307692e-01	4.615385e-01	3.076923e-01
62	2.307692e-01	3.846154e-01	3.846154e-01
63	2.307692e-01	3.076923e-01	4.615385e-01
64	2.307692e-01	2.307692e-01	5.384615e-01
65	2.307692e-01	1.538462e-01	6.153846e-01
66	2.307692e-01	7.692308e-02	6.923077e-01
67	2.307692e-01	0.000000e+00	7.692308e-01
68	1.538462e-01	8.461538e-01	0.000000e+00
69	1.538462e-01	7.692308e-01	7.692308e-02
70	1.538462e-01	6.923077e-01	1.538462e-01
71	1.538462e-01	6.153846e-01	2.307692e-01
72	1.538462e-01	5.384615e-01	3.076923e-01
73	1.538462e-01	4.615385e-01	3.846154e-01
74	1.538462e-01	3.846154e-01	4.615385e-01
75	1.538462e-01	3.076923e-01	5.384615e-01
76	1.538462e-01	2.307692e-01	6.153846e-01
77	1.538462e-01	1.538462e-01	6.923077e-01
78	1.538462e-01	7.692308e-02	7.692308e-01
79	1.538462e-01	0.000000e+00	8.461538e-01
80	7.692308e-02	9.230769e-01	0.000000e+00
81	7.692308e-02	8.461538e-01	7.692308e-02
82	7.692308e-02	7.692308e-01	1.538462e-01
83	7.692308e-02	6.923077e-01	2.307692e-01
84	7.692308e-02	6.153846e-01	3.076923e-01
85	7.692308e-02	5.384615e-01	3.846154e-01
86	7.692308e-02	4.615385e-01	4.615385e-01
87	7.692308e-02	3.846154e-01	5.384615e-01
88	7.692308e-02	3.076923e-01	6.153846e-01
89	7.692308e-02	2.307692e-01	6.923077e-01
90	7.692308e-02	1.538462e-01	7.692308e-01
91	7.692308e-02	7.692308e-02	8.461538e-01
92	7.692308e-02	0.000000e+00	9.230769e-01
93	0.000000e+00	1.000000e+00	0.000000e+00
94	0.000000e+00	9.230769e-01	7.692308e-02
95	0.000000e+00	8.461538e-01	1.538462e-01
96	0.000000e+00	7.692308e-01	2.307692e-01
97	0.000000e+00	6.923077e-01	3.076923e-01
98	0.000000e+00	6.153846e-01	3.846154e-01
99	0.000000e+00	5.384615e-01	4.615385e-01
100	0.000000e+00	4.615385e-01	5.384615e-01

*A.3. EJEMPLO DE ARCHIVO DE PESOS PARA MOMBI-II*

---

```
101 0.000000e+00 3.846154e-01 6.153846e-01
102 0.000000e+00 3.076923e-01 6.923077e-01
103 0.000000e+00 2.307692e-01 7.692308e-01
104 0.000000e+00 1.538462e-01 8.461538e-01
105 0.000000e+00 7.692308e-02 9.230769e-01
106 0.000000e+00 0.000000e+00 1.000000e+00
```

Código A.1: Archivo de pesos para 3 dimensiones



# B



## PARÁMETROS UTILIZADOS PARA LA GENERACIÓN DE NUEVAS FUNCIONES DE ESCALARIZACIÓN

### B.1 | ARCHIVO DE CONFIGURACIÓN DE ELGP

```
1 g                50 // number of generations
2 popsize          30 //population size
3 numits           1 // number of trials
4
5 // Generation Settings
6 sel              2 // 1: tournament 2: deterministic crowding 3:
   lexicase selection 4: age-fitness pareto algorithm
7 tourn_size      5
8 rt_rep          0.1
9 rt_cross        0.7
10 rt_mut          0.2
11 cross           1 // 1: ultra 2: one point
12 cross_ar       .15 //crossover alternation rate (ultra only)
13 mut_ar         .1 //mutation alternation rate (ultra only)
14 stoperror      0.0001 // stop condition / convergence
   condition
15
16 init_validate_on 1 // initial fitness validation of starting
   population
17 train          1 // split data into training and
   validation sets
18 train_ratio     0.5 // amount of data to be used for training
19
```



```
67 eHC_cluster      1
68 eHC_dev          1
69 eHC_best         1
70 eHC_init         .75 //0: start with true epilines, 1: start
    with a mix of on/off values
71 eHC_prob_scale   1
72 eHC_max_prob     .75
73 eHC_min_prob     0.25
74
75 // Pareto settings
76 prto_arch_on     1
77 prto_arch_size   425
78 prto_sel_on      1
79
80 // island model
81 islands          0
82 island_gens      82
```

Código B.1: Archivo de configuración de ELGP

## B.2 | ARCHIVO DE CONFIGURACIÓN DE MOMBI-II

```
1 #####
2 # Param_02D.cfg #
3 # #
4 # Configuration file #
5 #####
6
7 # File that contains the seeds of the random number generator
8 seed = ./input/seed.dat
9
10 # Number of objective functions
11 nobj = 2
12
13 # Number of decision variables (0 default values)
14 nvar = 0
15
16 # Number of decision variables (WFG)
17 nvar_wfg = 24
18
19 # Number of position-related parameters (WFG test suite)
20 npos = 4
21
22 # Population size
23 psize = 100
24
25 # Crossover probability
```

*APÉNDICE B. PARÁMETROS UTILIZADOS PARA LA GENERACIÓN DE NUEVAS FUNCIONES DE ESCALARIZACIÓN*

---

```
26 pc = 1.0
27
28 # Mutation probability (by default -1 is equivalent to Pm = 1/N
   )
29 pm = -1
30
31 # Crossover distribution index (SBX)
32 nc = 20
33
34 # Mutation distribution index (Real polynomial mutation)
35 nm = 20
36
37 ##### Stopping condition #####
38 # This section defines the stopping condition of the program,
39 # given by the number of function evaluations, execution
40 # time (in seconds), fitness achieved (hv, R2, IGD, etc.)
41 # or a combination of them.
42
43 feval = 15000
44
45 ##### End of stopping condition #####
46
47 # Output directory
48 output = ./output
49
50 # Plots for 2 or 3 objectives (0 disabled, 1 iterative, 2 end)
51 plot = 0
52
53 # Debug (0 disabled, 1 enabled)
54 debug = 0
55
56 # Terminal type in Gnuplot (wxt, x11, etc.)
57 term = x11
58
59 # Weight vector file (NSGA-III, MOMBI2, R2-IBEA, MOEA/D)
60 wfile = ./input/weight/weight_02D_95.sld
61
62 # Utility function (MOEA/D)
63 utility = tchebycheff
64 niche = 20
65
66 # MOMBI2 parameters
67 max_hist = 5
68 epsilon = 1e-3
69 alpha = 0.5
70
71 # IBEA parameters
72 rho = 1.1
```

```
73 # original
74 kappa = 0.05
75 # R2
76 kappa_r2 = 0.005
77 #HyPE
78 samples = 10000
79 bound = 2
80 #MOVAP
81 xrs = 3
82 #BIGDA
83 rfile = axis_02D.txt
84
```

Código B.2: Archivo de configuración de MOMBI-II

*APÉNDICE B. PARÁMETROS UTILIZADOS PARA LA GENERACIÓN DE  
NUEVAS FUNCIONES DE ESCALARIZACIÓN*

---

# C



## PUNTOS DE REFERENCIA

En este apéndice se muestran las coordenadas de los puntos de referencia utilizados para calcular el indicador de hipervolumen en cada uno de los problemas de prueba utilizados durante la validación experimental. Estos puntos están compuestos con las peores coordenadas obtenidas en cada función objetivo en todas las ejecuciones realizadas para cada problema de prueba. Esto es con el fin de garantizar que, para cada problema, todos los puntos de los conjuntos de aproximación dominen al punto de referencia.

Problema	Punto de referencia
DTLZ1_2D	{0.50, 0.50}
DTLZ2_2D	{1.01, 1.01}
DTLZ3_2D	{1.01, 1.01}
DTLZ4_2D	{1.01, 1.01}
DTLZ5_2D	{1.01, 1.01}
DTLZ6_2D	{1.20, 1.20}
DTLZ7_2D	{0.85, 4.04}
WFG1_2D	{2.62, 3.91}
WFG2_2D	{2.01, 4.03}
WFG3_2D	{2.01, 4.02}
WFG4_2D	{2.05, 4.06}
WFG5_2D	{2.07, 4.08}
WFG6_2D	{2.09, 4.12}
WFG7_2D	{2.04, 4.06}
WFG8_2D	{2.31, 4.26}
WFG9_2D	{2.12, 4.14}
IDTLZ1:IDTLZ7_2D	{0, 0}

Tabla C.1: Puntos de referencia utilizados para problemas de prueba en 2 dimensiones.

APÉNDICE C. PUNTOS DE REFERENCIA

---

Problema	Punto de referencia
DTLZ1_3D	{0.50, 0.50, 0.50}
DTLZ2_3D	{1.02, 1.02, 1.03}
DTLZ3_3D	{1.01, 1.01, 1.02}
DTLZ4_3D	{1.01, 1.02, 1.03}
DTLZ5_3D	{0.72, 0.72, 1.02}
DTLZ6_3D	{0.94, 0.94, 1.19}
DTLZ7_3D	{0.89, 0.91, 6.06}
WFG_3D	{2.52, 4.69, 5.61}
WFG_3D	{1.77, 3.58, 6.07}
WFG_3D	{3.00, 2.11, 6.09}
WFG_3D	{2.10, 4.10, 6.12}
WFG_3D	{2.10, 4.13, 6.14}
WFG_3D	{2.12, 4.15, 6.18}
WFG_3D	{2.05, 4.08, 6.12}
WFG_3D	{2.28, 4.17, 6.28}
WFG_3D	{2.13, 4.14, 6.16}
IDTLZ1-IDTLZ7_3D	{0, 0, 0}

Tabla C.2: Puntos de referencia utilizados para problemas de prueba en 3 dimensiones.

Problema	Punto de referencia
DTLZ1_4D	{2.06, 0.99, 0.50, 0.57}
DTLZ2_4D	{1.04, 1.02, 1.06, 1.03}
DTLZ3_4D	{1.02, 7.34, 1.02, 2.89}
DTLZ4_4D	{1.02, 1.02, 1.02, 1.04}
DTLZ5_4D	{0.77, 0.77, 3.44, 1.01}
DTLZ6_4D	{3.55, 3.22, 10.56, 2.96}
DTLZ7_4D	{0.91, 0.93, 0.92, 8.33}
WFG1_4D	{2.53, 4.45, 4.22, 6.90}
WFG2_4D	{1.86, 3.47, 5.31, 8.08}
WFG3_4D	{2.93, 4.65, 5.69, 8.13}
WFG4_4D	{2.10, 4.12, 6.13, 8.16}
WFG5_4D	{2.12, 4.13, 6.15, 8.16}
WFG6_4D	{2.13, 4.16, 6.16, 8.21}
WFG7_4D	{2.07, 4.10, 6.11, 8.15}
WFG8_4D	{2.80, 4.26, 6.21, 8.18}
WFG9_4D	{2.15, 4.15, 6.17, 8.19}
IDTLZ1-IDTLZ7_4D	{0, 0, 0, 0}

Tabla C.3: Puntos de referencia utilizados para problemas de prueba en 4 dimensiones.

---

Problema	Punto de referencia
DTLZ1_5D	{2.51, 0.50, 1.02, 1.31, 0.86}
DTLZ2_5D	{1.04, 1.04, 1.03, 1.04, 1.09}
DTLZ3_5D	{1.04, 1.39, 1.03, 1.19, 1.24}
DTLZ4_5D	{1.05, 1.04, 1.03, 1.02, 1.04}
DTLZ5_5D	{0.77, 0.76, 3.36, 3.45, 1.02}
DTLZ6_5D	{1.97, 2.64, 8.65, 8.92, 3.99}
DTLZ7_5D	{0.91, 0.91, 0.91, 0.92, 10.16}
WFG1_5D	{2.48, 1.34, 1.77, 2.23, 8.17}
WFG2_5D	{1.97, 3.51, 5.33, 7.01, 10.10}
WFG3_5D	{2.97, 4.70, 6.76, 8.21, 10.11}
WFG4_5D	{2.10, 4.10, 6.13, 8.16, 10.19}
WFG5_5D	{2.12, 4.13, 6.15, 8.19, 10.19}
WFG6_5D	{2.17, 4.14, 6.17, 8.20, 10.24}
WFG7_5D	{2.07, 4.08, 6.11, 8.13, 10.17}
WFG8_5D	{3.02, 4.58, 6.15, 8.18, 10.22}
WFG9_5D	{2.18, 4.17, 6.20, 8.22, 10.25}
IDTLZ1-IDTLZ7_5D	{0, 0, 0, 0, 0}

Tabla C.4: Puntos de referencia utilizados para problemas de prueba en 5 dimensiones.

Problema	Punto de referencia
DTLZ1_6D	{0.50, 0.74, 1.38, 0.50, 0.50, 0.51}
DTLZ2_6D	{1.02, 1.08, 1.04, 1.03, 1.04, 1.03}
DTLZ3_6D	{1.03, 2.31, 1.15, 5.23, 1.32, 1.07}
DTLZ4_6D	{1.03, 1.05, 1.03, 1.04, 1.05, 1.03}
DTLZ5_6D	{0.87, 0.77, 3.19, 3.34, 3.45, 1.77}
DTLZ6_6D	{5.04, 7.49, 6.73, 7.35, 8.63, 4.32}
DTLZ7_6D	{0.91, 0.92, 0.91, 0.92, 0.91, 12.43}
WFG1_6D	{2.43, 2.58, 3.01, 2.04, 2.58, 5.58}
WFG2_6D	{1.89, 3.41, 4.81, 6.61, 7.68, 12.12}
WFG3_6D	{2.98, 4.58, 6.77, 8.93, 10.51, 12.25}
WFG4_6D	{2.10, 4.10, 6.13, 8.16, 10.19, 12.22}
WFG5_6D	{2.12, 4.12, 6.15, 8.18, 10.20, 12.23}
WFG6_6D	{2.15, 4.15, 6.18, 8.20, 10.26, 12.26}
WFG7_6D	{2.08, 4.09, 6.12, 8.13, 10.18, 12.19}
WFG8_6D	{3.03, 4.98, 6.40, 8.18, 10.19, 12.28}
WFG9_6D	{2.22, 4.22, 6.22, 8.24, 10.26, 12.27}
IDTLZ1-IDTLZ7_6D	{0, 0, 0, 0, 0, 0}

Tabla C.5: Puntos de referencia utilizados para problemas de prueba en 6 dimensiones.

Problema	Punto de referencia
DTLZ1_10D	{0.50, 0.49, 0.49, 0.50, 0.50, 0.58, 0.51, 0.50, 0.50, 0.53}
DTLZ2_10D	{1.04, 1.04, 1.03, 1.08, 1.03, 1.05, 1.03, 1.03, 1.02, 1.06}
DTLZ3_10D	{1.03, 1.03, 1.04, 1.08, 1.03, 1.04, 1.04, 1.16, 1.12, 1.97}
DTLZ4_10D	{1.02, 1.04, 1.03, 1.05, 1.03, 1.07, 1.02, 1.08, 1.03, 1.04}
DTLZ5_10D	{1.61, 0.68, 0.79, 0.76, 0.81, 0.95, 1.13, 3.35, 3.45, 2.01}
DTLZ6_10D	{5.77, 4.49, 5.73, 6.24, 6.72, 6.67, 6.20, 5.56, 8.04, 4.32}
DTLZ7_10D	{1.00, 0.99, 0.98, 1.00, 0.98, 1.00, 1.00, 0.99, 0.99, 20.98}
WFG1_10D	{2.57, 2.77, 1.10, 1.42, 1.83, 2.05, 2.62, 2.74, 3.83, 13.64}
WFG2_10D	{1.61, 3.00, 4.22, 4.68, 6.55, 7.05, 8.60, 7.92, 9.52, 20.20}
WFG3_10D	{3.00, 4.95, 6.98, 8.99, 11.00, 12.98, 15.06, 17.09, 19.06, 20.34}
WFG4_10D	{1.91, 3.99, 6.02, 8.10, 10.14, 12.16, 14.18, 16.28, 18.25, 20.24}
WFG5_10D	{2.10, 4.10, 6.14, 8.16, 10.16, 12.22, 14.24, 16.28, 18.26, 20.31}
WFG6_10D	{2.21, 4.22, 6.24, 8.26, 10.28, 12.30, 14.35, 16.35, 18.42, 20.39}
WFG7_10D	{1.75, 4.04, 6.06, 8.11, 10.15, 12.18, 14.20, 16.22, 18.25, 20.28}
WFG8_10D	{2.67, 4.91, 7.04, 9.09, 11.10, 12.19, 14.25, 16.25, 18.26, 20.27}
WFG9_10D	{2.15, 4.12, 6.21, 8.08, 10.14, 12.24, 14.20, 16.19, 18.22, 20.23}
IDTLZ1-IDTLZ7_10D	{0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

Tabla C.6: Puntos de referencia utilizados para problemas de prueba en 10 dimensiones.

# D



## RESULTADOS NUMÉRICOS

En este apéndice se presenta un resumen de los resultados obtenidos al calcular los indicadores de desempeño de las pruebas realizadas para evaluar el desempeño de las nuevas funciones de escalarización F4A y F4B en comparación con la función ASF. Los datos se muestran en tablas clasificadas por indicador y número de objetivos de cada problema de optimización multi-objetivo resuelto. A continuación se describe brevemente la información almacenada en cada tabla.

- **Hipervolumen.** En cada tabla se muestra la comparación entre las funciones F4A y F4B contra ASF. Se incluye la media aritmética y la varianza de las 30 ejecuciones realizadas para cada problema. Los valores en negritas indican el mejor valor obtenido; en este caso corresponde al mayor hipervolumen. Adicionalmente, se muestran en celdas grises los problemas en los que la mejora observada se encuentra dentro del intervalo de confianza de 95 % obtenido al realizar el test de suma de rangos de Wilcoxon. Las filas que no tienen celdas grises indican que no pasaron dicho test, por lo que concluimos que, con la muestra de datos obtenidos, el desempeño de ambas funciones es similar.
- **Energía-s.** La información se muestra organizada de la misma forma que las tablas correspondientes al hipervolumen, por lo que su interpretación se realiza de la misma forma. La única diferencia es que el mejor valor corresponde al menor valor obtenido del indicador.
- **Cobertura de conjuntos.** En cada tabla se muestra la cobertura de conjuntos de los frentes de Pareto obtenidos mediante F4A y F4B contra

ASF. Debido a la naturaleza del indicador de desempeño (binario) se incluye el cálculo del indicador en ambos sentidos para cada función. En este caso los valores del indicador se muestran en forma de porcentaje, y la forma de leer cada valor es como sigue. El valor  $(A, B)$  corresponde a la fracción de soluciones obtenidas mediante  $A$  que son dominadas por soluciones obtenidas mediante  $B$ .

## D.1 | HIPERVOLUMEN

Problema	ASF		F4A		Problema	ASF		F4B	
	MED	VAR	MED	VAR		MED	VAR	MED	VAR
DTLZ1	<b>0.94021</b>	2.905E-03	0.27743	4.418E-03	DTLZ1	0.94021	2.905E-03	<b>0.95971</b>	6.988E-04
DTLZ2	0.57260	1.184E-02	<b>0.98881</b>	1.708E-03	DTLZ2	0.57260	1.184E-02	<b>0.74690</b>	5.437E-06
DTLZ3	0.64588	1.387E-02	<b>0.81145</b>	3.729E-02	DTLZ3	<b>0.64588</b>	1.387E-02	0.62899	2.484E-02
DTLZ4	0.96411	3.205E-02	<b>0.96656</b>	3.222E-02	DTLZ4	0.96411	3.205E-02	<b>0.99829</b>	5.436E-10
DTLZ5	0.57260	1.184E-02	<b>0.98881</b>	1.708E-03	DTLZ5	0.57260	1.184E-02	<b>0.74690</b>	5.437E-06
DTLZ6	0.53421	4.167E-02	<b>0.60970</b>	3.468E-02	DTLZ6	<b>0.53421</b>	4.167E-02	0.50781	3.422E-02
DTLZ7	<b>0.95753</b>	9.337E-04	0.21037	5.762E-03	DTLZ7	0.95753	9.337E-04	<b>0.97201</b>	5.284E-04
WFG1	<b>0.57355</b>	4.481E-02	0.16113	9.116E-03	WFG1	<b>0.57355</b>	4.481E-02	0.51501	4.153E-02
WFG2	<b>0.42430</b>	1.256E-02	0.11345	2.825E-03	WFG2	0.42430	1.256E-02	<b>0.43835</b>	3.925E-03
WFG3	<b>0.65929</b>	2.709E-02	0.32225	3.110E-02	WFG3	0.65929	2.709E-02	<b>0.68387</b>	2.577E-02
WFG4	0.31754	2.167E-02	<b>0.52158</b>	6.793E-02	WFG4	0.31754	2.167E-02	<b>0.56102</b>	2.610E-02
WFG5	0.15006	1.374E-02	<b>0.51551</b>	2.846E-02	WFG5	0.15006	1.374E-02	<b>0.49791</b>	2.151E-02
WFG6	<b>0.43905</b>	3.317E-02	0.39697	2.146E-02	WFG6	<b>0.43905</b>	3.317E-02	0.40587	4.639E-02
WFG7	0.27839	2.000E-02	<b>0.63681</b>	4.110E-02	WFG7	0.27839	2.000E-02	<b>0.59735</b>	1.587E-02
WFG8	<b>0.65718</b>	1.391E-02	0.26518	1.808E-02	WFG8	0.65718	1.391E-02	<b>0.67604</b>	2.244E-02
WFG9	<b>0.47164</b>	2.219E-01	0.45408	2.012E-01	WFG9	0.47164	2.219E-01	<b>0.51543</b>	2.290E-01
IDTLZ1	<b>0.93325</b>	6.221E-02	0.93085	6.189E-02	IDTLZ1	0.93325	6.221E-02	<b>0.93326</b>	6.221E-02
IDTLZ2	<b>0.89975</b>	8.995E-02	0.89422	8.884E-02	IDTLZ2	0.89975	8.995E-02	<b>0.89998</b>	9.000E-02
IDTLZ3	<b>0.95714</b>	2.519E-06	0.02772	6.265E-05	IDTLZ3	0.95714	2.519E-06	<b>0.99624</b>	1.741E-06
IDTLZ4	<b>0.93309</b>	6.219E-02	0.92736	6.143E-02	IDTLZ4	0.93309	6.219E-02	<b>0.93333</b>	6.222E-02
IDTLZ5	<b>0.89975</b>	8.995E-02	0.89422	8.884E-02	IDTLZ5	0.89975	8.995E-02	<b>0.89998</b>	9.000E-02
IDTLZ6	<b>0.96640</b>	3.220E-02	0.96046	3.181E-02	IDTLZ6	0.96640	3.220E-02	<b>0.96666</b>	3.222E-02
IDTLZ7	0.96648	3.221E-02	<b>0.96654</b>	3.150E-02	IDTLZ7	0.96648	3.221E-02	<b>0.97006</b>	2.393E-02

Tabla D.1: Comparación de indicador de hipervolumen en problemas con 2 funciones objetivo.

Problema	ASF		F4A	
	MED	VAR	MED	VAR
DTLZ1	<b>0.78115</b>	4.451E-03	0.20439	1.854E-02
DTLZ2	0.36573	7.296E-03	<b>0.51208</b>	2.341E-02
DTLZ3	0.45864	2.128E-02	<b>0.57649</b>	2.555E-02
DTLZ4	<b>0.94960</b>	2.760E-02	0.92533	5.717E-02
DTLZ5	0.10259	5.151E-03	<b>0.82681</b>	1.584E-03
DTLZ6	0.49945	4.155E-02	<b>0.50294</b>	4.076E-02
DTLZ7	<b>0.51209</b>	3.735E-03	0.16349	1.652E-02
WFG1	<b>0.72011</b>	7.633E-02	0.61325	7.563E-02
WFG2	0.66584	1.759E-01	<b>0.71352</b>	1.191E-01
WFG3	<b>0.60044</b>	3.081E-02	0.35630	2.009E-02
WFG4	0.31721	2.426E-02	<b>0.45204</b>	3.449E-02
WFG5	<b>0.48179</b>	3.249E-02	0.46342	2.629E-02
WFG6	<b>0.46938</b>	3.983E-02	0.46405	2.727E-02
WFG7	0.37713	1.257E-02	<b>0.42341</b>	3.178E-02
WFG8	0.50481	7.497E-03	<b>0.64052</b>	4.167E-02
WFG9	0.40139	1.827E-01	<b>0.54778</b>	1.669E-01
IDTLZ1	0.83644	4.997E-02	<b>0.93314</b>	6.220E-02
IDTLZ2	0.92708	6.139E-02	<b>0.93223</b>	6.208E-02
IDTLZ3	0.89394	8.879E-02	<b>0.89884</b>	8.977E-02
IDTLZ4	0.96078	3.183E-02	<b>0.96613</b>	3.219E-02
IDTLZ5	0.93026	6.164E-02	<b>0.93283</b>	6.215E-02
IDTLZ6	0.86290	1.145E-01	<b>0.86650</b>	1.155E-01
IDTLZ7	<b>0.89981</b>	8.987E-02	0.89971	8.994E-02

Problema	ASF		F4B	
	MED	VAR	MED	VAR
DTLZ1	0.78115	4.451E-03	<b>0.79505</b>	4.375E-03
DTLZ2	0.36573	7.296E-03	<b>0.52165</b>	2.479E-02
DTLZ3	0.45864	2.128E-02	<b>0.57568</b>	2.860E-02
DTLZ4	0.94960	2.760E-02	<b>0.98814</b>	9.041E-06
DTLZ5	0.10259	5.151E-03	<b>0.57040</b>	1.541E-02
DTLZ6	0.49945	4.155E-02	<b>0.51185</b>	6.151E-02
DTLZ7	0.51209	3.735E-03	<b>0.79868</b>	6.968E-03
WFG1	<b>0.72011</b>	7.633E-02	0.63706	8.738E-02
WFG2	0.66584	1.759E-01	<b>0.67507</b>	1.778E-01
WFG3	0.60044	3.081E-02	<b>0.65851</b>	4.467E-02
WFG4	0.31721	2.426E-02	<b>0.46655</b>	2.627E-02
WFG5	0.48179	3.249E-02	<b>0.59633</b>	3.369E-02
WFG6	0.46938	3.983E-02	<b>0.47637</b>	3.640E-02
WFG7	0.37713	1.257E-02	<b>0.54767</b>	1.850E-02
WFG8	0.50481	7.497E-03	<b>0.63066</b>	1.164E-02
WFG9	0.40139	1.827E-01	<b>0.58829</b>	1.787E-01
IDTLZ1	0.83644	4.997E-02	<b>0.90195</b>	5.811E-02
IDTLZ2	0.92708	6.139E-02	<b>0.93242</b>	6.210E-02
IDTLZ3	0.89394	8.879E-02	<b>0.89892</b>	8.978E-02
IDTLZ4	0.96078	3.183E-02	<b>0.96635</b>	3.220E-02
IDTLZ5	0.93026	6.164E-02	<b>0.93365</b>	6.111E-02
IDTLZ6	0.86290	1.145E-01	<b>0.86641</b>	1.155E-01
IDTLZ7	<b>0.89981</b>	8.987E-02	0.88463	8.675E-02

Tabla D.2: Comparación de indicador de hipervolumen en problemas con 3 funciones objetivo.

Problema	ASF		F4A	
	MED	VAR	MED	VAR
DTLZ1	<b>0.79463</b>	7.609E-03	0.42196	2.536E-02
DTLZ2	0.40867	2.609E-02	<b>0.71332</b>	1.889E-02
DTLZ3	<b>0.82567</b>	2.259E-03	0.44948	2.827E-02
DTLZ4	0.27359	1.613E-02	<b>0.75469</b>	1.977E-02
DTLZ5	0.49497	8.163E-04	<b>0.89203</b>	2.113E-03
DTLZ6	<b>0.60878</b>	7.215E-03	0.25492	1.378E-02
DTLZ7	<b>0.36283</b>	1.436E-02	0.18684	1.594E-02
WFG1	0.41544	8.635E-02	<b>0.41699</b>	6.774E-02
WFG2	<b>0.85445</b>	9.264E-02	0.69125	1.550E-01
WFG3	<b>0.61315</b>	1.045E-02	0.44094	2.478E-02
WFG4	<b>0.50368</b>	3.429E-02	0.50028	4.138E-02
WFG5	0.50659	3.402E-02	<b>0.59780</b>	2.086E-02
WFG6	0.49412	2.730E-02	<b>0.54457</b>	4.690E-02
WFG7	<b>0.33741</b>	1.888E-02	0.33588	2.287E-02
WFG8	0.37295	2.251E-02	<b>0.66888</b>	5.226E-02
WFG9	0.11596	4.320E-02	<b>0.39254</b>	1.357E-01
IDTLZ1	0.65308	1.500E-02	<b>0.70751</b>	2.482E-02
IDTLZ2	0.88085	2.682E-02	<b>0.94438</b>	3.081E-02
IDTLZ3	0.88272	2.697E-02	<b>0.91043</b>	5.928E-02
IDTLZ4	0.85240	5.198E-02	<b>0.91060</b>	5.928E-02
IDTLZ5	0.92851	2.975E-02	<b>0.98888</b>	2.105E-05
IDTLZ6	0.77558	1.203E-01	<b>0.82205</b>	1.352E-01
IDTLZ7	0.86633	1.155E-01	<b>0.86660</b>	1.155E-01

Problema	ASF		F4B	
	MED	VAR	MED	VAR
DTLZ1	<b>0.79463</b>	7.609E-03	0.79315	7.542E-03
DTLZ2	0.40867	2.609E-02	<b>0.66507</b>	1.640E-02
DTLZ3	<b>0.82567</b>	2.259E-03	0.80508	4.370E-03
DTLZ4	0.27359	1.613E-02	<b>0.52208</b>	1.354E-02
DTLZ5	0.49497	8.163E-04	<b>0.75741</b>	7.284E-04
DTLZ6	<b>0.60878</b>	7.215E-03	0.59730	9.713E-03
DTLZ7	0.36283	1.436E-02	<b>0.75108</b>	1.080E-02
WFG1	<b>0.41544</b>	8.635E-02	0.40384	5.813E-02
WFG2	0.85445	9.264E-02	<b>0.92206</b>	5.064E-02
WFG3	<b>0.61315</b>	1.045E-02	0.59961	1.314E-02
WFG4	0.50368	3.429E-02	<b>0.59369</b>	3.362E-02
WFG5	0.50659	3.402E-02	<b>0.67177</b>	2.476E-02
WFG6	<b>0.49412</b>	2.730E-02	0.48852	4.881E-02
WFG7	0.33741	1.888E-02	<b>0.44829</b>	2.663E-02
WFG8	0.37295	2.251E-02	<b>0.75818</b>	2.857E-02
WFG9	0.11596	4.320E-02	<b>0.16571</b>	5.126E-02
IDTLZ1	0.65308	1.500E-02	<b>0.68065</b>	1.764E-02
IDTLZ2	0.88085	2.682E-02	<b>0.96325</b>	1.178E-04
IDTLZ3	0.88272	2.697E-02	<b>0.89657</b>	5.747E-02
IDTLZ4	0.85240	5.198E-02	<b>0.90002</b>	5.800E-02
IDTLZ5	0.92851	2.975E-02	<b>0.95289</b>	3.129E-02
IDTLZ6	0.77558	1.203E-01	<b>0.90936</b>	5.911E-02
IDTLZ7	<b>0.86633</b>	1.155E-01	0.84093	1.091E-01

Tabla D.3: Comparación de indicador de hipervolumen en problemas con 4 funciones objetivo.

APÉNDICE D. RESULTADOS NUMÉRICOS

Problema	ASF		F4A		Problema	ASF		F4B	
	MED	VAR	MED	VAR		MED	VAR	MED	VAR
DTLZ1	<b>0.86779</b>	2.972E-03	0.46538	3.184E-02	DTLZ1	<b>0.86779</b>	2.972E-03	0.84491	7.041E-03
DTLZ2	0.44663	2.414E-02	<b>0.76285</b>	2.227E-02	DTLZ2	0.44663	2.414E-02	<b>0.76262</b>	1.163E-02
DTLZ3	<b>0.47940</b>	1.391E-02	0.39361	3.101E-02	DTLZ3	0.47940	1.391E-02	<b>0.54441</b>	2.319E-02
DTLZ4	0.90638	1.671E-04	<b>0.98160</b>	1.700E-04	DTLZ4	0.90638	1.671E-04	<b>0.90696</b>	2.846E-02
DTLZ5	<b>0.53888</b>	9.415E-02	0.43925	1.104E-01	DTLZ5	<b>0.53888</b>	9.415E-02	<b>0.78200</b>	5.722E-02
DTLZ6	<b>0.78701</b>	1.122E-02	0.55330	4.481E-02	DTLZ6	0.78701	1.122E-02	<b>0.81507</b>	9.215E-03
DTLZ7	0.15800	5.840E-03	<b>0.40469</b>	1.291E-02	DTLZ7	0.15800	5.840E-03	<b>0.85507</b>	5.447E-03
WFG1	<b>0.45271</b>	3.028E-02	0.28200	2.966E-02	WFG1	0.45271	3.028E-02	<b>0.52692</b>	4.216E-02
WFG2	<b>0.94906</b>	2.659E-02	0.81943	9.651E-02	WFG2	<b>0.94906</b>	2.659E-02	<b>0.95750</b>	2.736E-02
WFG3	<b>0.46443</b>	4.307E-02	0.42610	3.323E-02	WFG3	0.46443	4.307E-02	<b>0.51382</b>	4.173E-02
WFG4	<b>0.73234</b>	1.422E-02	0.42630	2.580E-02	WFG4	<b>0.73234</b>	1.422E-02	0.67221	1.709E-02
WFG5	0.55916	4.649E-02	<b>0.60429</b>	3.735E-02	WFG5	0.55916	4.649E-02	<b>0.63777</b>	2.809E-02
WFG6	0.50864	4.868E-02	<b>0.51723</b>	2.532E-02	WFG6	0.50864	4.868E-02	<b>0.58072</b>	3.023E-02
WFG7	<b>0.39129</b>	2.451E-02	0.21911	2.127E-02	WFG7	<b>0.39129</b>	2.451E-02	<b>0.46373</b>	2.818E-02
WFG8	<b>0.48954</b>	2.249E-02	0.43107	1.518E-02	WFG8	0.48954	2.249E-02	<b>0.52675</b>	2.394E-02
WFG9	0.46108	2.221E-02	<b>0.84287</b>	1.294E-02	WFG9	0.46108	2.221E-02	<b>0.61813</b>	2.034E-02
IDTLZ1	<b>0.07390</b>	3.146E-02	0.03128	1.101E-03	IDTLZ1	<b>0.07390</b>	3.146E-02	0.04525	2.338E-03
IDTLZ2	0.58141	3.776E-02	<b>0.78339</b>	1.534E-01	IDTLZ2	0.58141	3.776E-02	<b>0.69048</b>	1.452E-01
IDTLZ3	0.56714	4.973E-02	<b>0.84305</b>	1.095E-01	IDTLZ3	0.56714	4.973E-02	<b>0.77850</b>	9.330E-02
IDTLZ4	0.57595	3.695E-02	<b>0.94902</b>	3.113E-02	IDTLZ4	0.57595	3.695E-02	<b>0.81559</b>	7.397E-02
IDTLZ5	<b>0.81135</b>	4.693E-02	0.75983	1.757E-01	IDTLZ5	<b>0.81135</b>	4.693E-02	<b>0.89972</b>	5.784E-02
IDTLZ6	0.63062	6.120E-02	<b>0.95735</b>	3.162E-02	IDTLZ6	0.63062	6.120E-02	<b>0.80709</b>	1.002E-01
IDTLZ7	0.89739	8.951E-02	<b>0.89966</b>	8.993E-02	IDTLZ7	<b>0.89739</b>	8.951E-02	0.83150	1.050E-01

Tabla D.4: Comparación de indicador de hipervolumen en problemas con 5 funciones objetivo.

Problema	ASF		F4A		Problema	ASF		F4B	
	MED	VAR	MED	VAR		MED	VAR	MED	VAR
DTLZ1	<b>0.77527</b>	1.018E-02	0.22056	1.290E-02	DTLZ1	<b>0.77527</b>	1.018E-02	0.73828	1.096E-02
DTLZ2	0.34124	2.169E-02	<b>0.75911</b>	1.516E-02	DTLZ2	0.34124	2.169E-02	<b>0.66103</b>	1.192E-02
DTLZ3	<b>0.69014</b>	1.016E-02	0.33642	2.831E-02	DTLZ3	0.69014	1.016E-02	<b>0.70567</b>	6.821E-03
DTLZ4	0.29561	1.181E-02	<b>0.89387</b>	5.109E-02	DTLZ4	0.29561	1.181E-02	<b>0.61672</b>	8.676E-03
DTLZ5	<b>0.38145</b>	4.105E-02	0.27310	4.340E-02	DTLZ5	0.38145	4.105E-02	<b>0.66591</b>	3.344E-02
DTLZ6	<b>0.70644</b>	1.207E-02	0.67136	5.097E-02	DTLZ6	0.70644	1.207E-02	<b>0.77577</b>	1.058E-02
DTLZ7	<b>0.63776</b>	9.404E-03	0.26638	1.208E-02	DTLZ7	<b>0.63776</b>	9.404E-03	0.53999	9.003E-03
WFG1	<b>0.52651</b>	4.487E-02	0.47232	6.290E-02	WFG1	0.52651	4.487E-02	<b>0.53469</b>	4.619E-02
WFG2	<b>0.92164</b>	5.404E-02	0.85487	7.826E-02	WFG2	0.92164	5.404E-02	<b>0.92320</b>	5.297E-02
WFG3	0.48992	6.284E-02	<b>0.49084</b>	4.306E-02	WFG3	<b>0.48992</b>	6.284E-02	0.47490	6.429E-02
WFG4	<b>0.67214</b>	3.051E-02	0.52001	4.453E-02	WFG4	0.67214	3.051E-02	<b>0.69414</b>	4.477E-02
WFG5	0.42035	4.363E-02	<b>0.50445</b>	5.120E-02	WFG5	0.42035	4.363E-02	<b>0.60843</b>	2.731E-02
WFG6	0.47871	6.570E-02	<b>0.51014</b>	6.945E-02	WFG6	0.47871	6.570E-02	<b>0.54992</b>	4.755E-02
WFG7	<b>0.77093</b>	2.168E-02	0.61142	5.807E-02	WFG7	0.77093	2.168E-02	<b>0.77884</b>	2.737E-02
WFG8	0.42047	3.922E-02	<b>0.65890</b>	3.186E-02	WFG8	0.42047	3.922E-02	<b>0.61073</b>	3.562E-02
WFG9	0.44598	2.034E-02	<b>0.70867</b>	2.889E-02	WFG9	0.44598	2.034E-02	<b>0.66831</b>	1.776E-02
IDTLZ1	0.92238	2.958E-02	<b>0.92276</b>	2.959E-02	IDTLZ1	0.92238	2.958E-02	<b>0.92356</b>	2.965E-02
IDTLZ2	0.65039	3.066E-02	<b>0.66423</b>	3.196E-02	IDTLZ2	0.65039	3.066E-02	<b>0.69323</b>	1.981E-02
IDTLZ3	0.78181	4.481E-02	<b>0.79616</b>	4.744E-02	IDTLZ3	0.78181	4.481E-02	<b>0.80746</b>	4.771E-02
IDTLZ4	<b>0.89270</b>	2.814E-02	0.82396	7.558E-02	IDTLZ4	<b>0.89270</b>	2.814E-02	0.89220	2.796E-02
IDTLZ5	0.90936	5.913E-02	<b>0.91405</b>	5.976E-02	IDTLZ5	0.90936	5.913E-02	<b>0.91376</b>	5.972E-02
IDTLZ6	0.91392	2.899E-02	<b>0.92086</b>	2.882E-02	IDTLZ6	0.91392	2.899E-02	<b>0.92140</b>	2.868E-02
IDTLZ7	0.82823	1.368E-01	<b>0.83264</b>	1.387E-01	IDTLZ7	<b>0.82823</b>	1.368E-01	0.79901	1.262E-01

Tabla D.5: Comparación de indicador de hipervolumen en problemas con 6 funciones objetivo.

Problema	ASF		F4A	
	MED	VAR	MED	VAR
DTLZ1	<b>0.81294</b>	1.221E-02	0.58344	1.783E-02
DTLZ2	<b>0.65066</b>	1.990E-02	0.65001	2.341E-02
DTLZ3	<b>0.66770</b>	7.145E-03	0.43772	2.475E-02
DTLZ4	0.65936	5.525E-03	<b>0.84957</b>	1.962E-02
DTLZ5	<b>0.30730</b>	4.751E-02	0.25256	7.504E-02
DTLZ6	0.44166	5.897E-02	<b>0.57816</b>	5.579E-02
DTLZ7	<b>0.56313</b>	1.261E-02	0.29824	1.843E-02
WFG1	<b>0.54699</b>	3.918E-02	0.52031	4.859E-02
WFG2	<b>0.93733</b>	2.865E-02	0.92103	2.999E-02
WFG3	0.91771	2.943E-02	<b>0.95130</b>	5.904E-04
WFG4	0.59836	3.641E-02	<b>0.61382</b>	4.116E-02
WFG5	<b>0.59335</b>	3.429E-02	0.56807	3.829E-02
WFG6	<b>0.59188</b>	3.190E-02	0.51932	3.045E-02
WFG7	<b>0.56534</b>	1.667E-02	0.55873	2.632E-02
WFG8	0.71276	2.050E-02	<b>0.73562</b>	3.265E-02
WFG9	0.46988	3.007E-02	<b>0.62335</b>	3.273E-02
IDTLZ1	<b>0.60202</b>	5.145E-02	0.57452	4.498E-02
IDTLZ2	0.63038	4.277E-02	<b>0.65898</b>	4.269E-02
IDTLZ3	<b>0.70399</b>	3.470E-02	0.69497	3.150E-02
IDTLZ4	0.52129	7.325E-02	<b>0.63378</b>	5.688E-02
IDTLZ5	<b>0.86257</b>	5.367E-02	0.86257	5.367E-02
IDTLZ6	<b>0.78459</b>	4.725E-02	0.77397	4.679E-02
IDTLZ7	<b>0.71568</b>	1.863E-01	0.61684	2.203E-01

Problema	ASF		F4B	
	MED	VAR	MED	VAR
DTLZ1	<b>0.81294</b>	1.221E-02	0.76191	2.656E-02
DTLZ2	0.65066	1.990E-02	<b>0.75830</b>	1.047E-02
DTLZ3	0.66770	7.145E-03	<b>0.70229</b>	1.864E-02
DTLZ4	0.65936	5.525E-03	<b>0.70648</b>	4.212E-02
DTLZ5	0.30730	4.751E-02	<b>0.36810</b>	6.742E-02
DTLZ6	<b>0.44166</b>	5.897E-02	0.43486	4.642E-02
DTLZ7	<b>0.56313</b>	1.261E-02	0.46371	2.737E-02
WFG1	<b>0.54699</b>	3.918E-02	0.50190	3.429E-02
WFG2	0.93733	2.865E-02	<b>0.94825</b>	2.956E-02
WFG3	0.91771	2.943E-02	<b>0.95036</b>	3.130E-04
WFG4	0.59836	3.641E-02	<b>0.67055</b>	2.838E-02
WFG5	<b>0.59335</b>	3.429E-02	0.54048	3.426E-02
WFG6	0.59188	3.190E-02	<b>0.61110</b>	4.798E-02
WFG7	0.56534	1.667E-02	<b>0.58656</b>	1.310E-02
WFG8	0.71276	2.050E-02	<b>0.72063</b>	4.241E-02
WFG9	0.46988	3.007E-02	<b>0.57258</b>	2.498E-02
IDTLZ1	<b>0.60202</b>	5.145E-02	0.54422	5.607E-02
IDTLZ2	<b>0.63038</b>	4.277E-02	0.61508	4.142E-02
IDTLZ3	0.70399	3.470E-02	<b>0.72820</b>	3.483E-02
IDTLZ4	0.52129	7.325E-02	0.52129	7.325E-02
IDTLZ5	<b>0.86257</b>	5.367E-02	0.84874	5.255E-02
IDTLZ6	<b>0.78459</b>	4.725E-02	0.78410	4.779E-02
IDTLZ7	<b>0.71568</b>	1.863E-01	0.71295	1.848E-01

Tabla D.6: Comparación de indicador de hipervolumen en problemas con 10 funciones objetivo.

## D.2 | ENERGÍA-S

Problema	ASF		F4A	
	MED	VAR	MED	VAR
DTLZ1	<b>0.03396</b>	2.855E-03	0.04773	3.969E-03
DTLZ2	0.04250	3.239E-02	<b>0.00436</b>	2.088E-05
DTLZ3	0.10072	4.946E-02	<b>0.09638</b>	3.119E-02
DTLZ4	1.38E-05	3.400E-09	<b>1.19E-05</b>	3.343E-09
DTLZ5	0.04250	3.239E-02	<b>0.00436</b>	2.088E-05
DTLZ6	0.08795	3.459E-02	<b>0.07449</b>	2.418E-02
DTLZ7	<b>0.00444</b>	4.224E-05	0.03115	1.222E-03
WFG1	0.07402	1.355E-02	<b>0.05840</b>	3.148E-03
WFG2	0.16328	3.526E-02	<b>0.11361</b>	2.740E-02
WFG3	<b>0.05228</b>	3.141E-03	0.06130	3.364E-03
WFG4	0.07741	2.333E-02	<b>0.02509</b>	1.699E-03
WFG5	0.13371	1.279E-02	<b>0.10734</b>	1.256E-02
WFG6	0.07397	2.287E-02	<b>0.01898</b>	3.229E-04
WFG7	0.04333	3.722E-03	<b>0.01828</b>	3.455E-04
WFG8	<b>0.00512</b>	2.194E-04	0.03646	3.204E-02
WFG9	0.14668	2.236E-02	<b>0.09273</b>	2.563E-03
IDTLZ1	0.03470	3.215E-02	<b>0.00231</b>	9.716E-05
IDTLZ2	<b>0.00010</b>	2.932E-08	0.05152	4.050E-02
IDTLZ3	<b>0.00004</b>	1.614E-08	0.00005	3.568E-08
IDTLZ4	<b>0.00016</b>	6.857E-09	0.00026	6.342E-09
IDTLZ5	<b>0.00010</b>	2.932E-08	0.05152	4.050E-02
IDTLZ6	<b>0.00001</b>	1.251E-09	0.03335	3.222E-02
IDTLZ7	<b>0.02281</b>	2.308E-04	0.09989	4.796E-02

Problema	ASF		F4B	
	MED	VAR	MED	VAR
DTLZ1	0.03396	2.855E-03	<b>0.02294</b>	1.697E-03
DTLZ2	0.04250	3.239E-02	<b>0.02872</b>	1.721E-02
DTLZ3	0.10072	4.946E-02	<b>0.04910</b>	2.380E-03
DTLZ4	0.00001	3.400E-09	<b>2.45E-06</b>	4.965E-11
DTLZ5	0.04250	3.239E-02	<b>0.02872</b>	1.721E-02
DTLZ6	0.08795	3.459E-02	<b>0.06997</b>	4.973E-03
DTLZ7	<b>0.00444</b>	4.224E-05	0.02013	6.036E-03
WFG1	0.07402	1.355E-02	<b>0.06070</b>	5.986E-03
WFG2	0.16328	3.526E-02	<b>0.13370</b>	2.293E-02
WFG3	<b>0.05228</b>	3.141E-03	0.08831	1.750E-02
WFG4	<b>0.07741</b>	2.333E-02	0.10605	4.254E-02
WFG5	<b>0.13371</b>	1.279E-02	0.18122	2.818E-02
WFG6	0.07397	2.287E-02	<b>0.02855</b>	1.501E-03
WFG7	0.04333	3.722E-03	<b>0.02975</b>	6.190E-04
WFG8	0.00512	2.194E-04	<b>0.00295</b>	4.610E-06
WFG9	<b>0.14668</b>	2.236E-02	0.15609	2.756E-02
IDTLZ1	0.03470	3.215E-02	<b>0.01802</b>	9.152E-03
IDTLZ2	<b>0.00010</b>	2.932E-08	0.01968	3.583E-03
IDTLZ3	<b>0.00004</b>	1.614E-08	0.00020	8.999E-07
IDTLZ4	<b>0.00016</b>	6.857E-09	0.03446	3.217E-02
IDTLZ5	<b>0.00010</b>	2.932E-08	0.01968	3.583E-03
IDTLZ6	<b>0.00001</b>	1.251E-09	0.00042	4.463E-06
IDTLZ7	<b>0.02281</b>	2.308E-04	0.03939	8.619E-03

Tabla D.7: Comparación de indicador de energía-s en problemas con 2 funciones objetivo.

Problema	ASF		F4A		Problema	ASF		F4B	
	MED	VAR	MED	VAR		MED	VAR	MED	VAR
DTLZ1	<b>0.02023</b>	1.187E-02	0.03334	3.222E-02	DTLZ1	0.02023	1.187E-02	<b>1.96E-06</b>	8.707E-12
DTLZ2	<b>0.01456</b>	4.624E-04	0.05342	3.302E-02	DTLZ2	0.01456	4.624E-04	<b>8.00E-03</b>	2.563E-05
DTLZ3	<b>6.50E-07</b>	4.157E-12	0.00174	8.779E-05	DTLZ3	6.50E-07	4.157E-12	<b>1.51E-07</b>	1.931E-14
DTLZ4	0.01702	8.316E-03	<b>0.00198</b>	5.259E-05	DTLZ4	0.01702	8.316E-03	<b>0.00047</b>	5.533E-06
DTLZ5	0.13761	3.320E-02	<b>0.12105</b>	3.035E-02	DTLZ5	<b>0.13761</b>	3.320E-02	0.24996	6.269E-02
DTLZ6	0.30194	7.029E-02	<b>0.10278</b>	2.483E-02	DTLZ6	0.30194	7.029E-02	<b>0.10464</b>	3.529E-02
DTLZ7	0.22586	9.812E-02	<b>0.21113</b>	8.756E-02	DTLZ7	<b>0.22586</b>	9.812E-02	0.26026	9.895E-02
WFG1	0.03947	2.396E-02	<b>0.00124</b>	1.547E-05	WFG1	0.03947	2.396E-02	<b>0.02587</b>	1.230E-02
WFG2	0.05319	4.040E-02	<b>0.00678</b>	2.260E-04	WFG2	0.05319	4.040E-02	<b>0.00968</b>	9.969E-04
WFG3	0.11941	6.147E-02	<b>0.04173</b>	1.769E-02	WFG3	0.11941	6.147E-02	<b>0.05415</b>	3.203E-02
WFG4	0.00771	1.861E-05	<b>0.00630</b>	9.973E-07	WFG4	0.00771	1.861E-05	<b>0.00670</b>	1.659E-06
WFG5	<b>2.06E-06</b>	1.180E-11	0.03333	3.222E-02	WFG5	2.06E-06	1.180E-11	<b>1.66E-06</b>	1.396E-11
WFG6	0.09337	2.859E-02	<b>0.05733</b>	1.920E-04	WFG6	0.09337	2.859E-02	<b>0.05543</b>	6.978E-05
WFG7	<b>0.00103</b>	6.202E-08	0.00137	3.294E-06	WFG7	<b>0.00103</b>	6.202E-08	0.03520	3.212E-02
WFG8	0.06543	4.200E-02	<b>0.04181</b>	3.241E-02	WFG8	0.06543	4.200E-02	<b>0.00137</b>	5.948E-06
WFG9	0.03334	3.222E-02	<b>2.04E-06</b>	3.278E-11	WFG9	0.03334	3.222E-02	<b>1.31E-06</b>	1.305E-11
IDTLZ1	0.23695	9.253E-02	<b>0.05408</b>	2.033E-02	IDTLZ1	0.23695	9.253E-02	<b>0.00233</b>	2.541E-05
IDTLZ2	0.04852	1.865E-02	<b>0.02185</b>	1.377E-02	IDTLZ2	0.04852	1.865E-02	<b>0.00023</b>	1.299E-06
IDTLZ3	0.10140	8.970E-02	<b>0.00520</b>	7.764E-04	IDTLZ3	0.10140	8.970E-02	<b>0.00004</b>	3.520E-09
IDTLZ4	0.07753	4.055E-02	<b>0.07705</b>	4.787E-02	IDTLZ4	<b>0.07753</b>	4.055E-02	0.01100	3.266E-03
IDTLZ5	0.06258	3.891E-02	<b>0.00042</b>	5.039E-06	IDTLZ5	0.06258	3.891E-02	<b>0.02224</b>	2.589E-03
IDTLZ6	0.04757	2.575E-02	<b>0.03524</b>	2.284E-02	IDTLZ6	<b>0.04757</b>	2.575E-02	0.06766	4.561E-02
IDTLZ7	0.04302	3.243E-02	<b>0.00234</b>	7.625E-06	IDTLZ7	0.04302	3.243E-02	<b>0.01335</b>	3.101E-03

Tabla D.8: Comparación de indicador de energía-s en problemas con 3 funciones objetivo.

Problema	ASF		F4A		Problema	ASF		F4B	
	MED	VAR	MED	VAR		MED	VAR	MED	VAR
DTLZ1	<b>0.01848</b>	2.038E-03	0.08178	5.045E-02	DTLZ1	<b>0.01848</b>	2.038E-03	0.02254	6.962E-03
DTLZ2	<b>0.00002</b>	3.890E-09	0.00031	1.049E-06	DTLZ2	<b>0.00002</b>	3.890E-09	0.00007	5.709E-08
DTLZ3	0.00901	2.347E-03	<b>0.00155</b>	6.589E-05	DTLZ3	<b>0.00901</b>	2.347E-03	0.03406	2.102E-02
DTLZ4	<b>0.00289</b>	9.293E-05	0.00679	1.247E-03	DTLZ4	0.00289	9.293E-05	<b>0.00185</b>	3.168E-05
DTLZ5	0.30118	5.379E-02	<b>0.29328</b>	5.494E-02	DTLZ5	0.30118	5.379E-02	<b>0.24575</b>	3.009E-02
DTLZ6	0.26745	1.922E-02	<b>0.23423</b>	2.270E-02	DTLZ6	0.26745	1.922E-02	<b>0.25200</b>	2.134E-02
DTLZ7	<b>0.06111</b>	6.251E-03	0.10539	1.537E-02	DTLZ7	0.06111	6.251E-03	<b>0.06099</b>	5.265E-03
WFG1	0.15004	5.379E-02	<b>0.10807</b>	5.025E-02	WFG1	<b>0.15004</b>	5.379E-02	0.17604	6.652E-02
WFG2	<b>0.00208</b>	7.709E-05	0.07611	5.984E-02	WFG2	0.00208	7.709E-05	<b>0.00072</b>	7.490E-06
WFG3	0.35852	3.771E-02	<b>0.22826</b>	3.025E-02	WFG3	0.35852	3.771E-02	<b>0.35003</b>	4.154E-02
WFG4	1.72E-06	3.009E-11	<b>1.44E-07</b>	1.137E-13	WFG4	<b>1.72E-06</b>	3.009E-11	0.00001	6.365E-10
WFG5	<b>0.00145</b>	6.093E-05	0.03484	3.216E-02	WFG5	0.00145	6.093E-05	<b>1.52E-06</b>	2.049E-11
WFG6	0.03196	5.994E-03	<b>0.03058</b>	5.011E-03	WFG6	<b>0.03196</b>	5.994E-03	0.05567	3.752E-02
WFG7	<b>0.00061</b>	1.857E-06	0.00350	1.345E-04	WFG7	<b>0.00061</b>	1.857E-06	0.03797	1.689E-02
WFG8	0.09748	4.218E-02	<b>0.01868</b>	3.280E-03	WFG8	0.09748	4.218E-02	<b>0.02312</b>	4.662E-03
WFG9	<b>7.11E-06</b>	1.373E-09	0.03355	3.221E-02	WFG9	<b>7.11E-06</b>	1.373E-09	1.25E-05	3.680E-09
IDTLZ1	<b>0.20303</b>	2.623E-02	0.20585	1.337E-02	IDTLZ1	0.20303	2.623E-02	<b>0.19638</b>	1.577E-02
IDTLZ2	0.22802	5.741E-02	<b>0.20800</b>	2.774E-02	IDTLZ2	0.22802	5.741E-02	<b>0.16250</b>	2.458E-02
IDTLZ3	<b>0.20221</b>	4.006E-02	0.21904	2.769E-02	IDTLZ3	<b>0.20221</b>	4.006E-02	0.21936	4.274E-02
IDTLZ4	0.04939	3.926E-03	<b>0.04081</b>	2.339E-03	IDTLZ4	0.04939	3.926E-03	<b>0.04817</b>	2.231E-03
IDTLZ5	0.03609	3.204E-02	<b>0.00245</b>	4.667E-06	IDTLZ5	0.03609	3.204E-02	<b>0.00498</b>	1.907E-04
IDTLZ6	0.06322	4.396E-02	<b>0.05788</b>	3.946E-02	IDTLZ6	0.06322	4.396E-02	<b>0.01442</b>	4.661E-03
IDTLZ7	<b>0.00485</b>	1.490E-04	0.03573	3.207E-02	IDTLZ7	0.00485	1.490E-04	<b>0.00308</b>	5.157E-05

Tabla D.9: Comparación de indicador de energía-s en problemas con 4 funciones objetivo.

APÉNDICE D. RESULTADOS NUMÉRICOS

Problema	ASF		F4A		Problema	ASF		F4B	
	MED	VAR	MED	VAR		MED	VAR	MED	VAR
DTLZ1	<b>0.09211</b>	9.902E-03	0.11242	2.303E-02	DTLZ1	<b>0.09211</b>	9.902E-03	0.09653	1.751E-02
DTLZ2	<b>0.02271</b>	2.335E-03	0.03855	1.941E-02	DTLZ2	<b>0.02271</b>	2.335E-03	0.02979	4.602E-03
DTLZ3	0.04598	3.193E-02	<b>0.01271</b>	5.740E-04	DTLZ3	<b>0.04598</b>	3.193E-02	0.06622	2.275E-02
DTLZ4	<b>0.02071</b>	7.190E-03	0.11250	7.878E-02	DTLZ4	<b>0.02071</b>	7.190E-03	0.05334	1.342E-02
DTLZ5	<b>0.47548</b>	1.238E-02	0.58109	2.690E-02	DTLZ5	<b>0.47548</b>	1.238E-02	0.48196	1.967E-02
DTLZ6	<b>0.35530</b>	4.570E-02	0.46193	3.783E-02	DTLZ6	<b>0.35530</b>	4.570E-02	0.39761	3.615E-02
DTLZ7	<b>0.35466</b>	3.804E-02	0.36159	4.281E-02	DTLZ7	0.35466	3.804E-02	<b>0.22952</b>	2.095E-02
WFG1	0.23667	3.490E-02	<b>0.14168</b>	1.328E-02	WFG1	0.23667	3.490E-02	<b>0.21233</b>	6.776E-02
WFG2	<b>0.02767</b>	4.748E-03	0.18631	4.781E-02	WFG2	0.02767	4.748E-03	<b>0.01192</b>	6.305E-04
WFG3	0.45411	4.221E-02	<b>0.33911</b>	2.377E-02	WFG3	<b>0.45411</b>	4.221E-02	0.46662	3.206E-02
WFG4	0.03362	3.220E-02	<b>0.00001</b>	1.078E-09	WFG4	0.03362	3.220E-02	<b>0.00024</b>	7.166E-07
WFG5	<b>0.00493</b>	3.863E-04	0.01050	3.101E-03	WFG5	<b>0.00493</b>	3.863E-04	0.04316	3.422E-02
WFG6	<b>0.00718</b>	9.048E-04	0.01238	4.096E-03	WFG6	0.00718	9.048E-04	<b>0.00305</b>	8.130E-05
WFG7	<b>0.00969</b>	1.360E-03	0.01125	3.375E-03	WFG7	<b>0.00969</b>	1.360E-03	0.04716	3.449E-02
WFG8	<b>0.00264</b>	1.457E-04	0.00435	2.367E-04	WFG8	<b>0.00264</b>	1.457E-04	0.02136	2.848E-03
WFG9	<b>0.00004</b>	6.103E-09	0.00044	4.150E-06	WFG9	<b>0.00004</b>	6.103E-09	0.03835	3.216E-02
IDTLZ1	0.42105	5.134E-02	<b>0.36516</b>	1.817E-02	IDTLZ1	0.42105	5.134E-02	<b>0.37583</b>	3.491E-02
IDTLZ2	0.42158	3.106E-02	<b>0.39433</b>	2.724E-02	IDTLZ2	<b>0.42158</b>	3.106E-02	0.44151	3.454E-02
IDTLZ3	0.00420	5.483E-05	<b>0.00271</b>	4.031E-06	IDTLZ3	0.00420	5.483E-05	<b>0.00340</b>	4.728E-06
IDTLZ4	<b>0.39638</b>	2.255E-02	0.41305	2.003E-02	IDTLZ4	<b>0.39638</b>	2.255E-02	0.42975	4.702E-02
IDTLZ5	<b>0.00404</b>	1.741E-06	0.08645	6.231E-02	IDTLZ5	0.00404	1.741E-06	<b>0.00388</b>	2.038E-06
IDTLZ6	<b>0.00106</b>	2.424E-07	0.03432	3.216E-02	IDTLZ6	<b>0.00106</b>	2.424E-07	0.00319	1.517E-04
IDTLZ7	0.00634	1.260E-05	<b>0.00541</b>	7.914E-05	IDTLZ7	<b>0.00634</b>	1.260E-05	0.06161	3.606E-02

Tabla D.10: Comparación de indicador de energía-s en problemas con 5 funciones objetivo.

Problema	ASF		F4A		Problema	ASF		F4B	
	MED	VAR	MED	VAR		MED	VAR	MED	VAR
DTLZ1	0.21791	1.745E-02	<b>0.18048</b>	7.306E-03	DTLZ1	0.21791	1.745E-02	<b>0.20438</b>	1.424E-02
DTLZ2	0.23022	4.898E-02	<b>0.20171</b>	5.378E-02	DTLZ2	0.23022	4.898E-02	<b>0.17225</b>	3.427E-02
DTLZ3	<b>0.14514</b>	3.102E-02	0.15298	4.863E-02	DTLZ3	0.14514	3.102E-02	<b>0.10612</b>	3.714E-02
DTLZ4	0.08846	2.946E-02	<b>0.06157</b>	9.497E-03	DTLZ4	0.08846	2.946E-02	<b>0.05743</b>	1.697E-02
DTLZ5	0.32194	2.434E-02	<b>0.25564</b>	3.442E-02	DTLZ5	0.32194	2.434E-02	<b>0.21800</b>	9.175E-03
DTLZ6	<b>0.39549</b>	3.456E-02	0.49549	4.301E-02	DTLZ6	<b>0.39549</b>	3.456E-02	0.41407	2.007E-02
DTLZ7	<b>0.24238</b>	2.378E-02	0.29417	1.783E-02	DTLZ7	<b>0.24238</b>	2.378E-02	0.28490	4.295E-02
WFG1	<b>0.23283</b>	2.568E-02	0.30558	5.544E-02	WFG1	<b>0.23283</b>	2.568E-02	0.24275	2.160E-02
WFG2	<b>0.06166</b>	1.236E-02	0.26726	6.510E-02	WFG2	0.06166	1.236E-02	<b>0.06107</b>	7.896E-03
WFG3	<b>0.40026</b>	1.897E-02	0.46579	4.077E-02	WFG3	<b>0.40026</b>	1.897E-02	0.41310	1.842E-02
WFG4	0.03638	1.560E-02	<b>0.00205</b>	2.513E-05	WFG4	0.03638	1.560E-02	<b>0.02977</b>	8.869E-03
WFG5	<b>0.00806</b>	4.363E-04	0.03415	1.434E-02	WFG5	<b>0.00806</b>	4.363E-04	0.06099	3.598E-02
WFG6	0.05204	3.567E-02	<b>0.00049</b>	1.101E-06	WFG6	0.05204	3.567E-02	<b>0.02432</b>	1.645E-02
WFG7	<b>0.06101</b>	3.822E-02	0.07192	3.774E-02	WFG7	<b>0.06101</b>	3.822E-02	0.06444	3.903E-02
WFG8	0.42120	2.449E-02	<b>0.30553</b>	2.361E-02	WFG8	0.42120	2.449E-02	<b>0.40457</b>	3.835E-02
WFG9	<b>0.05270</b>	1.418E-02	0.08660	5.974E-02	WFG9	<b>0.05270</b>	1.418E-02	0.10652	4.337E-02
IDTLZ1	<b>0.27139</b>	1.549E-02	0.30806	1.982E-02	IDTLZ1	<b>0.27139</b>	1.549E-02	0.31907	3.649E-02
IDTLZ2	0.28669	2.533E-02	<b>0.25054</b>	6.265E-03	IDTLZ2	0.28669	2.533E-02	<b>0.25578</b>	1.346E-02
IDTLZ3	<b>0.00005</b>	3.925E-08	0.00009	1.113E-07	IDTLZ3	<b>0.00005</b>	3.925E-08	0.00006	4.545E-08
IDTLZ4	0.53030	3.554E-02	<b>0.46159</b>	1.998E-02	IDTLZ4	0.53030	3.554E-02	<b>0.51955</b>	5.613E-02
IDTLZ5	<b>0.06674</b>	1.305E-02	0.08063	2.948E-02	IDTLZ5	0.06674	1.305E-02	<b>0.06600</b>	9.308E-03
IDTLZ6	0.41202	3.280E-02	<b>0.41163</b>	2.642E-02	IDTLZ6	<b>0.41202</b>	3.280E-02	0.42941	4.209E-02
IDTLZ7	0.01611	3.026E-03	<b>0.00762</b>	7.567E-04	IDTLZ7	<b>0.01611</b>	3.026E-03	0.18035	5.764E-02

Tabla D.11: Comparación de indicador de energía-s en problemas con 6 funciones objetivo.

Problema	ASF		F4A	
	MED	VAR	MED	VAR
DTLZ1	<b>0.33773</b>	2.974E-02	0.53746	3.199E-02
DTLZ2	0.44721	3.873E-02	<b>0.35538</b>	2.229E-02
DTLZ3	0.39859	3.869E-02	<b>0.34639</b>	2.121E-02
DTLZ4	<b>0.16349</b>	2.289E-02	0.19282	1.608E-02
DTLZ5	<b>0.18951</b>	2.728E-02	0.22913	6.571E-03
DTLZ6	<b>0.23592</b>	1.061E-02	0.25320	1.025E-02
DTLZ7	<b>0.11122</b>	1.139E-02	0.18736	3.806E-02
WFG1	<b>0.05130</b>	4.034E-04	0.05535	6.204E-04
WFG2	<b>0.26651</b>	2.512E-02	0.39618	4.663E-02
WFG3	<b>0.18477</b>	3.717E-02	0.20172	1.797E-02
WFG4	0.10415	3.468E-02	<b>0.08143</b>	7.362E-03
WFG5	0.10478	2.563E-02	<b>0.08112</b>	3.148E-02
WFG6	0.21392	7.896E-02	<b>0.08966</b>	2.990E-02
WFG7	<b>0.14863</b>	3.085E-02	0.14944	4.005E-02
WFG8	0.31137	3.879E-02	<b>0.23653</b>	3.178E-02
WFG9	0.21529	5.355E-02	<b>0.12855</b>	4.029E-02
IDTLZ1	2.23E-07	1.447E-12	<b>4.29E-10</b>	5.282E-18
IDTLZ2	0.15901	2.318E-02	<b>0.11532</b>	1.747E-02
IDTLZ3	<b>2.53E-23</b>	1.855E-44	1.30E-16	4.873E-31
IDTLZ4	0.15096	6.290E-02	<b>0.13664</b>	4.395E-02
IDTLZ5	0.03375	3.219E-02	<b>0.01551</b>	3.185E-03
IDTLZ6	<b>0.00493</b>	6.738E-04	0.03346	3.221E-02
IDTLZ7	<b>0.00088</b>	1.771E-05	0.05170	4.037E-02

Problema	ASF		F4B	
	MED	VAR	MED	VAR
DTLZ1	<b>0.33773</b>	2.974E-02	0.36237	2.449E-02
DTLZ2	0.44721	3.873E-02	<b>0.37708</b>	3.357E-02
DTLZ3	<b>0.39859</b>	3.869E-02	0.45988	5.006E-02
DTLZ4	<b>0.16349</b>	2.289E-02	0.24254	6.227E-02
DTLZ5	<b>0.18951</b>	2.728E-02	0.21848	1.019E-02
DTLZ6	<b>0.23592</b>	1.061E-02	0.57038	5.652E-02
DTLZ7	<b>0.11122</b>	1.139E-02	0.13158	1.637E-02
WFG1	<b>0.05130</b>	4.034E-04	0.05754	1.369E-03
WFG2	<b>0.26651</b>	2.512E-02	0.28387	2.348E-02
WFG3	0.18477	3.717E-02	<b>0.12599</b>	3.219E-03
WFG4	0.10415	3.468E-02	<b>0.08544</b>	9.626E-03
WFG5	<b>0.10478</b>	2.563E-02	0.11690	4.710E-02
WFG6	0.21392	7.896E-02	<b>0.15141</b>	5.258E-02
WFG7	0.14863	3.085E-02	<b>0.12281</b>	4.305E-02
WFG8	0.31137	3.879E-02	<b>0.21146</b>	4.911E-02
WFG9	0.21529	5.355E-02	<b>0.14267</b>	2.023E-02
IDTLZ1	<b>2.23E-07</b>	1.447E-12	3.33E-07	1.675E-12
IDTLZ2	<b>0.15901</b>	2.318E-02	0.18295	4.306E-02
IDTLZ3	<b>2.53E-23</b>	1.855E-44	2.86E-17	2.366E-32
IDTLZ4	0.15096	6.290E-02	<b>0.14840</b>	5.898E-02
IDTLZ5	0.03375	3.219E-02	<b>0.01226</b>	3.946E-03
IDTLZ6	0.00493	6.738E-04	<b>0.00011</b>	4.174E-08
IDTLZ7	<b>0.00088</b>	1.771E-05	0.00364	1.844E-04

Tabla D.12: Comparación de indicador de energía-s en problemas con 10 funciones objetivo.

## D.3 | COBERTURA DE CONJUNTOS

Problema	F4A				F4B			
	C(ASF,F4A)		C(F4A,ASF)		C(ASF,F4B)		C(F4B,ASF)	
	MED	VAR	MED	VAR	MED	VAR	MED	VAR
DTLZ1	0.742 %	2.320E-04	0.710 %	2.335E-04	0.677 %	1.638E-04	1.871 %	1.024E-03
DTLZ2	0.032 %	3.122E-06	0.194 %	4.142E-05	0.129 %	1.124E-05	0.355 %	6.805E-05
DTLZ3	4.903 %	3.906E-03	1.097 %	4.216E-04	10.677 %	1.016E-02	2.742 %	3.393E-03
DTLZ4	0.032 %	3.122E-06	1.710 %	8.434E-03	0.355 %	3.580E-05	0.226 %	2.393E-05
DTLZ5	0.032 %	3.122E-06	0.194 %	4.142E-05	0.129 %	1.124E-05	0.355 %	6.805E-05
DTLZ6	35.613 %	2.097E-01	53.935 %	2.115E-01	48.226 %	2.481E-01	47.290 %	2.416E-01
DTLZ7	2.000 %	2.452E-04	0.677 %	7.347E-05	0.839 %	7.159E-05	0.742 %	5.786E-05
WFG1	34.161 %	1.808E-01	23.419 %	7.756E-02	38.484 %	1.536E-01	26.710 %	1.361E-01
WFG2	15.774 %	3.979E-02	14.226 %	3.780E-02	14.000 %	5.019E-02	22.065 %	5.918E-02
WFG3	1.194 %	9.188E-04	6.806 %	5.751E-03	3.548 %	4.870E-03	6.000 %	1.603E-02
WFG4	3.452 %	2.044E-03	1.419 %	4.050E-04	2.613 %	1.617E-03	3.290 %	1.195E-03
WFG5	0.129 %	1.124E-05	0 %	0.000E+00	0.097 %	1.519E-05	0.032 %	3.122E-06
WFG6	25.452 %	6.907E-02	16.968 %	7.452E-02	28.290 %	7.500E-02	13.968 %	4.437E-02
WFG7	2.129 %	5.145E-04	0.677 %	1.767E-04	1.194 %	4.092E-04	1.806 %	7.446E-04
WFG8	39.710 %	1.806E-02	11.548 %	1.244E-03	22.903 %	2.631E-02	22.774 %	2.557E-02
WFG9	19.774 %	1.102E-01	15.065 %	9.998E-02	13.323 %	1.089E-01	24.645 %	1.453E-01
IDTLZ1	3.226 %	3.122E-02	0.774 %	1.650E-03	0.774 %	1.650E-03	0 %	0.000E+00
IDTLZ2	0 %	0.000E+00	6.452 %	6.035E-02	6.484 %	6.032E-02	0.032 %	3.122E-06
IDTLZ3	0 %	0.000E+00	0.032 %	3.122E-06	0.032 %	3.122E-06	0 %	0.000E+00
IDTLZ4	0 %	0.000E+00	0 %	0.000E+00	0.161 %	1.998E-05	0.032 %	3.122E-06
IDTLZ5	0 %	0.000E+00	6.452 %	6.035E-02	6.484 %	6.032E-02	0.032 %	3.122E-06
IDTLZ6	0 %	0.000E+00	3.258 %	3.120E-02	3.226 %	3.122E-02	0.032 %	3.122E-06
IDTLZ7	1.452 %	2.506E-04	1.258 %	2.450E-04	2.677 %	1.551E-03	1.258 %	1.611E-04
MED	8.252 %	2.90E-02	7.255 %	2.92E-02	8.927 %	3.63E-02	7.663 %	2.93E-02

Tabla D.13: Comparación de indicador de cobertura de conjuntos en problemas con 2 funciones objetivo.

D.3. COBERTURA DE CONJUNTOS

Problema	F4A				F4B			
	C(ASF,F4A)		C(F4A,ASF)		C(ASF,F4B)		C(F4B,ASF)	
	MED	VAR	MED	VAR	MED	VAR	MED	VAR
DTLZ1	0 %	0.000E+00	0 %	0.000E+00	0.301 %	1.400E-04	0.043 %	5.550E-06
DTLZ2	0.022 %	1.387E-06	0 %	0.000E+00	0.280 %	2.516E-05	0.086 %	4.995E-06
DTLZ3	0.925 %	1.654E-04	0.151 %	1.637E-05	1.871 %	4.772E-04	0.710 %	9.444E-05
DTLZ4	0.043 %	2.682E-06	0.065 %	3.885E-06	0.172 %	1.138E-05	0.022 %	1.387E-06
DTLZ5	0.280 %	3.376E-05	0.194 %	1.202E-05	0.516 %	3.931E-05	0.151 %	1.350E-05
DTLZ6	42.086 %	1.906E-01	28.645 %	1.159E-01	44.323 %	2.006E-01	37.011 %	1.760E-01
DTLZ7	0.452 %	3.839E-05	0.516 %	3.931E-05	0.516 %	3.071E-05	0.516 %	3.644E-05
WFG1	1.269 %	5.171E-04	4.946 %	3.672E-03	1.226 %	2.755E-04	2.882 %	1.836E-03
WFG2	0.301 %	4.255E-05	0.151 %	1.064E-05	1.290 %	3.898E-04	1.075 %	3.518E-04
WFG3	4.108 %	1.904E-03	2.108 %	9.494E-04	6.925 %	6.628E-03	2.559 %	1.709E-03
WFG4	1.656 %	1.344E-04	0.280 %	1.369E-05	1.677 %	1.745E-04	1.054 %	7.677E-05
WFG5	0.344 %	4.264E-05	0.022 %	1.387E-06	0.710 %	4.569E-05	0.151 %	7.770E-06
WFG6	2.108 %	9.637E-04	0.581 %	9.962E-05	4.043 %	5.287E-03	1.527 %	8.866E-04
WFG7	0.688 %	4.440E-05	0.086 %	7.862E-06	0.903 %	7.612E-05	0.559 %	2.608E-05
WFG8	2.774 %	1.551E-04	0.645 %	4.440E-05	4.710 %	4.672E-04	2.129 %	2.048E-04
WFG9	2.774 %	1.084E-02	13.290 %	7.176E-02	6.495 %	4.455E-02	21.419 %	1.242E-01
IDTLZ1	3.226 %	3.122E-02	0 %	0.000E+00	5.763 %	4.890E-02	0 %	0.000E+00
IDTLZ2	0 %	0.000E+00						
IDTLZ3	3.892 %	2.285E-02	9.613 %	8.625E-02	2.516 %	1.866E-02	5.742 %	4.841E-02
IDTLZ4	0 %	0.000E+00	0 %	0.000E+00	0.022 %	1.387E-06	0 %	0.000E+00
IDTLZ5	0 %	0.000E+00	0 %	0.000E+00	0.323 %	3.122E-04	0 %	0.000E+00
IDTLZ6	10.344 %	7.743E-02	11.226 %	8.586E-02	12.538 %	1.063E-01	10.817 %	7.971E-02
IDTLZ7	1.806 %	1.783E-04	6.624 %	2.987E-02	3.871 %	1.406E-02	8.258 %	5.037E-02
MED	3.439 %	1.47E-02	3.441 %	1.72E-02	4.391 %	1.95E-02	4.205 %	2.10E-02

Tabla D.14: Comparación de indicador de cobertura de conjuntos en problemas con 3 funciones objetivo.

Problema	F4A				F4B			
	C(ASF,F4A)		C(F4A,ASF)		C(ASF,F4B)		C(F4B,ASF)	
	MED	VAR	MED	VAR	MED	VAR	MED	VAR
DTLZ1	0.022 %	1.387E-06	0 %	0.000E+00	0.022 %	1.387E-06	0 %	0.000E+00
DTLZ2	0.151 %	1.064E-05	0.022 %	1.387E-06	0.129 %	6.937E-06	0.151 %	1.064E-05
DTLZ3	0.710 %	1.403E-04	0.043 %	2.682E-06	0.495 %	6.872E-05	0.258 %	1.915E-05
DTLZ4	0.086 %	4.995E-06	0.108 %	6.012E-06	0.237 %	1.591E-05	0.129 %	9.805E-06
DTLZ5	0 %	0.000E+00	0.022 %	1.387E-06	0.086 %	7.862E-06	0.086 %	7.862E-06
DTLZ6	12.151 %	2.424E-02	12.796 %	1.470E-02	15.226 %	3.133E-02	19.075 %	3.488E-02
DTLZ7	0.258 %	1.341E-05	0.215 %	1.258E-05	0.172 %	1.424E-05	0.172 %	1.424E-05
WFG1	0.602 %	2.189E-04	2.215 %	2.737E-03	1.269 %	6.949E-04	3.892 %	3.404E-03
WFG2	0.280 %	2.229E-05	0.280 %	2.516E-05	0.516 %	3.644E-05	0.516 %	3.358E-05
WFG3	1.935 %	3.394E-04	1.161 %	3.899E-04	2.043 %	1.021E-03	3.097 %	1.782E-03
WFG4	1.785 %	1.703E-04	0.108 %	6.012E-06	1.419 %	1.226E-04	0.602 %	4.116E-05
WFG5	0.323 %	1.683E-05	0 %	0.000E+00	0.538 %	4.421E-05	0.258 %	1.341E-05
WFG6	1.742 %	3.604E-04	0.151 %	1.637E-05	2.860 %	9.439E-04	0.753 %	2.028E-04
WFG7	1.183 %	8.232E-05	0.086 %	4.995E-06	0.946 %	4.810E-05	0.452 %	3.839E-05
WFG8	3.226 %	3.243E-04	0.409 %	5.356E-05	2.860 %	3.360E-04	1.355 %	2.107E-04
WFG9	0.366 %	5.975E-05	5.398 %	8.153E-03	3.161 %	1.209E-02	3.441 %	9.110E-03
IDTLZ1	3.118 %	2.754E-02	0 %	0.000E+00	3.398 %	3.112E-02	0 %	0.000E+00
IDTLZ2	0.129 %	9.805E-06	0.215 %	1.258E-05	0.043 %	2.682E-06	0.323 %	2.257E-05
IDTLZ3	6.538 %	5.855E-02	0.301 %	2.248E-05	3.355 %	3.072E-02	0.323 %	2.830E-05
IDTLZ4	0.280 %	3.089E-05	0.065 %	6.752E-06	0.151 %	1.350E-05	0.129 %	9.805E-06
IDTLZ5	0.065 %	3.885E-06	0.129 %	6.937E-06	0.581 %	6.702E-04	1.871 %	7.319E-03
IDTLZ6	12.000 %	8.379E-02	11.720 %	7.915E-02	2.925 %	2.238E-02	4.968 %	3.595E-02
IDTLZ7	2.258 %	4.292E-04	6.774 %	3.007E-02	4.258 %	3.106E-02	1.183 %	1.339E-04
MED	2.139 %	8.54E-03	1.835 %	5.89E-03	2.030 %	7.08E-03	1.871 %	4.05E-03

Tabla D.15: Comparación de indicador de cobertura de conjuntos en problemas con 4 funciones objetivo.

APÉNDICE D. RESULTADOS NUMÉRICOS

Problema	F4A				F4B			
	C(ASF,F4A)		C(F4A,ASF)		C(ASF,F4B)		C(F4B,ASF)	
	MED	VAR	MED	VAR	MED	VAR	MED	VAR
DTLZ1	0.013 %	4.995E-07	0 %	0.000E+00	0.013 %	4.995E-07	0.013 %	4.995E-07
DTLZ2	0.245 %	8.957E-06	0 %	0.000E+00	0.181 %	4.995E-06	0.090 %	3.829E-06
DTLZ3	0.594 %	4.219E-05	0 %	0.000E+00	0.632 %	4.932E-05	0.090 %	5.894E-06
DTLZ4	0.090 %	2.797E-06	0.026 %	9.657E-07	0.181 %	1.016E-05	0.013 %	4.995E-07
DTLZ5	0.426 %	5.671E-05	0.077 %	4.562E-06	0.684 %	7.143E-05	0.374 %	5.671E-05
DTLZ6	2.103 %	5.966E-04	11.045 %	3.147E-03	33.419 %	3.652E-02	6.477 %	3.563E-03
DTLZ7	0.232 %	1.215E-05	0.181 %	6.027E-06	0.181 %	8.092E-06	0.142 %	1.089E-05
WFG1	1.084 %	5.246E-04	1.794 %	1.056E-03	1.161 %	5.929E-04	3.110 %	2.087E-03
WFG2	0.103 %	1.029E-05	0.090 %	2.797E-06	0.284 %	1.775E-05	0.181 %	1.119E-05
WFG3	3.110 %	7.925E-04	1.806 %	4.252E-04	3.265 %	2.314E-03	3.587 %	2.278E-03
WFG4	1.574 %	9.077E-05	0.013 %	4.995E-07	1.071 %	5.821E-05	0.245 %	1.309E-05
WFG5	0.271 %	1.175E-05	0.013 %	4.995E-07	0.400 %	2.994E-05	0.065 %	3.197E-06
WFG6	1.123 %	7.269E-05	0.052 %	2.830E-06	1.084 %	2.634E-04	0.916 %	2.314E-04
WFG7	0.994 %	5.251E-05	0.052 %	1.798E-06	0.671 %	2.827E-05	0.348 %	1.935E-05
WFG8	2.787 %	2.162E-04	0.568 %	3.590E-05	1.729 %	2.502E-04	0.903 %	3.300E-05
WFG9	0.103 %	3.063E-06	0.103 %	7.193E-06	0.168 %	6.993E-06	0.142 %	1.089E-05
IDTLZ1	2.323 %	1.618E-02	3.226 %	3.122E-02	2.710 %	2.160E-02	3.226 %	3.122E-02
IDTLZ2	0.103 %	1.855E-05	0.039 %	2.431E-06	0.013 %	4.995E-07	0.065 %	4.229E-06
IDTLZ3	6.503 %	6.029E-02	4.929 %	3.790E-02	5.961 %	5.125E-02	3.884 %	3.171E-02
IDTLZ4	0.013 %	4.995E-07	0.103 %	1.132E-05	0.013 %	4.995E-07	0.052 %	4.895E-06
IDTLZ5	0.013 %	4.995E-07	0 %	0.000E+00	0.026 %	9.657E-07	0.026 %	1.998E-06
IDTLZ6	0 %	0.000E+00						
IDTLZ7	5.316 %	3.009E-02	0.671 %	6.543E-05	5.884 %	5.070E-02	1.123 %	9.850E-05
MED	1.266 %	4.74E-03	1.078 %	3.21E-03	2.597 %	7.12E-03	1.090 %	3.10E-03

Tabla D.16: Comparación de indicador de cobertura de conjuntos en problemas con 5 funciones objetivo.

Problema	F4A				F4B			
	C(ASF,F4A)		C(F4A,ASF)		C(ASF,F4B)		C(F4B,ASF)	
	MED	VAR	MED	VAR	MED	VAR	MED	VAR
DTLZ1	0 %	0.000E+00						
DTLZ2	0.086 %	2.844E-06	0.022 %	6.706E-07	0.140 %	3.422E-06	0.022 %	6.706E-07
DTLZ3	0.398 %	2.969E-05	0.032 %	1.688E-06	0.237 %	1.878E-05	0.151 %	5.619E-06
DTLZ4	0.151 %	1.422E-05	0.011 %	3.469E-07	0.097 %	2.289E-06	0.065 %	2.451E-06
DTLZ5	1.247 %	2.100E-04	0.473 %	3.425E-05	8.731 %	8.429E-03	0.548 %	5.201E-05
DTLZ6	1.935 %	4.792E-04	7.280 %	1.968E-03	26.280 %	1.520E-02	6.280 %	2.314E-03
DTLZ7	0.194 %	6.290E-06	0.226 %	9.596E-06	0.280 %	1.154E-05	0.129 %	4.070E-06
WFG1	0.538 %	1.740E-04	1.473 %	9.081E-04	0.409 %	6.646E-05	1.828 %	2.509E-03
WFG2	0.043 %	1.249E-06	0.075 %	2.659E-06	0.108 %	7.446E-06	0.043 %	1.249E-06
WFG3	1.527 %	5.790E-04	1.699 %	2.304E-04	1.602 %	2.447E-04	2.505 %	5.748E-04
WFG4	1.032 %	3.466E-05	0.022 %	6.706E-07	0.484 %	3.859E-05	0.097 %	2.289E-06
WFG5	0.290 %	1.272E-05	0.043 %	1.966E-06	0.247 %	1.001E-05	0.043 %	1.249E-06
WFG6	0.882 %	7.208E-05	0.043 %	1.249E-06	0.817 %	1.275E-04	0.484 %	3.573E-05
WFG7	1.333 %	5.376E-05	0.022 %	6.706E-07	0.710 %	4.426E-05	0.269 %	8.186E-06
WFG8	1.495 %	1.117E-04	0.796 %	6.213E-05	1.301 %	1.142E-04	1.215 %	8.426E-05
WFG9	0.151 %	6.336E-06	0.086 %	7.145E-06	0.140 %	4.139E-06	0.129 %	6.220E-06
IDTLZ1	0 %	0.000E+00	0.011 %	3.469E-07	0 %	0.000E+00	0.043 %	5.550E-06
IDTLZ2	0.097 %	1.304E-05	0.022 %	6.706E-07	0.054 %	4.370E-06	0.065 %	6.035E-06
IDTLZ3	5.871 %	4.699E-02	5.860 %	4.699E-02	6.731 %	6.002E-02	4.925 %	3.387E-02
IDTLZ4	0.172 %	2.285E-05	0.140 %	1.489E-05	0.194 %	6.292E-05	0.118 %	2.190E-05
IDTLZ5	0.366 %	2.124E-04	0.968 %	1.361E-03	0.215 %	1.387E-04	1.097 %	1.989E-03
IDTLZ6	0 %	0.000E+00	0.011 %	3.469E-07	0 %	0.000E+00	0 %	0.000E+00
IDTLZ7	4.839 %	3.044E-02	3.796 %	3.094E-02	7.667 %	6.322E-02	0.323 %	5.698E-05
MED	0.985 %	3.45E-03	1.005 %	3.59E-03	2.454 %	6.42E-03	0.886 %	1.81E-03

Tabla D.17: Comparación de indicador de cobertura de conjuntos en problemas con 6 funciones objetivo.

Problema	F4A				F4B			
	C(ASF,F4A)		C(F4A,ASF)		C(ASF,F4B)		C(F4B,ASF)	
	MED	VAR	MED	VAR	MED	VAR	MED	VAR
<b>DTLZ1</b>	0 %	0.000E+00						
<b>DTLZ2</b>	0 %	0.000E+00						
<b>DTLZ3</b>	0.411 %	4.979E-05	0 %	0.000E+00	0.073 %	4.128E-06	0 %	0.000E+00
<b>DTLZ4</b>	0 %	0.000E+00						
<b>DTLZ5</b>	0.469 %	6.196E-05	0.235 %	2.915E-05	2.214 %	2.090E-03	0.528 %	4.678E-05
<b>DTLZ6</b>	0.543 %	4.855E-05	3.314 %	8.360E-04	16.232 %	4.943E-03	3.372 %	7.475E-04
<b>DTLZ7</b>	0.103 %	4.945E-06	0.088 %	3.225E-06	0 %	0.000E+00	0.015 %	6.450E-07
<b>WFG1</b>	1.642 %	1.831E-03	1.496 %	2.812E-03	0.982 %	3.960E-04	1.144 %	1.100E-03
<b>WFG2</b>	0 %	0.000E+00	0.015 %	6.450E-07	0.044 %	1.806E-06	0.015 %	6.450E-07
<b>WFG3</b>	0.733 %	1.475E-04	0.484 %	8.123E-05	0.880 %	1.265E-04	0.674 %	1.211E-04
<b>WFG4</b>	0.029 %	1.247E-06	0 %	0.000E+00	0.015 %	6.450E-07	0.029 %	2.580E-06
<b>WFG5</b>	0.059 %	3.655E-06	0.015 %	6.450E-07	0.073 %	6.794E-06	0 %	0.000E+00
<b>WFG6</b>	2.478 %	1.683E-03	0.059 %	6.321E-06	1.290 %	7.826E-04	0.777 %	2.055E-04
<b>WFG7</b>	0.059 %	2.322E-06	0.015 %	6.450E-07	0.088 %	5.891E-06	0.059 %	2.322E-06
<b>WFG8</b>	1.085 %	1.462E-04	0.235 %	3.449E-05	1.085 %	1.382E-04	0.337 %	3.728E-05
<b>WFG9</b>	0.015 %	6.450E-07	0 %	0.000E+00	0.088 %	4.558E-06	0.029 %	1.247E-06
<b>IDTLZ1</b>	0 %	0.000E+00	0.015 %	6.450E-07	0 %	0.000E+00	0.015 %	6.450E-07
<b>IDTLZ2</b>	0 %	0.000E+00						
<b>IDTLZ3</b>	3.226 %	3.122E-02	0 %	0.000E+00	3.240 %	3.121E-02	0 %	0.000E+00
<b>IDTLZ4</b>	0 %	0.000E+00						
<b>IDTLZ5</b>	2.361 %	8.177E-03	2.771 %	1.668E-02	2.801 %	1.477E-02	4.194 %	3.185E-02
<b>IDTLZ6</b>	0 %	0.000E+00						
<b>IDTLZ7</b>	0.044 %	5.805E-06	0.073 %	8.127E-06	13.519 %	1.020E-01	0.103 %	1.028E-05
MED	0.576 %	1.89E-03	0.383 %	8.91E-04	1.853 %	6.80E-03	0.491 %	1.48E-03

Tabla D.18: Comparación de indicador de cobertura de conjuntos en problemas con 10 funciones objetivo.



# E



## PRUEBA DE SUMA DE RANGOS DE WILCOXON

En este apéndice se muestran los resultados obtenidos al aplicar la prueba de suma de rangos de Wilcoxon a los resultados numéricos obtenidos al calcular los indicadores de hipervolumen y energía-s en los problemas de prueba utilizados. Los datos mostrados en cada tabla corresponden a los valores-p calculados mediante la función `wilcox.test()` del lenguaje de programación R. Un valor mayor a 0.05 indica que no existe suficiente evidencia de que una función de escalarización se comporte mejor que otra en el problema en cuestión.



## E.1 | HIPERVOLUMEN

Problema	Función	ASF	F4A	F4B
DTLZ1_2D	ASF	-	1.5080e-11	8.9956e-01
	F4A	1.0000e+00	-	1.0000e+00
	F4B	1.0307e-01	1.5071e-11	-
DTLZ2_2D	ASF	-	1.0000e+00	1.0000e+00
	F4A	1.4790e-11	-	1.4386e-11
	F4B	1.4215e-11	1.0000e+00	-
DTLZ3_2D	ASF	-	1.0000e+00	4.5585e-01
	F4A	2.0556e-07	-	3.7982e-07
	F4B	5.5000e-01	1.0000e+00	-
DTLZ4_2D	ASF	-	1.0000e+00	1.0000e+00
	F4A	2.5049e-10	-	2.6071e-10
	F4B	1.3515e-11	1.0000e+00	-
DTLZ5_2D	ASF	-	1.0000e+00	1.0000e+00
	F4A	1.4790e-11	-	1.4386e-11
	F4B	1.4215e-11	1.0000e+00	-
DTLZ6_2D	ASF	-	8.9149e-01	2.6007e-01
	F4A	1.1129e-01	-	2.9714e-02
	F4B	7.4470e-01	9.7127e-01	-
DTLZ7_2D	ASF	-	1.5099e-11	9.9971e-01
	F4A	1.0000e+00	-	1.0000e+00
	F4B	3.0170e-04	1.5090e-11	-
WFG1_2D	ASF	-	3.3478e-11	1.5209e-01
	F4A	1.0000e+00	-	1.0000e+00
	F4B	8.5136e-01	6.6443e-11	-
WFG2_2D	ASF	-	1.5099e-11	9.3327e-01
	F4A	1.0000e+00	-	1.0000e+00
	F4B	6.8661e-02	1.5099e-11	-
WFG3_2D	ASF	-	1.6760e-08	6.4141e-01
	F4A	1.0000e+00	-	1.0000e+00
	F4B	3.6413e-01	3.8844e-09	-
WFG4_2D	ASF	-	2.1792e-02	1.0000e+00
	F4A	9.7897e-01	-	1.0000e+00
	F4B	4.0973e-07	6.9193e-07	-
WFG5_2D	ASF	-	1.0000e+00	1.0000e+00
	F4A	2.5348e-10	-	2.1019e-01
	F4B	8.0661e-11	7.9405e-01	-
WFG6_2D	ASF	-	1.5209e-01	1.8554e-01
	F4A	8.5136e-01	-	5.7331e-01
	F4B	8.1839e-01	4.3250e-01	-
WFG7_2D	ASF	-	1.3442e-03	1.0000e+00
	F4A	9.9872e-01	-	1.0000e+00
	F4B	9.2500e-09	1.5794e-10	-
WFG8_2D	ASF	-	3.3478e-11	7.2534e-01
	F4A	1.0000e+00	-	1.0000e+00
	F4B	2.7962e-01	6.0283e-11	-
WFG9_2D	ASF	-	7.4470e-01	9.9308e-01
	F4A	2.6007e-01	-	6.2471e-01
	F4B	7.2061e-03	3.8091e-01	-
IDTLZ1_2D	ASF	-	3.2917e-09	9.9827e-01
	F4A	1.0000e+00	-	1.0000e+00
	F4B	1.8140e-03	3.3589e-09	-
IDTLZ2_2D	ASF	-	4.3414e-08	1.0000e+00
	F4A	1.0000e+00	-	1.0000e+00
	F4B	3.8300e-08	4.5186e-08	-
IDTLZ3_2D	ASF	-	1.5052e-11	1.0000e+00
	F4A	1.0000e+00	-	1.0000e+00
	F4B	1.5014e-11	1.5043e-11	-
IDTLZ4_2D	ASF	-	3.4884e-09	1.0000e+00
	F4A	1.0000e+00	-	1.0000e+00
	F4B	2.9010e-09	2.9123e-09	-
IDTLZ5_2D	ASF	-	4.3414e-08	1.0000e+00
	F4A	1.0000e+00	-	1.0000e+00
	F4B	3.8300e-08	4.5186e-08	-
IDTLZ6_2D	ASF	-	2.7711e-10	1.0000e+00
	F4A	1.0000e+00	-	1.0000e+00
	F4B	2.7741e-10	2.7711e-10	-
IDTLZ7_2D	ASF	-	1.0134e-07	1.7390e-01
	F4A	1.0000e+00	-	1.0000e+00
	F4B	8.2987e-01	3.4842e-06	-

Tabla E.1: Prueba de suma de rangos de Wilcoxon aplicada al indicador de hipervolumen en problemas de prueba con 2 funciones objetivo.

APÉNDICE E. PRUEBA DE SUMA DE RANGOS DE WILCOXON

Problema	Función	ASF	F4A	F4B
DTLZ1_3D	ASF	-	1.5099e-11	7.5874e-01
	F4A	1.0000e+00	-	1.0000e+00
	F4B	2.4589e-01	1.5099e-11	-
DTLZ2_3D	ASF	-	9.9996e-01	1.0000e+00
	F4A	4.6056e-05	-	7.5411e-01
	F4B	4.1460e-06	2.5057e-01	-
DTLZ3_3D	ASF	-	9.9974e-01	9.9363e-01
	F4A	2.7806e-04	-	4.3250e-01
	F4B	6.6359e-03	5.7331e-01	-
DTLZ4_3D	ASF	-	1.0000e+00	1.0000e+00
	F4A	4.4414e-06	-	5.3828e-01
	F4B	4.0507e-10	4.6760e-01	-
DTLZ5_3D	ASF	-	1.0000e+00	1.0000e+00
	F4A	1.5099e-11	-	2.0913e-09
	F4B	2.7863e-10	1.0000e+00	-
DTLZ6_3D	ASF	-	4.5585e-01	6.2471e-01
	F4A	5.5000e-01	-	6.0208e-01
	F4B	3.8091e-01	4.0364e-01	-
DTLZ7_3D	ASF	-	6.0283e-11	1.0000e+00
	F4A	1.0000e+00	-	1.0000e+00
	F4B	1.5099e-11	1.5099e-11	-
WFG1_3D	ASF	-	6.0283e-11	1.0000e+00
	F4A	1.0000e+00	-	1.0000e+00
	F4B	1.5099e-11	1.5099e-11	-
WFG2_3D	ASF	-	9.7634e-04	9.8243e-01
	F4A	9.9907e-01	-	9.9907e-01
	F4B	1.8219e-02	9.7634e-04	-
WFG3_3D	ASF	-	1.7600e-07	9.1194e-01
	F4A	1.0000e+00	-	1.0000e+00
	F4B	9.0447e-02	2.0556e-07	-
WFG4_3D	ASF	-	9.6136e-01	9.9944e-01
	F4A	3.9891e-02	-	8.5136e-01
	F4B	5.8688e-04	1.5209e-01	-
WFG5_3D	ASF	-	1.8952e-01	9.9484e-01
	F4A	8.1446e-01	-	9.9986e-01
	F4B	5.3813e-03	1.5029e-04	-
WFG6_3D	ASF	-	3.9223e-01	6.4141e-01
	F4A	6.1344e-01	-	7.9404e-01
	F4B	3.6413e-01	2.1019e-01	-
WFG7_3D	ASF	-	5.2652e-01	1.0000e+00
	F4A	4.7937e-01	-	9.9983e-01
	F4B	2.2102e-06	1.7819e-04	-
WFG8_3D	ASF	-	9.6716e-01	9.9986e-01
	F4A	3.3934e-02	-	9.2734e-01
	F4B	1.5029e-04	7.4724e-02	-
WFG9_3D	ASF	-	9.7970e-01	9.9887e-01
	F4A	2.1033e-02	-	7.6786e-01
	F4B	1.1900e-03	2.3667e-01	-
IDTLZ1_3D	ASF	-	1.0000e+00	1.0000e+00
	F4A	3.5542e-09	-	3.5576e-09
	F4B	4.2384e-09	1.0000e+00	-
IDTLZ2_3D	ASF	-	1.0000e+00	1.0000e+00
	F4A	3.5593e-09	-	9.8337e-01
	F4B	3.5593e-09	1.7249e-02	-
IDTLZ3_3D	ASF	-	1.0000e+00	1.0000e+00
	F4A	3.0052e-08	-	5.9063e-01
	F4B	3.8475e-08	4.1513e-01	-
IDTLZ4_3D	ASF	-	1.0000e+00	1.0000e+00
	F4A	2.5362e-10	-	9.9998e-01
	F4B	2.5348e-10	1.7957e-05	-
IDTLZ5_3D	ASF	-	1.0000e+00	1.0000e+00
	F4A	4.2384e-09	-	1.0000e+00
	F4B	3.2544e-09	3.8844e-09	-
IDTLZ6_3D	ASF	-	1.0000e+00	1.0000e+00
	F4A	2.7989e-07	-	1.8550e-02
	F4B	4.4205e-07	9.8211e-01	-
IDTLZ7_3D	ASF	-	3.1150e-03	1.3086e-04
	F4A	9.9702e-01	-	2.6183e-02
	F4B	9.9988e-01	9.7470e-01	-

Tabla E.2: Prueba de suma de rangos de Wilcoxon aplicada al indicador de hipervolumen en problemas de prueba con 3 funciones objetivo.

Problema	Función	ASF	F4A	F4B
DTLZ1_4D	ASF	-	3.0329e-11	3.2602e-01
	F4A	1.0000e+00	-	1.0000e+00
	F4B	6.7929e-01	2.0386e-11	-
DTLZ2_4D	ASF	-	1.0000e+00	1.0000e+00
	F4A	1.0076e-08	-	3.8091e-01
	F4B	7.7904e-09	6.2471e-01	-
DTLZ3_4D	ASF	-	1.6692e-11	1.6276e-01
	F4A	1.0000e+00	-	1.0000e+00
	F4B	8.4085e-01	1.8449e-11	-
DTLZ4_4D	ASF	-	1.0000e+00	1.0000e+00
	F4A	4.0764e-11	-	2.0564e-07
	F4B	4.6301e-09	1.0000e+00	-
DTLZ5_4D	ASF	-	1.0000e+00	1.0000e+00
	F4A	1.5099e-11	-	6.0283e-11
	F4B	1.5099e-11	1.0000e+00	-
DTLZ6_4D	ASF	-	8.0661e-11	3.0500e-01
	F4A	1.0000e+00	-	1.0000e+00
	F4B	7.0016e-01	2.5362e-10	-
DTLZ7_4D	ASF	-	5.4683e-11	1.0000e+00
	F4A	1.0000e+00	-	1.0000e+00
	F4B	6.1423e-10	1.5099e-11	-
WFG1_4D	ASF	-	4.6760e-01	4.4415e-01
	F4A	5.3828e-01	-	4.9115e-01
	F4B	5.6168e-01	5.1474e-01	-
WFG2_4D	ASF	-	4.5316e-08	9.9848e-01
	F4A	1.0000e+00	-	1.0000e+00
	F4B	1.5915e-03	1.0169e-09	-
WFG3_4D	ASF	-	1.0141e-07	6.0777e-01
	F4A	1.0000e+00	-	1.0000e+00
	F4B	3.9792e-01	8.0311e-07	-
WFG4_4D	ASF	-	2.1448e-01	9.4884e-01
	F4A	7.8981e-01	-	9.9308e-01
	F4B	5.2735e-02	7.2061e-03	-
WFG5_4D	ASF	-	9.0712e-01	9.9990e-01
	F4A	9.5365e-02	-	9.9218e-01
	F4B	1.0632e-04	8.1424e-03	-
WFG6_4D	ASF	-	8.3357e-01	4.0937e-01
	F4A	1.7014e-01	-	1.6276e-01
	F4B	5.9636e-01	8.4085e-01	-
WFG7_4D	ASF	-	6.8661e-02	9.8882e-01
	F4A	9.3327e-01	-	9.9957e-01
	F4B	1.1622e-02	4.5153e-04	-
WFG8_4D	ASF	-	9.9999e-01	1.0000e+00
	F4A	5.4534e-06	-	9.4059e-01
	F4B	5.5117e-09	6.1176e-02	-
WFG9_4D	ASF	-	1.0000e+00	9.9950e-01
	F4A	7.7904e-09	-	3.6104e-06
	F4B	5.2878e-04	1.0000e+00	-
IDTLZ1_4D	ASF	-	1.0000e+00	1.0000e+00
	F4A	8.6470e-08	-	3.3137e-01
	F4B	2.0564e-07	6.7398e-01	-
IDTLZ2_4D	ASF	-	1.0000e+00	1.0000e+00
	F4A	2.5362e-10	-	4.1460e-06
	F4B	1.5099e-11	1.0000e+00	-
IDTLZ3_4D	ASF	-	1.0000e+00	1.0000e+00
	F4A	4.2424e-09	-	1.7854e-06
	F4B	4.2424e-09	1.0000e+00	-
IDTLZ4_4D	ASF	-	1.0000e+00	1.0000e+00
	F4A	3.2591e-09	-	1.3403e-04
	F4B	3.8863e-09	9.9987e-01	-
IDTLZ5_4D	ASF	-	1.0000e+00	1.0000e+00
	F4A	1.5099e-11	-	1.8354e-03
	F4B	2.5362e-10	9.9825e-01	-
IDTLZ6_4D	ASF	-	1.0000e+00	1.0000e+00
	F4A	2.3722e-06	-	3.1013e-04
	F4B	2.4990e-09	9.9971e-01	-
IDTLZ7_4D	ASF	-	1.0000e+00	2.6425e-03
	F4A	2.0533e-07	-	2.4227e-07
	F4B	9.9748e-01	1.0000e+00	-

Tabla E.3: Prueba de suma de rangos de Wilcoxon aplicada al indicador de hipervolumen en problemas de prueba con 4 funciones objetivo.

APÉNDICE E. PRUEBA DE SUMA DE RANGOS DE WILCOXON

Problema	Función	ASF	F4A	F4B
DTLZ1_5D	ASF	-	1.8449e-11	1.2441e-01
	F4A	1.0000e+00	-	1.0000e+00
	F4B	8.7859e-01	6.0283e-11	-
DTLZ2_5D	ASF	-	1.0000e+00	1.0000e+00
	F4A	6.5555e-09	-	5.5000e-01
	F4B	8.4736e-10	4.5585e-01	-
DTLZ3_5D	ASF	-	1.1180e-02	9.7316e-01
	F4A	9.8925e-01	-	9.9974e-01
	F4B	2.7773e-02	2.7806e-04	-
DTLZ4_5D	ASF	-	1.0000e+00	1.0000e+00
	F4A	1.5099e-11	-	5.4683e-11
	F4B	2.2863e-09	1.0000e+00	-
DTLZ5_5D	ASF	-	5.9636e-01	1.0000e+00
	F4A	4.0937e-01	-	1.0000e+00
	F4B	3.2639e-08	1.5405e-08	-
DTLZ6_5D	ASF	-	2.3722e-06	8.4264e-01
	F4A	1.0000e+00	-	1.0000e+00
	F4B	1.6095e-01	5.1387e-07	-
DTLZ7_5D	ASF	-	1.0000e+00	1.0000e+00
	F4A	2.4005e-07	-	1.0000e+00
	F4B	1.5099e-11	1.5099e-11	-
WFG1_5D	ASF	-	8.3988e-04	8.5476e-01
	F4A	9.9920e-01	-	9.9998e-01
	F4B	1.4864e-01	1.9153e-05	-
WFG2_5D	ASF	-	6.4352e-10	9.9862e-01
	F4A	1.0000e+00	-	1.0000e+00
	F4B	1.4457e-03	2.0998e-10	-
WFG3_5D	ASF	-	2.2321e-01	8.2986e-01
	F4A	7.8118e-01	-	9.5333e-01
	F4B	1.7391e-01	4.8131e-02	-
WFG4_5D	ASF	-	2.0913e-09	4.2500e-02
	F4A	1.0000e+00	-	1.0000e+00
	F4B	9.5882e-01	6.2704e-08	-
WFG5_5D	ASF	-	7.1035e-01	8.8008e-01
	F4A	2.9473e-01	-	7.5874e-01
	F4B	1.2291e-01	2.4589e-01	-
WFG6_5D	ASF	-	5.0295e-01	8.4085e-01
	F4A	5.0295e-01	-	8.6625e-01
	F4B	1.6276e-01	1.3696e-01	-
WFG7_5D	ASF	-	3.3825e-05	9.4727e-01
	F4A	9.9997e-01	-	1.0000e+00
	F4B	5.4345e-02	4.7666e-07	-
WFG8_5D	ASF	-	5.7681e-02	8.4085e-01
	F4A	9.4401e-01	-	9.9981e-01
	F4B	1.6276e-01	1.9941e-04	-
WFG9_5D	ASF	-	1.0000e+00	9.9950e-01
	F4A	1.5794e-10	-	3.0230e-07
	F4B	5.2878e-04	1.0000e+00	-
IDTLZ1_5D	ASF	-	1.3216e-01	6.1344e-01
	F4A	8.7097e-01	-	9.0463e-01
	F4B	3.9223e-01	9.7895e-02	-
IDTLZ2_5D	ASF	-	9.9999e-01	9.9992e-01
	F4A	1.2164e-05	-	4.1460e-06
	F4B	8.9178e-05	1.0000e+00	-
IDTLZ3_5D	ASF	-	1.0000e+00	1.0000e+00
	F4A	2.5928e-07	-	2.8000e-07
	F4B	3.0230e-07	1.0000e+00	-
IDTLZ4_5D	ASF	-	1.0000e+00	1.0000e+00
	F4A	2.5362e-10	-	2.5362e-10
	F4B	4.1760e-08	1.0000e+00	-
IDTLZ5_5D	ASF	-	9.9984e-01	1.0000e+00
	F4A	1.6840e-04	-	1.5029e-04
	F4B	3.5593e-09	9.9986e-01	-
IDTLZ6_5D	ASF	-	1.0000e+00	1.0000e+00
	F4A	2.7863e-10	-	2.7863e-10
	F4B	3.0230e-07	1.0000e+00	-
IDTLZ7_5D	ASF	-	9.8815e-01	1.5273e-08
	F4A	1.2312e-02	-	1.9764e-08
	F4B	1.0000e+00	1.0000e+00	-

Tabla E.4: Prueba de suma de rangos de Wilcoxon aplicada al indicador de hipervolumen en problemas de prueba con 5 funciones objetivo.

Problema	Función	ASF	F4A	F4B
DTLZ1_6D	ASF	-	1.5099e-11	9.7895e-02
	F4A	1.0000e+00	-	1.0000e+00
	F4B	9.0463e-01	1.5099e-11	-
DTLZ2_6D	ASF	-	1.0000e+00	1.0000e+00
	F4A	2.7470e-11	-	1.7506e-03
	F4B	3.0605e-10	9.9833e-01	-
DTLZ3_6D	ASF	-	4.4431e-10	6.5239e-01
	F4A	1.0000e+00	-	1.0000e+00
	F4B	3.5308e-01	9.7839e-11	-
DTLZ4_6D	ASF	-	1.0000e+00	1.0000e+00
	F4A	1.5099e-11	-	2.2522e-11
	F4B	3.3478e-11	1.0000e+00	-
DTLZ5_6D	ASF	-	6.9158e-03	1.0000e+00
	F4A	9.9336e-01	-	1.0000e+00
	F4B	3.8693e-06	2.7664e-08	-
DTLZ6_6D	ASF	-	5.0885e-01	9.9566e-01
	F4A	4.9705e-01	-	9.6716e-01
	F4B	4.5344e-03	3.3934e-02	-
DTLZ7_6D	ASF	-	1.5099e-11	1.2653e-04
	F4A	1.0000e+00	-	1.0000e+00
	F4B	9.9988e-01	1.6692e-11	-
WFG1_6D	ASF	-	1.9355e-01	2.0596e-01
	F4A	8.1048e-01	-	5.0885e-01
	F4B	7.9823e-01	4.9705e-01	-
WFG2_6D	ASF	-	4.2424e-09	5.6750e-01
	F4A	1.0000e+00	-	1.0000e+00
	F4B	4.3832e-01	2.9822e-09	-
WFG3_6D	ASF	-	4.3832e-01	3.8091e-01
	F4A	5.6750e-01	-	4.5585e-01
	F4B	6.2471e-01	5.5000e-01	-
WFG4_6D	ASF	-	6.5083e-04	7.9404e-01
	F4A	9.9938e-01	-	9.9974e-01
	F4B	2.1019e-01	2.7806e-04	-
WFG5_6D	ASF	-	9.2734e-01	9.9964e-01
	F4A	7.4724e-02	-	9.6011e-01
	F4B	3.8486e-04	4.1179e-02	-
WFG6_6D	ASF	-	6.3587e-01	8.4441e-01
	F4A	3.6970e-01	-	6.8456e-01
	F4B	1.5915e-01	3.2071e-01	-
WFG7_6D	ASF	-	3.3825e-05	8.1839e-01
	F4A	9.9997e-01	-	1.0000e+00
	F4B	1.8554e-01	2.2102e-06	-
WFG8_6D	ASF	-	9.9998e-01	9.9965e-01
	F4A	2.3195e-05	-	1.5559e-01
	F4B	3.6476e-04	8.4791e-01	-
WFG9_6D	ASF	-	1.0000e+00	1.0000e+00
	F4A	3.5215e-07	-	5.2735e-02
	F4B	2.5928e-07	9.4884e-01	-
IDTLZ1_6D	ASF	-	5.9063e-01	5.7331e-01
	F4A	4.1513e-01	-	4.9115e-01
	F4B	4.3250e-01	5.1474e-01	-
IDTLZ2_6D	ASF	-	9.8042e-01	9.6716e-01
	F4A	2.0298e-02	-	4.0364e-01
	F4B	3.3934e-02	6.0208e-01	-
IDTLZ3_6D	ASF	-	8.3357e-01	9.9218e-01
	F4A	1.7014e-01	-	9.1427e-01
	F4B	8.1424e-03	8.8064e-02	-
IDTLZ4_6D	ASF	-	2.9473e-01	5.7331e-01
	F4A	7.1035e-01	-	8.7097e-01
	F4B	4.3250e-01	1.3216e-01	-
IDTLZ5_6D	ASF	-	9.6259e-01	9.2101e-01
	F4A	3.8636e-02	-	3.9792e-01
	F4B	8.1188e-02	6.0777e-01	-
IDTLZ6_6D	ASF	-	9.6259e-01	9.2528e-01
	F4A	3.8636e-02	-	3.8091e-01
	F4B	7.6834e-02	6.2471e-01	-
IDTLZ7_6D	ASF	-	9.6136e-01	2.1961e-06
	F4A	3.9891e-02	-	2.2028e-06
	F4B	1.0000e+00	1.0000e+00	-

Tabla E.5: Prueba de suma de rangos de Wilcoxon aplicada al indicador de hipervolumen en problemas de prueba con 6 funciones objetivo.

APÉNDICE E. PRUEBA DE SUMA DE RANGOS DE WILCOXON

Problema	Función	ASF	F4A	F4B
DTLZ1_10D	ASF	-	3.8475e-08	6.4835e-02
	F4A	1.0000e+00	-	1.0000e+00
	F4B	9.3701e-01	1.9773e-07	-
DTLZ2_10D	ASF	-	4.2090e-01	1.0000e+00
	F4A	5.8487e-01	-	9.9999e-01
	F4B	4.4414e-06	7.6458e-06	-
DTLZ3_10D	ASF	-	1.3008e-08	9.7821e-01
	F4A	1.0000e+00	-	1.0000e+00
	F4B	2.2573e-02	9.3655e-08	-
DTLZ4_10D	ASF	-	1.0000e+00	9.9992e-01
	F4A	1.5405e-08	-	9.3408e-06
	F4B	8.4066e-05	9.9999e-01	-
DTLZ5_10D	ASF	-	8.4774e-03	9.1427e-01
	F4A	9.9186e-01	-	9.9816e-01
	F4B	8.8064e-02	1.9240e-03	-
DTLZ6_10D	ASF	-	9.8306e-01	4.5000e-01
	F4A	1.7568e-02	-	9.1840e-03
	F4B	5.5585e-01	9.9118e-01	-
DTLZ7_10D	ASF	-	2.4990e-09	5.6139e-03
	F4A	1.0000e+00	-	9.9988e-01
	F4B	9.9462e-01	1.2653e-04	-
WFG1_10D	ASF	-	3.5859e-01	1.6276e-01
	F4A	6.4691e-01	-	1.7774e-01
	F4B	8.4085e-01	8.2609e-01	-
WFG2_10D	ASF	-	1.7568e-02	9.9044e-01
	F4A	9.8306e-01	-	9.9993e-01
	F4B	9.9415e-03	7.0334e-05	-
WFG3_10D	ASF	-	6.5784e-01	5.5585e-01
	F4A	3.4761e-01	-	4.9115e-01
	F4B	4.5000e-01	5.1474e-01	-
WFG4_10D	ASF	-	6.6325e-01	9.6823e-01
	F4A	3.4216e-01	-	8.6465e-01
	F4B	3.2836e-02	1.3859e-01	-
WFG5_10D	ASF	-	3.8091e-01	1.0851e-01
	F4A	6.2471e-01	-	2.2321e-01
	F4B	8.9422e-01	7.8118e-01	-
WFG6_10D	ASF	-	3.1766e-02	6.1344e-01
	F4A	9.6927e-01	-	9.5037e-01
	F4B	3.9223e-01	5.1163e-02	-
WFG7_10D	ASF	-	6.1909e-01	8.3724e-01
	F4A	3.8656e-01	-	7.3511e-01
	F4B	1.6643e-01	2.6976e-01	-
WFG8_10D	ASF	-	8.4085e-01	8.2986e-01
	F4A	1.6276e-01	-	4.9705e-01
	F4B	1.7391e-01	5.0885e-01	-
WFG9_10D	ASF	-	9.9722e-01	9.7743e-01
	F4A	2.9141e-03	-	1.4189e-01
	F4B	2.3378e-02	8.6141e-01	-
IDTLZ1_10D	ASF	-	3.0500e-01	1.6643e-01
	F4A	7.0016e-01	-	2.9727e-01
	F4B	8.3724e-01	7.0783e-01	-
IDTLZ2_10D	ASF	-	7.0527e-01	3.3137e-01
	F4A	2.9984e-01	-	2.2321e-01
	F4B	6.7398e-01	7.8118e-01	-
IDTLZ3_10D	ASF	-	3.8373e-01	7.0782e-01
	F4A	6.2190e-01	-	8.4965e-01
	F4B	2.9728e-01	1.5383e-01	-
IDTLZ4_10D	ASF	-	9.7897e-01	5.0295e-01
	F4A	2.1792e-02	-	2.1033e-02
	F4B	5.0295e-01	9.7970e-01	-
IDTLZ5_10D	ASF	-	5.0295e-01	7.2660e-02
	F4A	5.0295e-01	-	6.8661e-02
	F4B	9.2936e-01	9.3327e-01	-
IDTLZ6_10D	ASF	-	2.9984e-01	5.2064e-01
	F4A	7.0527e-01	-	6.7929e-01
	F4B	4.8525e-01	3.2602e-01	-
IDTLZ7_10D	ASF	-	1.4719e-01	2.2429e-01
	F4A	8.5658e-01	-	6.7723e-01
	F4B	7.8096e-01	3.2902e-01	-

Tabla E.6: Prueba de suma de rangos de Wilcoxon aplicada al indicador de hipervolumen en problemas de prueba con 10 funciones objetivo.



## E.2 | ENERGÍA-S

Problema	Función	ASF	F4A	F4B
DTLZ1_2D	ASF	-	1.8161e-01	7.7235e-01
	F4A	8.2226e-01	-	9.4059e-01
	F4B	2.3214e-01	6.1176e-02	-
DTLZ2_2D	ASF	-	6.6325e-01	9.9118e-01
	F4A	3.4216e-01	-	9.9152e-01
	F4B	9.1840e-03	8.8245e-03	-
DTLZ3_2D	ASF	-	1.2291e-01	5.0295e-01
	F4A	8.8008e-01	-	9.2101e-01
	F4B	5.0295e-01	8.1188e-02	-
DTLZ4_2D	ASF	-	2.5601e-04	9.9964e-01
	F4A	9.9976e-01	-	9.9998e-01
	F4B	3.8486e-04	1.6841e-05	-
DTLZ5_2D	ASF	-	6.6325e-01	9.9118e-01
	F4A	3.4216e-01	-	9.9152e-01
	F4B	9.1840e-03	8.8245e-03	-
DTLZ6_2D	ASF	-	1.3535e-01	6.0777e-01
	F4A	8.6784e-01	-	7.7235e-01
	F4B	3.9792e-01	2.3214e-01	-
DTLZ7_2D	ASF	-	1.2987e-05	3.3675e-01
	F4A	9.9999e-01	-	9.9993e-01
	F4B	6.6863e-01	7.0334e-05	-
WFG1_2D	ASF	-	3.6223e-02	2.4589e-01
	F4A	9.6494e-01	-	7.9404e-01
	F4B	7.5874e-01	2.1019e-01	-
WFG2_2D	ASF	-	9.4232e-01	6.7929e-01
	F4A	5.9409e-02	-	1.0047e-01
	F4B	3.2602e-01	9.0210e-01	-
WFG3_2D	ASF	-	1.7391e-01	3.8656e-01
	F4A	8.2986e-01	-	6.6325e-01
	F4B	6.1909e-01	3.4216e-01	-
WFG4_2D	ASF	-	9.9972e-01	5.4415e-01
	F4A	2.9369e-04	-	6.8517e-04
	F4B	4.6172e-01	9.9935e-01	-
WFG5_2D	ASF	-	9.2101e-01	2.3214e-01
	F4A	8.1188e-02	-	1.8891e-02
	F4B	7.7235e-01	9.8178e-01	-
WFG6_2D	ASF	-	9.9279e-01	9.2528e-01
	F4A	7.5071e-03	-	1.0047e-01
	F4B	7.6834e-02	9.0210e-01	-
WFG7_2D	ASF	-	9.9789e-01	5.6168e-01
	F4A	2.2136e-03	-	4.5344e-03
	F4B	4.4415e-01	9.9566e-01	-
WFG8_2D	ASF	-	1.9355e-01	7.8988e-02
	F4A	8.1048e-01	-	2.7962e-01
	F4B	9.2317e-01	7.2534e-01	-
WFG9_2D	ASF	-	8.4441e-01	5.6750e-01
	F4A	1.5915e-01	-	2.6489e-01
	F4B	4.3832e-01	7.3993e-01	-
IDTLZ1_2D	ASF	-	3.6136e-01	3.9223e-01
	F4A	6.4417e-01	-	5.5000e-01
	F4B	6.1344e-01	4.5585e-01	-
IDTLZ2_2D	ASF	-	8.9394e-07	9.9768e-01
	F4A	1.0000e+00	-	1.0000e+00
	F4B	2.4268e-03	2.8000e-07	-
IDTLZ3_2D	ASF	-	3.5215e-07	9.9985e-01
	F4A	1.0000e+00	-	1.0000e+00
	F4B	1.5910e-04	2.5456e-06	-
IDTLZ4_2D	ASF	-	3.0121e-07	9.9998e-01
	F4A	1.0000e+00	-	1.0000e+00
	F4B	1.7957e-05	3.6094e-06	-
IDTLZ5_2D	ASF	-	8.9394e-07	9.9768e-01
	F4A	1.0000e+00	-	1.0000e+00
	F4B	2.4268e-03	2.8000e-07	-
IDTLZ6_2D	ASF	-	2.7309e-09	1.0000e+00
	F4A	1.0000e+00	-	1.0000e+00
	F4B	3.0230e-07	3.5593e-09	-
IDTLZ7_2D	ASF	-	6.8661e-02	8.0848e-01
	F4A	9.3327e-01	-	9.8367e-01
	F4B	1.9558e-01	1.6937e-02	-

Tabla E.7: Prueba de suma de rangos de Wilcoxon aplicada al indicador de energía-s en problemas de prueba con 2 funciones objetivo.

Problema	Función	ASF	F4A	F4B
DTLZ1_3D	ASF	-	4.4414e-06	9.6011e-01
	F4A	1.0000e+00	-	1.0000e+00
	F4B	4.1179e-02	1.9124e-09	-
DTLZ2_3D	ASF	-	4.9705e-01	7.2038e-01
	F4A	5.0885e-01	-	7.5874e-01
	F4B	2.8461e-01	2.4589e-01	-
DTLZ3_3D	ASF	-	3.1544e-01	7.0527e-01
	F4A	6.8980e-01	-	7.4943e-01
	F4B	2.9984e-01	2.5530e-01	-
DTLZ4_3D	ASF	-	8.6784e-01	9.8306e-01
	F4A	1.3535e-01	-	8.6465e-01
	F4B	1.7568e-02	1.3859e-01	-
DTLZ5_3D	ASF	-	4.2669e-01	1.7568e-02
	F4A	5.7910e-01	-	2.3378e-02
	F4B	9.8306e-01	9.7743e-01	-
DTLZ6_3D	ASF	-	9.9992e-01	9.9996e-01
	F4A	8.9178e-05	-	6.6863e-01
	F4B	3.8294e-05	3.3675e-01	-
DTLZ7_3D	ASF	-	5.9063e-01	3.8656e-01
	F4A	4.1513e-01	-	3.2071e-01
	F4B	6.1909e-01	6.8456e-01	-
WFG1_3D	ASF	-	9.8966e-01	5.2652e-01
	F4A	1.0753e-02	-	8.8245e-03
	F4B	4.7937e-01	9.9152e-01	-
WFG2_3D	ASF	-	6.4835e-02	7.8552e-01
	F4A	9.3701e-01	-	9.8966e-01
	F4B	2.1882e-01	1.0753e-02	-
WFG3_3D	ASF	-	9.9875e-01	9.9722e-01
	F4A	1.3121e-03	-	5.9063e-01
	F4B	2.9141e-03	4.1513e-01	-
WFG4_3D	ASF	-	1.0000e+00	9.9959e-01
	F4A	2.5928e-07	-	3.5063e-02
	F4B	4.2821e-04	9.6607e-01	-
WFG5_3D	ASF	-	9.9944e-01	9.9798e-01
	F4A	5.8688e-04	-	2.3667e-01
	F4B	2.1130e-03	7.6786e-01	-
WFG6_3D	ASF	-	9.9848e-01	9.9920e-01
	F4A	1.5915e-03	-	2.4589e-01
	F4B	8.3988e-04	7.5874e-01	-
WFG7_3D	ASF	-	7.7235e-01	8.5811e-01
	F4A	2.3214e-01	-	5.9636e-01
	F4B	1.4524e-01	4.0937e-01	-
WFG8_3D	ASF	-	9.7662e-01	6.9500e-01
	F4A	2.4207e-02	-	7.4724e-02
	F4B	3.1020e-01	9.2734e-01	-
WFG9_3D	ASF	-	9.9218e-01	9.9875e-01
	F4A	8.1424e-03	-	6.3587e-01
	F4B	1.3121e-03	3.6970e-01	-
IDTLZ1_3D	ASF	-	9.9991e-01	9.9999e-01
	F4A	9.4581e-05	-	6.0208e-01
	F4B	1.3863e-05	4.0364e-01	-
IDTLZ2_3D	ASF	-	9.9999e-01	9.9998e-01
	F4A	7.6458e-06	-	8.4085e-01
	F4B	2.4713e-05	1.6276e-01	-
IDTLZ3_3D	ASF	-	9.9988e-01	9.9975e-01
	F4A	1.2653e-04	-	3.1544e-01
	F4B	2.6320e-04	6.8980e-01	-
IDTLZ4_3D	ASF	-	9.7970e-01	9.9999e-01
	F4A	2.1033e-02	-	9.9993e-01
	F4B	6.6834e-06	7.9230e-05	-
IDTLZ5_3D	ASF	-	1.0000e+00	6.0777e-01
	F4A	5.3328e-08	-	2.7664e-08
	F4B	3.9792e-01	1.0000e+00	-
IDTLZ6_3D	ASF	-	9.9984e-01	9.9833e-01
	F4A	1.6840e-04	-	5.7331e-01
	F4B	1.7506e-03	4.3250e-01	-
IDTLZ7_3D	ASF	-	9.9979e-01	9.9778e-01
	F4A	2.2296e-04	-	8.6824e-01
	F4B	2.3251e-03	1.3495e-01	-

Tabla E.8: Prueba de suma de rangos de Wilcoxon aplicada al indicador de energía-s en problemas de prueba con 3 funciones objetivo.

APÉNDICE E. PRUEBA DE SUMA DE RANGOS DE WILCOXON

Problema	Función	ASF	F4A	F4B
DTLZ1_4D	ASF	-	2.1882e-01	2.3214e-01
	F4A	7.8552e-01	-	6.0777e-01
	F4B	7.7235e-01	3.9792e-01	-
DTLZ2_4D	ASF	-	5.7681e-02	6.4141e-01
	F4A	9.4401e-01	-	9.8111e-01
	F4B	3.6413e-01	1.9584e-02	-
DTLZ3_4D	ASF	-	2.3667e-01	5.0295e-01
	F4A	7.6786e-01	-	7.8981e-01
	F4B	5.0295e-01	2.1448e-01	-
DTLZ4_4D	ASF	-	8.5136e-01	9.8646e-01
	F4A	1.5209e-01	-	9.4059e-01
	F4B	1.4064e-02	6.1176e-02	-
DTLZ5_4D	ASF	-	6.8980e-01	8.4085e-01
	F4A	3.1544e-01	-	5.7331e-01
	F4B	1.6276e-01	4.3250e-01	-
DTLZ6_4D	ASF	-	8.0645e-01	7.3993e-01
	F4A	1.9763e-01	-	3.1544e-01
	F4B	2.6489e-01	6.8980e-01	-
DTLZ7_4D	ASF	-	8.5725e-02	5.5585e-01
	F4A	9.1657e-01	-	9.2101e-01
	F4B	4.5000e-01	8.1188e-02	-
WFG1_4D	ASF	-	8.3357e-01	2.7466e-01
	F4A	1.7014e-01	-	6.1176e-02
	F4B	7.3024e-01	9.4059e-01	-
WFG2_4D	ASF	-	7.5890e-04	6.9500e-01
	F4A	9.9928e-01	-	9.9989e-01
	F4B	3.1020e-01	1.1942e-04	-
WFG3_4D	ASF	-	9.9584e-01	5.3240e-01
	F4A	4.3422e-03	-	1.0753e-02
	F4B	4.7348e-01	9.8966e-01	-
WFG4_4D	ASF	-	9.9484e-01	6.1909e-01
	F4A	5.3813e-03	-	2.8730e-02
	F4B	3.8656e-01	9.7223e-01	-
WFG5_4D	ASF	-	8.4441e-01	9.9336e-01
	F4A	1.5915e-01	-	9.0712e-01
	F4B	6.9158e-03	9.5365e-02	-
WFG6_4D	ASF	-	7.3024e-01	9.3516e-01
	F4A	2.7466e-01	-	7.6786e-01
	F4B	6.6727e-02	2.3667e-01	-
WFG7_4D	ASF	-	7.4724e-02	2.1019e-01
	F4A	9.2734e-01	-	5.9063e-01
	F4B	7.9404e-01	4.1513e-01	-
WFG8_4D	ASF	-	1.0000e+00	9.9985e-01
	F4A	2.9293e-06	-	2.2573e-02
	F4B	1.5910e-04	9.7821e-01	-
WFG9_4D	ASF	-	4.6172e-01	9.6927e-01
	F4A	5.4415e-01	-	9.3701e-01
	F4B	3.1766e-02	6.4835e-02	-
IDTLZ1_4D	ASF	-	2.3667e-01	3.2071e-01
	F4A	7.6786e-01	-	6.2471e-01
	F4B	6.8456e-01	3.8091e-01	-
IDTLZ2_4D	ASF	-	4.3250e-01	8.2226e-01
	F4A	5.7331e-01	-	8.8588e-01
	F4B	1.8161e-01	1.1699e-01	-
IDTLZ3_4D	ASF	-	1.5915e-01	3.1020e-01
	F4A	8.4441e-01	-	6.1344e-01
	F4B	6.9500e-01	3.9223e-01	-
IDTLZ4_4D	ASF	-	5.0295e-01	2.2321e-01
	F4A	5.0295e-01	-	1.4524e-01
	F4B	7.8118e-01	8.5811e-01	-
IDTLZ5_4D	ASF	-	8.4791e-01	7.6333e-01
	F4A	1.5559e-01	-	5.0295e-01
	F4B	2.4126e-01	5.0295e-01	-
IDTLZ6_4D	ASF	-	1.6276e-01	5.3828e-01
	F4A	8.4085e-01	-	9.2936e-01
	F4B	4.6760e-01	7.2660e-02	-
IDTLZ7_4D	ASF	-	8.7406e-01	9.9951e-01
	F4A	1.2903e-01	-	9.9924e-01
	F4B	5.1389e-04	8.0281e-04	-

Tabla E.9: Prueba de suma de rangos de Wilcoxon aplicada al indicador de energía-s en problemas de prueba con 4 funciones objetivo.

Problema	Función	ASF	F4A	F4B
DTLZ1_5D	ASF	-	5.2652e-01	7.8118e-01
	F4A	4.7937e-01	-	7.7679e-01
	F4B	2.2321e-01	2.2765e-01	-
DTLZ2_5D	ASF	-	7.4943e-01	6.9500e-01
	F4A	2.5530e-01	-	4.8526e-01
	F4B	3.1020e-01	5.2063e-01	-
DTLZ3_5D	ASF	-	4.5000e-01	2.6489e-01
	F4A	5.5585e-01	-	2.9984e-01
	F4B	7.3993e-01	7.0527e-01	-
DTLZ4_5D	ASF	-	8.1188e-02	1.5159e-02
	F4A	9.2101e-01	-	4.2669e-01
	F4B	9.8540e-01	5.7910e-01	-
DTLZ5_5D	ASF	-	3.6442e-03	5.2063e-01
	F4A	9.9651e-01	-	9.8792e-01
	F4B	4.8526e-01	1.2551e-02	-
DTLZ6_5D	ASF	-	1.3543e-02	1.6643e-01
	F4A	9.8696e-01	-	9.1881e-01
	F4B	8.3724e-01	8.3433e-02	-
DTLZ7_5D	ASF	-	5.6750e-01	9.9389e-01
	F4A	4.3832e-01	-	9.9336e-01
	F4B	6.3661e-03	6.9158e-03	-
WFG1_5D	ASF	-	9.8925e-01	9.5614e-01
	F4A	1.1180e-02	-	5.2063e-01
	F4B	4.5245e-02	4.8526e-01	-
WFG2_5D	ASF	-	1.5786e-05	9.0210e-01
	F4A	9.9999e-01	-	1.0000e+00
	F4B	1.0047e-01	9.3042e-07	-
WFG3_5D	ASF	-	9.8966e-01	4.0364e-01
	F4A	1.0753e-02	-	4.5344e-03
	F4B	6.0208e-01	9.9566e-01	-
WFG4_5D	ASF	-	9.8966e-01	6.9500e-01
	F4A	1.0753e-02	-	4.2500e-02
	F4B	3.1020e-01	9.5882e-01	-
WFG5_5D	ASF	-	9.8178e-01	9.3327e-01
	F4A	1.8891e-02	-	3.4216e-01
	F4B	6.8661e-02	6.6325e-01	-
WFG6_5D	ASF	-	8.7709e-01	7.4470e-01
	F4A	1.2594e-01	-	2.6976e-01
	F4B	2.6007e-01	7.3511e-01	-
WFG7_5D	ASF	-	8.4791e-01	4.5000e-01
	F4A	1.5559e-01	-	1.1699e-01
	F4B	5.5585e-01	8.8588e-01	-
WFG8_5D	ASF	-	8.7406e-01	4.3250e-01
	F4A	1.2903e-01	-	1.0851e-01
	F4B	5.7331e-01	8.9422e-01	-
WFG9_5D	ASF	-	5.4415e-01	5.4415e-01
	F4A	4.6172e-01	-	4.7937e-01
	F4B	4.6172e-01	5.2652e-01	-
IDTLZ1_5D	ASF	-	6.8980e-01	7.8118e-01
	F4A	3.1544e-01	-	6.0208e-01
	F4B	2.2321e-01	4.0364e-01	-
IDTLZ2_5D	ASF	-	7.8981e-01	2.3667e-01
	F4A	2.1448e-01	-	1.0578e-01
	F4B	7.6786e-01	8.9690e-01	-
IDTLZ3_5D	ASF	-	6.5239e-01	1.6643e-01
	F4A	3.5309e-01	-	1.1992e-01
	F4B	8.3724e-01	8.8301e-01	-
IDTLZ4_5D	ASF	-	2.6007e-01	2.6007e-01
	F4A	7.4470e-01	-	4.1513e-01
	F4B	7.4470e-01	5.9063e-01	-
IDTLZ5_5D	ASF	-	2.2321e-01	6.0208e-01
	F4A	7.8118e-01	-	8.3357e-01
	F4B	4.0364e-01	1.7014e-01	-
IDTLZ6_5D	ASF	-	4.5000e-01	7.2534e-01
	F4A	5.5585e-01	-	7.2534e-01
	F4B	2.7962e-01	2.7962e-01	-
IDTLZ7_5D	ASF	-	9.9955e-01	9.9923e-01
	F4A	4.7596e-04	-	9.9822e-01
	F4B	8.1440e-04	1.8710e-03	-

Tabla E.10: Prueba de suma de rangos de Wilcoxon aplicada al indicador de energía-s en problemas de prueba con 5 funciones objetivo.

APÉNDICE E. PRUEBA DE SUMA DE RANGOS DE WILCOXON

Problema	Función	ASF	F4A	F4B
DTLZ1_6D	ASF	-	6.8980e-01	6.1909e-01
	F4A	3.1544e-01	-	3.9223e-01
	F4B	3.8656e-01	6.1344e-01	-
DTLZ2_6D	ASF	-	9.9955e-01	9.9923e-01
	F4A	4.7596e-04	-	9.9822e-01
	F4B	8.1440e-04	1.8710e-03	-
DTLZ3_6D	ASF	-	5.3240e-01	8.9149e-01
	F4A	4.7348e-01	-	8.4791e-01
	F4B	1.1129e-01	1.5559e-01	-
DTLZ4_6D	ASF	-	8.3357e-01	9.6607e-01
	F4A	1.7014e-01	-	7.2038e-01
	F4B	3.5063e-02	2.8461e-01	-
DTLZ5_6D	ASF	-	9.6378e-01	9.9484e-01
	F4A	3.7414e-02	-	6.4691e-01
	F4B	5.3813e-03	3.5859e-01	-
DTLZ6_6D	ASF	-	2.3378e-02	2.2765e-01
	F4A	9.7743e-01	-	9.0712e-01
	F4B	7.7679e-01	9.5365e-02	-
DTLZ7_6D	ASF	-	7.2660e-02	2.8965e-01
	F4A	9.2936e-01	-	7.9404e-01
	F4B	7.1539e-01	2.1019e-01	-
WFG1_6D	ASF	-	1.5209e-01	3.0500e-01
	F4A	8.5136e-01	-	7.6333e-01
	F4B	7.0016e-01	2.4126e-01	-
WFG2_6D	ASF	-	2.8036e-05	4.2090e-01
	F4A	9.9997e-01	-	9.9997e-01
	F4B	5.8487e-01	3.1780e-05	-
WFG3_6D	ASF	-	8.8064e-02	3.1544e-01
	F4A	9.1427e-01	-	8.1446e-01
	F4B	6.8980e-01	1.8952e-01	-
WFG4_6D	ASF	-	6.6325e-01	7.6786e-01
	F4A	3.4216e-01	-	4.0364e-01
	F4B	2.3667e-01	6.0208e-01	-
WFG5_6D	ASF	-	5.6750e-01	9.0955e-01
	F4A	4.3832e-01	-	8.9149e-01
	F4B	9.2883e-02	1.1129e-01	-
WFG6_6D	ASF	-	8.8588e-01	8.0237e-01
	F4A	1.1699e-01	-	3.0500e-01
	F4B	2.0177e-01	7.0016e-01	-
WFG7_6D	ASF	-	6.8980e-01	2.1882e-01
	F4A	3.1544e-01	-	9.7895e-02
	F4B	7.8552e-01	9.0463e-01	-
WFG8_6D	ASF	-	9.9825e-01	6.4141e-01
	F4A	1.8354e-03	-	3.0726e-02
	F4B	3.6413e-01	9.7029e-01	-
WFG9_6D	ASF	-	3.6413e-01	7.0527e-01
	F4A	6.4141e-01	-	8.3724e-01
	F4B	2.9984e-01	1.6643e-01	-
IDTLZ1_6D	ASF	-	1.9763e-01	2.4126e-01
	F4A	8.0645e-01	-	4.6172e-01
	F4B	7.6333e-01	5.4415e-01	-
IDTLZ2_6D	ASF	-	7.0016e-01	9.2101e-01
	F4A	3.0500e-01	-	7.6786e-01
	F4B	8.1188e-02	2.3667e-01	-
IDTLZ3_6D	ASF	-	7.3511e-01	7.0527e-01
	F4A	2.6976e-01	-	4.6172e-01
	F4B	2.9984e-01	5.4415e-01	-
IDTLZ4_6D	ASF	-	9.2528e-01	5.7331e-01
	F4A	7.6834e-02	-	1.4864e-01
	F4B	4.3250e-01	8.5476e-01	-
IDTLZ5_6D	ASF	-	1.1699e-01	5.6750e-01
	F4A	8.8588e-01	-	9.1427e-01
	F4B	4.3832e-01	8.8064e-02	-
IDTLZ6_6D	ASF	-	4.2669e-01	4.5000e-01
	F4A	5.7910e-01	-	5.0885e-01
	F4B	5.5585e-01	4.9705e-01	-
IDTLZ7_6D	ASF	-	9.3707e-01	7.2418e-03
	F4A	6.4782e-02	-	3.9275e-03
	F4B	9.9305e-01	9.9624e-01	-

Tabla E.11: Prueba de suma de rangos de Wilcoxon aplicada al indicador de energía-s en problemas de prueba con 6 funciones objetivo.

Problema	Función	ASF	F4A	F4B
DTLZ1_10D	ASF	-	4.8959e-05	1.8554e-01
	F4A	9.9995e-01	-	9.9993e-01
	F4B	8.1839e-01	7.9230e-05	-
DTLZ2_10D	ASF	-	9.6927e-01	9.5187e-01
	F4A	3.1766e-02	-	4.6172e-01
	F4B	4.9629e-02	5.4415e-01	-
DTLZ3_10D	ASF	-	8.2609e-01	1.5209e-01
	F4A	1.7774e-01	-	3.7414e-02
	F4B	8.5136e-01	9.6378e-01	-
DTLZ4_10D	ASF	-	1.3859e-01	1.2903e-01
	F4A	8.6465e-01	-	4.9115e-01
	F4B	8.7406e-01	5.1474e-01	-
DTLZ5_10D	ASF	-	6.1809e-04	2.3378e-02
	F4A	9.9941e-01	-	8.1446e-01
	F4B	9.7743e-01	1.8952e-01	-
DTLZ6_10D	ASF	-	2.9473e-01	7.1471e-09
	F4A	7.1035e-01	-	7.9820e-08
	F4B	1.0000e+00	1.0000e+00	-
DTLZ7_10D	ASF	-	4.8131e-02	2.6976e-01
	F4A	9.5333e-01	-	8.4791e-01
	F4B	7.3511e-01	1.5559e-01	-
WFG1_10D	ASF	-	3.4216e-01	4.3832e-01
	F4A	6.6325e-01	-	6.0777e-01
	F4B	5.6750e-01	3.9792e-01	-
WFG2_10D	ASF	-	6.9158e-03	2.9473e-01
	F4A	9.9336e-01	-	9.8306e-01
	F4B	7.1035e-01	1.7568e-02	-
WFG3_10D	ASF	-	9.2883e-02	7.7235e-01
	F4A	9.0955e-01	-	9.9279e-01
	F4B	2.3214e-01	7.5071e-03	-
WFG4_10D	ASF	-	5.0295e-01	5.0885e-01
	F4A	5.0295e-01	-	4.9115e-01
	F4B	4.9705e-01	5.1474e-01	-
WFG5_10D	ASF	-	8.8008e-01	5.5585e-01
	F4A	1.2291e-01	-	1.7391e-01
	F4B	4.5000e-01	8.2986e-01	-
WFG6_10D	ASF	-	9.9308e-01	7.0016e-01
	F4A	7.2061e-03	-	2.2573e-02
	F4B	3.0500e-01	9.7821e-01	-
WFG7_10D	ASF	-	8.1048e-01	8.8871e-01
	F4A	1.9355e-01	-	6.4691e-01
	F4B	1.1412e-01	3.5859e-01	-
WFG8_10D	ASF	-	9.4727e-01	9.9118e-01
	F4A	5.4345e-02	-	8.8588e-01
	F4B	9.1840e-03	1.1699e-01	-
WFG9_10D	ASF	-	9.7821e-01	8.3724e-01
	F4A	2.2573e-02	-	1.3535e-01
	F4B	1.6643e-01	8.6784e-01	-
IDTLZ1_10D	ASF	-	1.4524e-01	2.3214e-01
	F4A	8.5811e-01	-	6.8194e-01
	F4B	7.7235e-01	3.2336e-01	-
IDTLZ2_10D	ASF	-	8.9422e-01	5.0885e-01
	F4A	1.0851e-01	-	8.8064e-02
	F4B	4.9705e-01	9.1427e-01	-
IDTLZ3_10D	ASF	-	7.6333e-01	7.3511e-01
	F4A	2.4126e-01	-	4.9705e-01
	F4B	2.6976e-01	5.0885e-01	-
IDTLZ4_10D	ASF	-	7.8552e-01	5.0295e-01
	F4A	2.1882e-01	-	2.1882e-01
	F4B	5.0295e-01	7.8552e-01	-
IDTLZ5_10D	ASF	-	3.4210e-01	3.6970e-01
	F4A	6.6331e-01	-	5.3240e-01
	F4B	6.3587e-01	4.7348e-01	-
IDTLZ6_10D	ASF	-	4.2090e-01	7.6793e-01
	F4A	5.8487e-01	-	7.8118e-01
	F4B	2.3661e-01	2.2321e-01	-
IDTLZ7_10D	ASF	-	8.4260e-01	9.2118e-01
	F4A	1.6255e-01	-	6.6915e-01
	F4B	8.2087e-02	3.4052e-01	-

Tabla E.12: Prueba de suma de rangos de Wilcoxon aplicada al indicador de energía-s en problemas de prueba con 10 funciones objetivo.



# Bibliografía

- [1] Eigen (c++ linear algebra library). [http://eigen.tuxfamily.org/index.php?title=Main\\_Page](http://eigen.tuxfamily.org/index.php?title=Main_Page). Accessed: 2019-09-26.
- [2] Alp Akcay, Gurdal Ertek, and Gülçin Büyüközkan. Analyzing the solutions of dea through information visualization and data mining techniques: Smartdea framework. *Expert Systems with Applications*, 39:77637775, 07 2012.
- [3] Howard Anton. Calculus with analytic geometry. *The Mathematical Gazette*, 73, 06 1989.
- [4] Thomas Back. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, 1996.
- [5] J. Bader. *Hypervolume-Based Search for Multiobjective Optimization: Theory and Methods*. PhD thesis, Swiss Federal Institute of Technology (ETH), 01 2010.
- [6] J. Buchanan and L. Gardiner. A comparison of two reference point methods in multiple objective mathematical programming. *European Journal of Operational Research*, 149:17–34, 08 2003.
- [7] Ran Cheng, Miqing Li, Ye Tian, Xingyi Zhang, Shengxiang Yang, Yaochu Jin, and Xin Yao. A benchmark test suite for evolutionary many-objective optimization. *Complex & Intelligent Systems*, 3(1):67–81, March 2017.
- [8] Chih-Ming Chen, Ying-ping Chen, and Qingfu Zhang. Enhancing MOEA/D with guided mutation and priority update for multi-objective optimization. In *2009 IEEE Congress on Evolutionary Computation*, pages 209–216, May 2009.
- [9] Carlos A. Coello Coello and Gary B Lamont. *Applications of multi-objective evolutionary algorithms*, volume 1. World Scientific, 2004.
- [10] Carlos A. Coello Coello, Gary B Lamont, and David A et. al. Van Veldhuizen. *Evolutionary algorithms for solving multi-objective problems*, volume 5. Springer, 2007.

- [11] Carlos A. Coello Coello. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems*, 1(3):269–308, Aug 1999.
- [12] David W. Corne and Joshua D. Knowles. Techniques for highly multi-objective optimisation: Some nondominated points are better than others. *CoRR*, abs/0908.3025, 2009.
- [13] Michael L. Cramer. A representation for the adaptive generation of simple sequential programs. In *Proceedings of the first international conference on genetic algorithms*, pages 183–187, 1985.
- [14] K. Deb. Multi-objective optimization. In *Search Methodologies*, pages 403–449. Springer US, jul 2013.
- [15] K. Deb and H. Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, Aug 2014.
- [16] Kalyanmoy Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 2001.
- [17] Kalyanmoy Deb and Kaisa Miettinen. Nadir point estimation using evolutionary approaches: Better accuracy and computational speed through focused search. In *Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems*, pages 339–354, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
- [18] Kalyanmoy Deb and J. Sundar. Reference point based multi-objective optimization using evolutionary algorithms. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, GECCO '06*, pages 635–642, New York, NY, USA, 2006. ACM.
- [19] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary multiobjective optimization*, pages 105–145. Springer, 2005.
- [20] M. Farina and P. Amato. On the optimal solution definition for many-criteria optimization problems. In *2002 Annual Meeting of the North American Fuzzy Information Processing Society Proceedings.*, pages 233–238, June 2002.

- [21] M. Fleischer. The measure of pareto optima applications to multi-objective metaheuristics. In *Lecture Notes in Computer Science*, pages 519–533. Springer Berlin Heidelberg, 2003.
- [22] David Fogel. Artificial intelligence through simulated evolution. *Evolutionary Computation: The Fossil Record*, pages 227–296, 11 2009.
- [23] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13:2171–2175, jul 2012.
- [24] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming. Towards understanding the cost of adaptation in decomposition-based optimization algorithms. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 615–620, Oct 2013.
- [25] Fred Glover and Manuel Laguna. Tabu search. In *Handbook of combinatorial optimization*, pages 2093–2229. Springer, 1998.
- [26] Y. Sakane H. Ishibuchi, N. Tsukamoto and Y. Nojima. Indicator-based evolutionary algorithm with hypervolume approximation by achievement scalarizing functions. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation - GECCO '10*. ACM Press, 2010.
- [27] M. P. Hansen and A. Jaszkiwicz. Evaluating the quality of approximations to the non-dominated set, March 1998.
- [28] D. Hardin and E. SAFF. Discretizing manifolds via minimum energy points. *Notices of the American Mathematical Society*, 51, 11 2004.
- [29] Raquel Hernández Gómez and Carlos A Coello Coello. Mombi: A new metaheuristic for many-objective optimization based on the R2 indicator. In *2013 IEEE Congress on Evolutionary Computation*, pages 2488–2495. IEEE, 2013.
- [30] Raquel Hernández Gómez and Carlos A Coello Coello. *MOMBI: A New Metaheuristic for Many-Objective Optimization Based on the R2 Indicator*. PhD thesis, CINVESTAV-IPN, 2013.
- [31] Raquel Hernández Gómez and Carlos A Coello Coello. Improved metaheuristic based on the r2 indicator for many-objective optimization. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 679–686. ACM, 2015.

- [32] Raquel Hernández Gómez and Carlos A Coello Coello. *Parallel Hyper-Heuristics for Multi-Objective Optimization*. PhD thesis, CINVESTAV-IPN, 2018.
- [33] John H. Holland. Concerning efficient adaptive systems. *Self-Organizing Systems*, 230, 1962.
- [34] R Holliday. The inheritance of epigenetic defects. *Science*, 238(4824):163–170, 1987.
- [35] Simon Huband, Luigi Barone, Lyndon While, and Phil Hingston. A scalable multi-objective test problem toolkit. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization*, pages 280–295, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
- [36] Evan Hughes. Evolutionary many-objective optimisation: Many once or one many? volume 1, pages 222 – 227 Vol.1, 10 2005.
- [37] H. Ishibuchi, Y. Nojima, and T. Doi. Comparison between single-objective and multi-objective genetic algorithms: Performance comparison and performance measures. pages 1143 – 1150, 01 2006.
- [38] Hisao Ishibuchi and Yusuke Nojima. Optimization of scalarizing functions through evolutionary multiobjective optimization. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Evolutionary Multi-Criterion Optimization*, pages 51–65, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
- [39] Hisao Ishibuchi, Yuji Sakane, Noritaka Tsukamoto, and Yusuke Nojima. Adaptation of scalarizing functions in MOEA/D: An adaptive scalarizing function-based multiobjective evolutionary algorithm. In Matthias Ehrgott, Carlos M. Fonseca, Xavier Gandibleux, Jin-Kao Hao, and Marc Sevaux, editors, *Evolutionary Multi-Criterion Optimization*, pages 438–452, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
- [40] Hisao Ishibuchi, Yuji Sakane, Noritaka Tsukamoto, and Yusuke Nojima. Simultaneous use of different scalarizing functions in MOEA/D. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO '10*, pages 519–526, New York, NY, USA, 2010. ACM.

- 
- [41] Hisao Ishibuchi, Yu Setoguchi, Hiroyuki Masuda, and Yusuke Nojima. Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes. *IEEE Transactions on Evolutionary Computation*, 21:1–1, 01 2016.
- [42] Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. Evolutionary many-objective optimization: A short review. In *2008 IEEE Congress on Evolutionary Computation*, pages 2419–2426, June 2008.
- [43] H. Jain and K. Deb. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach. *IEEE Transactions on Evolutionary Computation*, 18(4):602–622, Aug 2014.
- [44] J Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of IEEE international conference on neural networks*, volume 4, pages 1942–1948. IEEE Press, 1995.
- [45] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.
- [46] Mario Köppen and Kaori Yoshida. Visualization of pareto-sets in evolutionary multi-objective optimization. *7th International Conference on Hybrid Intelligent Systems (HIS 2007)*, pages 156–161, 2007.
- [47] John R. Koza. Hierarchical genetic algorithms operating on populations of computer programs. In *IJCAI*, volume 89, pages 768–774. Springer-Verlag, 1989.
- [48] John R. Koza. *Genetic programming: on the programming of computers by means of natural selection*, volume 1. MIT press, 1992.
- [49] Saku Kukkonen and Jouni Lampinen. Ranking-dominance and many-objective optimization. *2007 IEEE Congress on Evolutionary Computation*, pages 3983–3990, 2007.
- [50] William La Cava, Thomas Helmuth, Lee Spector, and Kourosh Danai. Genetic programming with epigenetic local search. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO '15*, pages 1055–1062, New York, NY, USA, 2015. ACM.
- [51] Sean Luke. ECJ evolutionary computation library, 1998. Available for free at <http://cs.gmu.edu/~eclab/projects/ecj/>.

- [52] K. Miettinen and M. M. Mäkelä. On scalarizing functions in multiobjective optimization. *OR Spectrum*, 24(2):193–213, May 2002.
- [53] Kaisa Miettinen. *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media, 2012.
- [54] Kaisa Miettinen and Marko M. Mäkelä. Interactive multiobjective optimization system WWW-NIMBUS on the internet. *Computers & Operations Research*, 27(7-8):709–723, June 2000.
- [55] Seshahayee Murthy, Rama Akkiraju, John Rachlin, and Frederick Wu. Agent-based cooperative scheduling. In *Proceedings of AAAI Workshop on Constraints and Agents*, pages 112–117, 1997.
- [56] Hirotaka Nakayama, Yeboon Yun, and Min Yoon. Sequential approximate multiobjective optimization using computational intelligence. 05 2009.
- [57] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Science & Business Media, 2006.
- [58] Shigeru Obayashi and Daisuke Sasaki. Visualization and data mining of pareto solutions using self-organizing map. 2003.
- [59] A Osyczka. Multicriterion optimization in engineering with fortran programs. *Ellis Harward Lim. Publ., Chichester*, 1984.
- [60] V. Pareto. Cours d’économie politique. I and II, nov 1896.
- [61] Miriam Pescador-Rojas, Raquel Hernández Gómez, Elizabeth Montero, Nicolás Rojas-Morales, María-Cristina Riff, and Carlos A. Coello Coello. An overview of weighted and unconstrained scalarizing functions. In *Evolutionary Multi-Criterion Optimization*, pages 499–513, Cham, 2017. Springer International Publishing.
- [62] Ted K. Ralphs, Matthew J. Saltzman, and Margaret M. Wiecek. An improved algorithm for solving biobjective integer programs. *Annals of Operations Research*, 147(1):43–70, Oct 2006.
- [63] Robert G Reynolds. An introduction to cultural algorithms. In *Proceedings of the third annual conference on evolutionary programming*, pages 131–139. World Scientific, 1994.

- 
- [64] M. Pescador Rojas. *Adaptation Techniques for Scalarizing Functions used in Decomposition-Based Multi-Objective Evolutionary Algorithms*. PhD thesis, CINVESTAV-IPN, 2019.
- [65] F Sibel Salman, Jayant R Kalagnanam, Sesh Murthy, and Andrew Davenport. Cooperative strategies for solving the bicriteria sparse multiple knapsack problem. *Journal of heuristics*, 8(2):215–239, 2002.
- [66] Hiroyuki Sato, Hernán E. Aguirre, and Kiyoshi Tanaka. Controlling dominance area of solutions and its impact on the performance of moeas. In *Evolutionary Multi-Criterion Optimization*, pages 5–20, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
- [67] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009.
- [68] Karthik Sindhya, Ana Belen Ruiz, and Kaisa Miettinen. A preference based interactive evolutionary algorithm for multi-objective optimization: Pie. In Ricardo H. C. Takahashi, Kalyanmoy Deb, Elizabeth F. Wanner, and Salvatore Greco, editors, *Evolutionary Multi-Criterion Optimization*, pages 212–225, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
- [69] André Süßflow, Nicole Drechsler, and Rolf Drechsler. Robust multi-objective optimization in high dimensional spaces. In *Evolutionary Multi-Criterion Optimization*, pages 715–726, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
- [70] Yan-Yan Tan, Yong-Chang Jiao, Hong Li, and Xin kuan Wang. MOEA/D+ uniform design: A new version of MOEA/D for optimization problems with many objectives. *Computers & Operations Research*, 40(6):1648–1660, June 2013.
- [71] Kalyan Veeramachaneni, Ignacio Arnaldo, Owen Derby, and Una-May O’Reilly. Flexgp. *Journal of Grid Computing*, 13(3):391–407, Sep 2015.
- [72] Z. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao. Adaptive replacement strategies for MOEA/D. *IEEE Transactions on Cybernetics*, 46(2):474–486, Feb 2016.
- [73] D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, apr 1997.
-

- [74] Q. Zhang and H. Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, dec 2007.
- [75] Q. Zhang, W. Liu, and H. Li. The performance of a new version of MOEA/D on CEC09 Unconstrained MOP Test Instances. In *2009 IEEE Congress on Evolutionary Computation*, pages 203–208, May 2009.
- [76] Eckart Zitzler. *Evolutionary algorithms for multiobjective optimization: methods and applications*. PhD thesis, Swiss Federal Institute of Technology (ETH), 1999.
- [77] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, June 2000.