



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

Unidad Zacatenco

Departamento de Computación

**iMOACO_ℝ: un nuevo algoritmo de optimización
multi-objetivo basado en colonias de hormigas
para espacios de búsqueda continuos**

T E S I S

Que presenta

Jesús Guillermo Falcón Cardona

Para obtener el grado de

Maestro en Ciencias en Computación

Director de la tesis:

Dr. Carlos Artemio Coello Coello

Ciudad de México

Septiembre, 2016

Resumen

En los ámbitos ingenieril, científico e industrial existen problemas que requieren de la optimización simultánea de varios objetivos que se encuentran normalmente en conflicto entre sí. Éstos son los llamados problemas de optimización multi-objetivo (POMs). Debido al conflicto entre los objetivos, existen varias soluciones que satisfacen un POM en lugar de tener una única solución global como en el caso de un problema de optimización mono-objetivo. Estas soluciones conforman el denominado «conjunto óptimo de Pareto», mientras que su imagen es denominada «frente óptimo de Pareto».

Las técnicas de programación matemática surgieron como un medio de solución de POMs. Sin embargo, bajo ciertas circunstancias, éstas presentan varios inconvenientes que deterioran su desempeño. En consecuencia, el uso de algoritmos evolutivos y otras metaheurísticas bioinspiradas para resolver POMs ha aumentado de manera significativa en los últimos 20 años. Esto ha dado pie al surgimiento de una gran variedad de algoritmos evolutivos multi-objetivo (AEMOs) en la literatura especializada.

Por otra parte, la optimización mediante colonias de hormigas (OCH) es una metaheurística inspirada por el comportamiento social de las hormigas en su búsqueda de alimento. Las ideas principales de OCH son la auto-organización y la comunicación indirecta basada en modificaciones al ambiente empleando una sustancia denominada feromona. A través de la feromona, una hormiga puede sesgar probabilísticamente las decisiones de otras hormigas. OCH fue diseñado inicialmente para resolver problemas de optimización combinatoria (como el problema del viajero) debido a que se descubrió que los trazos de feromona convergían al camino más corto entre el nido y la fuente de alimento. Posteriormente, OCH fue extendido para la resolución de problemas de optimización continua dando buenos resultados y siendo $ACO_{\mathbb{R}}$ una de las mejores propuestas de este tipo disponible en la actualidad. Sin embargo, el diseño de algoritmos de optimización multi-objetivo basados en OCH ha sido muy poco explorada en la literatura especializada.

En esta tesis se propone un nuevo algoritmo de optimización multi-objetivo basado en OCH para la resolución de POMs en espacios de búsqueda continuos. La propuesta emplea $ACO_{\mathbb{R}}$ como su motor de búsqueda y es denominada *indicator-based Multi-Objective Ant Colony Optimization Algorithm for Continuous Search Spaces* ($iMOACO_{\mathbb{R}}$). Este algoritmo también es capaz de resolver POMs de alta dimensionalidad debido al uso de un esquema de selección basado en el indicador de desempeño R2. El enfoque propuesto es comparado con respecto a AEMOs representativos del estado del arte (NSGA-III, MOEA/D, SMS-EMOA y $MOACO_{\mathbb{R}}$) empleado problemas de prueba e indicadores de desempeño estándar de la literatura especializada. Los resultados experimentales indican que $iMOACO_{\mathbb{R}}$ es competitivo con respecto a NSGA-III y MOEA/D y supera a SMS-EMOA y $MOACO_{\mathbb{R}}$ en la mayoría de los problemas de prueba adoptados. Además, $iMOACO_{\mathbb{R}}$ presenta un mejor desempeño

conforme aumenta la dimensionalidad del problema. De esta forma, iMOACO_R parece ser un buen punto de partida para obtener un optimizador muti-objetivo basado en OCH altamente competitivo.

Abstract

Within engineering, science and industry, there are problems that require the simultaneous optimization of several objectives, which are normally in conflict with each other. These are the so-called multi-objective optimization problems (MOPs). Due to the conflict among the objectives, there are several solutions that satisfy a MOP, instead of having a single global solution as it happens when dealing with a single-objective optimization problem. These solutions constitute the so-called «Pareto optimal set», whereas the image of this set is called «Pareto optimal front».

Mathematical programming techniques were proposed as an alternative for solving MOPs. However, under several circumstances, such techniques have several drawbacks that deteriorate their performance. Consequently, the use of evolutionary algorithms and other bio-inspired metaheuristics for solving MOPs has significantly increased in the last 20 years. This has given rise to a wide variety of multi-objective evolutionary algorithms (MOEAs) in the specialized literature.

On the other hand, ant colony optimization (ACO) is a metaheuristic inspired on the social behavior of ants that are seeking for food. The core ideas of ACO are self-organization and indirect communication based on modifications to the environment using a substance called pheromone. Through the pheromone, an ant can probabilistically bias the decisions of other ants. ACO was initially designed to solve combinatorial optimization problems (such as the traveling salesperson problem) because it was discovered that pheromone trails converge to the shortest path between the nest and the food source. Later on, ACO was extended for the solution of continuous optimization problems, providing good results, being $ACO_{\mathbb{R}}$ one of the best currently available proposals. However, the design of multi-objective optimization algorithms based on ACO has been only scarcely explored in the specialized literature.

In this thesis, we propose a new multi-objective optimization algorithm based on ACO for solving MOPs in continuous search spaces. The proposed approach adopts $ACO_{\mathbb{R}}$ as its search engine and is called *indicator-based Multi-Objective Ant Colony Optimization Algorithm for Continuous Search Spaces* ($iMOACO_{\mathbb{R}}$). This approach is also able to solve high-dimensionality MOPs because it adopts a selection scheme based on the performance indicator R2. The proposed approach is compared with respect to MOEAs that are representative of the state-of-the-art (NSGA-III, MOEA/D, SMS-EMOA and $MOACO_{\mathbb{R}}$) using standard test problems and performance indicators taken from the specialized literature. Our experimental results indicate that $iMOACO_{\mathbb{R}}$ is competitive with respect to NSGA-III and MOEA/D and it outperforms SMS-EMOA and $MOACO_{\mathbb{R}}$ in most of the test problems adopted. Furthermore, $iMOACO_{\mathbb{R}}$ has a better performance as the dimensionality of a problem increases. This way, $iMOACO_{\mathbb{R}}$ seems to be a good departing point for producing a highly competitive multi-objective optimizer based on ACO.

Agradecimientos

Al CINESTAV y al CONACyT por las becas económicas que me otorgaron durante mis estudios de maestría. Sin estos apoyos no habría sido posible concluir satisfactoriamente esta etapa de mi formación académica. Este trabajo de tesis se derivó del proyecto CONACyT titulado “Nuevos Paradigmas Algorítmicos en Optimización Evolutiva Multi-Objetivo” (Ref. 221551), cuyo responsable es el Dr. Carlos A. Coello Coello.

En primer lugar, quiero agradecer a mis padres Blanca Estela Cardona y Guillermo Falcón por su apoyo incondicional a lo largo de todo el proceso de la maestría. Sin lugar a dudas, ellos han sido un gran soporte en muchos momentos muy difíciles que viví en estos dos años. Su amor ha sido vital.

Quisiera agradecer de forma especial al Dr. Carlos Coello por su guía y apoyo constante en el proyecto de tesis. Sus comentarios y observaciones han sido sumamente importantes para mejorar la tesis. Gracias por el apoyo para poder asistir al PPSN en Edimburgo. Su ejemplo como investigador es una fuente muy importante de inspiración en mi vida profesional.

A mis revisores de tesis el Dr. Luis Gerardo de la Fraga y el Dr. Gregorio Toscano, por su tiempo y valiosas aportaciones que me han permitido obtener un mejor trabajo. Asimismo, a cada uno de los doctores del departamento por sus enseñanzas en los cursos impartidos.

Mi especial reconocimiento a la Dra. Dolores Lara quien fue mi tutora durante el primer año de maestría. Siempre su apoyo y atención hacia mis estudios fueron invaluable, puesto que gracias a su dirección pude obtener una base sólida en las ciencias de la computación.

A Brittany Santomé quien me apoyó por largo tiempo de la maestría. Sin aquellas veces en las que sólo me escuchaba hablar de mis ideas para mejorar el algoritmo, no hubiese surgido la idea final que dió paso a iMOACO_ℝ. Muchas gracias.

Un profundo reconocimiento a mi amigo y hermano Julio Cesar Gómez. Tú siempre has estado conmigo en los buenos y malos momentos. Sin ti todo este tiempo habría sido más difícil. Muchas gracias hermano.

También quiero a agradecer a mis amigos de la maestría Diana, Manuel, Luis, y Erik. Sin ustedes los días en Computación habrían sido aburridos. Disfruté cada momento con ustedes desde estar estudiando todos los días (incluyendo sábados y domingos) para algoritmos u otras materias, como simplemente reír y hablar de tantas

cosas. Gracias amigos. A Raquel Hernández por toda la amabilidad y apoyo brindado durante el desarrollo de la tesis, ya que su trabajo fue de gran inspiración para mí. A Edgar y Luis Miguel Antonio por su ayuda y consejos en algunos aspectos de la tesis.

Índice general

Índice de figuras	XI
Índice de tablas	XIX
1. Introducción	1
1.1. Antecedentes y motivación	2
1.2. Planteamiento del problema y propuesta de solución	3
1.3. Objetivo general	4
1.4. Contribuciones	4
1.5. Trabajo previo	5
1.6. Estructura de la tesis	7
2. Conceptos de optimización multi-objetivo	9
2.1. Optimización multi-objetivo	10
2.2. Optimalidad de Pareto	11
2.3. Puntos de referencia	13
2.4. Tomador de decisiones	16
2.5. Características de los problemas multi-objetivo	16
2.5.1. Paisaje de aptitud	17
2.5.2. Geometrías del frente óptimo de Pareto	20
2.6. Métodos para resolver problemas multi-objetivo	22
2.7. Optimización de muchas funciones objetivo	27
2.8. Indicadores de calidad como mecanismos de selección	30
2.8.1. Propiedades	31
2.8.2. Hipervolumen	32
2.8.2.1. IBEA	33
2.8.2.2. SMS-EMOA	34
2.8.3. R2	34
2.8.3.1. MOMBI y MOMBI-II	35
2.8.3.2. R2-MOGA y R2-MODE	36
2.9. Resumen	36

3. Optimización mediante colonias de hormigas	39
3.1. Metáfora biológica	39
3.2. Descripción de la metaheurística	43
3.3. OCHs para optimización mono-objetivo	45
3.3.1. Ant System	45
3.3.2. Ant Colony System	47
3.3.3. Metáfora OCH para espacios de diseño continuos	49
3.3.4. ACO _{\mathbb{R}}	52
3.3.5. Aplicaciones en problemas mono-objetivo	55
3.4. OCHs para optimización multi-objetivo	55
3.4.1. CPACO	57
3.4.2. MO-ant	58
3.4.3. PACO-MOFO	58
3.4.4. MOACO _{\mathbb{R}}	59
3.5. Resumen	60
4. Metaheurística iMOACO_{\mathbb{R}}	63
4.1. Algoritmo clasificación-R2	63
4.2. Función de utilidad	65
4.3. Generación de vectores de pesos	66
4.4. Manejo de puntos de referencia	66
4.5. Construcción de iMOACO _{\mathbb{R}}	69
4.6. Resumen	71
5. Estudio experimental	73
5.1. Problemas de prueba	75
5.2. Evaluación de desempeño	75
5.3. Experimento 1: Comparación de desempeño	77
5.4. Experimento 2: Análisis de varianza	87
5.5. Análisis de debilidades	92
5.6. Resumen	94
6. Conclusiones y trabajo futuro	97
A. Problemas de prueba	99
A.1. Conjunto de problemas Zitzler-Deb-Thiele	99
A.2. Conjunto de problemas Deb-Thiele-Laumanns-Zitzler	104
A.3. Conjunto de problemas Walking-Fish-Group	114
A.4. Resumen	127
B. Resultados numéricos	129
B.1. Resultados del experimento 1	129
B.2. Resultados del experimento 2	157

ÍNDICE GENERAL

XI

Bibliografía

170

Índice de figuras

2.1. Ilustración del concepto de dominancia de Pareto.	12
2.2. Transformación del espacio de decisión al espacio objetivo. Los puntos en tono más oscuro en el espacio de decisión corresponden a \mathcal{P}^* y aquellos en el espacio objetivo representan \mathcal{PF}^*	13
2.3. Características deseables de una aproximación al frente de Pareto. . .	14
2.4. Puntos de referencia: vector ideal (z^*), vector de nadir (z^{nad}) y vector utópico (z^{**}).	15
2.5. Características del paisaje de aptitud para un POM [99].	17
2.6. (a) Región plana dentro de la función $f(x) = 0.7 + 1.75 \min(0, x - 0.4)(0.4 - x) - 0.75 \min(0, 0.6 - x)(x - 0.6)$, donde $x \in [0, 1]$. (b) Ejemplo de una función multimodal $f(x) = x^2 + \sin\frac{5}{3}\pi x - \frac{\pi}{2}$, donde $x \in [-3, 3]$	18
2.7. (a) Ejemplo de una función deceptiva $f(x) = x + x ^{0.001}$, donde $x \in [-0.5, 0.5]$. (b) Los vectores objetivo de 40,000 vectores de decisión seleccionados aleatoriamente. La función se define como sigue: $F(x, y) = (x^5 + y, x^5 + 1 - y)$, donde $x, y \in [0, 1]$. Se puede notar claramente el sesgo hacia la región del frente óptimo de Pareto.	19
2.8. Ejemplo del cierre convexo.	21
2.9. Ejemplo de un frente mixto y desconectado conteniendo un componente convexo, un componente cóncavo, un punto degenerado y un componente lineal.	22
2.10. Técnicas de búsqueda y optimización global.	23
2.11. (a) conjuntos de soluciones en un problema bi-dimensional, (b) conjuntos de soluciones en un problema tri-dimensional. Las buenas soluciones se denotan por ‘b’, las malas soluciones por ‘w’ y las soluciones equivalentes mediante ‘e’.	29
2.12. Ejemplos del hipervolumen dominado en dos y tres dimensiones. La figura de la izquierda muestra un caso de minimización, mientras que la de la derecha presenta un caso de maximización.	33
3.1. Configuración del experimento del doble puente. (a) Los brazos del puente tienen la misma longitud. (b) Los brazos tienen longitud diferente.	41

3.2. (a) Ejemplo de la discretización del espacio de búsqueda empleando vectores de dirección. (b) Modelo del vecindario de nido en un espacio bidimensional. Cada vector evoluciona con respecto al tiempo guiado por la aptitud de las soluciones halladas por las hormigas. El radio de búsqueda máximo es R	51
3.3. Ejemplo de cuatro FDPs gaussianas y su superposición para formar una FDP multimodal gaussiana.	53
3.4. Modelo de feromonas de $ACO_{\mathbb{R}}$. Las soluciones se encuentran ordenadas con respecto a la calidad de la función objetivo $u(s_1) \leq u(s_2) \leq \dots \leq u(s_k)$, tomando en cuenta un problema de minimización Además, cada solución i tiene asociado un peso w_i proporcional a su calidad. Por tanto, $w_1 \geq w_2 \geq \dots \geq w_k$. Cada variable de decisión se modela empleando una FDP-G.	53
4.1. Generación de vectores de pesos usando SLD con $h = 5$ y $m = 3$	67
5.1. Clasificación de desempeño de las metaheurísticas probadas en el conjunto ZDT con base en los indicadores H, IGD+ y S. Por cada algoritmo, se muestra el número de POMs en los que obtiene una clasificación dada.	80
5.2. Clasificación de desempeño de las metaheurísticas, por cada dimensión del espacio objetivo, probadas en el conjunto DTLZ con base en los indicadores H, IGD+ y S. Por cada algoritmo, se muestra el número de POMs en los que obtiene una clasificación dada.	81
5.3. Clasificación de desempeño de las metaheurísticas, por cada dimensión del espacio objetivo, probadas en el conjunto WFG con base en los indicadores H, IGD+ y S. Por cada algoritmo, se muestra el número de POMs en los que obtiene una clasificación dada.	82
5.4. Comparación de la cobertura y distribución de soluciones para el problema ZDT1 obtenida por $iMOACO_{\mathbb{R}}$ con respecto a $MOACO_{\mathbb{R}}$, MOEA/D, NSGA-III y SMS-EMOA. La aproximación corresponde con la ejecución en la mediana del valor de hipervolumen.	84
5.5. Comparación de la cobertura y distribución de soluciones para el problema DTLZ2 en tres dimensiones obtenidas por $iMOACO_{\mathbb{R}}$ con respecto a $MOACO_{\mathbb{R}}$, MOEA/D, NSGA-III y SMS-EMOA. La aproximación corresponde con la ejecución en la mediana del valor de hipervolumen.	85
5.6. Comparación de la cobertura y distribución de soluciones para el problema WFG6 en tres dimensiones obtenidas por $iMOACO_{\mathbb{R}}$ con respecto a $MOACO_{\mathbb{R}}$, MOEA/D, NSGA-III y SMS-EMOA. La aproximación corresponde con la ejecución en la mediana del valor de hipervolumen.	86
5.7. Datos de estadística descriptiva para el problema WFG1 empleando $iMOACO_{\mathbb{R}}$ con las diferentes combinaciones.	89

5.8. Intervalos de confianza generados por la prueba Tukey en el problema WFG1. Aquellos intervalos que intersectan el valor cero, indican que no hay una diferencia estadísticamente significativa entre las combinaciones pareadas.	90
5.9. Para los problemas ZDT4, DTLZ1 y DTLZ3, se muestra el espacio de las variables de decisión correspondiente al conjunto óptimo de Pareto y para la aproximación al conjunto de Pareto generada por iMOACO _ℝ	93
A.1. Frente óptimo de Pareto para ZDT1.	101
A.2. Frente óptimo de Pareto para ZDT2.	102
A.3. Frente óptimo de Pareto para ZDT3.	103
A.4. Frente óptimo de Pareto para ZDT4.	104
A.5. Frente óptimo de Pareto para ZDT6.	105
A.6. Frente óptimo de Pareto para DTLZ1 con tres objetivos.	107
A.7. Frente óptimo de Pareto para DTLZ2 con tres objetivos.	108
A.8. Frente óptimo de Pareto para DTLZ3 con tres objetivos.	109
A.9. Frente óptimo de Pareto para DTLZ4 con tres objetivos.	110
A.10. Frente óptimo de Pareto para DTLZ5 con tres objetivos.	112
A.11. Frente óptimo de Pareto para DTLZ6 con tres objetivos.	113
A.12. Frente óptimo de Pareto para DTLZ7 con tres objetivos.	114
A.13. Frente óptimo de Pareto para WFG1 con tres objetivos.	116
A.14. Frente óptimo de Pareto para WFG2 con tres objetivos.	117
A.15. Frente óptimo de Pareto para WFG3 con tres objetivos.	119
A.16. Frente óptimo de Pareto para WFG4 con tres objetivos.	120
A.17. Frente óptimo de Pareto para WFG5 con tres objetivos.	121
A.18. Frente óptimo de Pareto para WFG6 con tres objetivos.	122
A.19. Frente óptimo de Pareto para WFG7 con tres objetivos.	123
A.20. Frente óptimo de Pareto para WFG8 con tres objetivos.	124
A.21. Frente óptimo de Pareto para WFG9 con tres objetivos.	125
B.1. Gráficas de las aproximaciones obtenidas por iMOACO _ℝ para el conjunto de problemas ZDT, excepto ZDT4 donde no hay convergencia. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes.	131
B.2. Gráficas de las aproximaciones obtenidas por iMOACO _ℝ para el problema DTLZ2 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes.	136
B.3. Gráfica de la aproximación obtenida por iMOACO _ℝ para DTLZ2 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen.	136
B.4. Gráficas de las aproximaciones obtenidas por iMOACO _ℝ para el problema DTLZ4 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes.	137

B.5. Gráfica de la aproximación obtenida por $iMOACO_{\mathbb{R}}$ para DTLZ4 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen.	137
B.6. Gráficas de las aproximaciones obtenidas por $iMOACO_{\mathbb{R}}$ para el problema DTLZ5 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes.	138
B.7. Gráfica de la aproximación obtenida por $iMOACO_{\mathbb{R}}$ para DTLZ5 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen.	138
B.8. Gráficas de las aproximaciones obtenidas por $iMOACO_{\mathbb{R}}$ para el problema DTLZ6 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes.	139
B.9. Gráfica de la aproximación obtenida por $iMOACO_{\mathbb{R}}$ para DTLZ6 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen.	139
B.10. Gráficas de las aproximaciones obtenidas por $iMOACO_{\mathbb{R}}$ para el problema DTLZ7 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes.	140
B.11. Gráfica de la aproximación obtenida por $iMOACO_{\mathbb{R}}$ para DTLZ7 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen.	140
B.12. Gráficas de las aproximaciones obtenidas por $iMOACO_{\mathbb{R}}$ para el problema WFG1 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes.	148
B.13. Gráfica de la aproximación obtenida por $iMOACO_{\mathbb{R}}$ para WFG1 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen.	148
B.14. Gráficas de las aproximaciones obtenidas por $iMOACO_{\mathbb{R}}$ para el problema WFG2 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes.	149
B.15. Gráfica de la aproximación obtenida por $iMOACO_{\mathbb{R}}$ para WFG2 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen.	149
B.16. Gráficas de las aproximaciones obtenidas por $iMOACO_{\mathbb{R}}$ para el problema WFG3 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes.	150
B.17. Gráfica de la aproximación obtenida por $iMOACO_{\mathbb{R}}$ para WFG3 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen.	150
B.18. Gráficas de las aproximaciones obtenidas por $iMOACO_{\mathbb{R}}$ para el problema WFG4 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes.	151

B.19. Gráfica de la aproximación obtenida por $iMOACO_{\mathbb{R}}$ para WFG4 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen. 151

B.20. Gráficas de las aproximaciones obtenidas por $iMOACO_{\mathbb{R}}$ para el problema WFG5 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes. 152

B.21. Gráfica de la aproximación obtenida por $iMOACO_{\mathbb{R}}$ para WFG5 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen. 152

B.22. Gráficas de las aproximaciones obtenidas por $iMOACO_{\mathbb{R}}$ para el problema WFG6 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes. 153

B.23. Gráfica de la aproximación obtenida por $iMOACO_{\mathbb{R}}$ para WFG6 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen. 153

B.24. Gráficas de las aproximaciones obtenidas por $iMOACO_{\mathbb{R}}$ para el problema WFG7 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes. 154

B.25. Gráfica de la aproximación obtenida por $iMOACO_{\mathbb{R}}$ para WFG7 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen. 154

B.26. Gráficas de las aproximaciones obtenidas por $iMOACO_{\mathbb{R}}$ para el problema WFG8 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes. 155

B.27. Gráfica de la aproximación obtenida por $iMOACO_{\mathbb{R}}$ para WFG8 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen. 155

B.28. Gráficas de las aproximaciones obtenidas por $iMOACO_{\mathbb{R}}$ para el problema WFG9 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes. 156

B.29. Gráfica de la aproximación obtenida por $iMOACO_{\mathbb{R}}$ para WFG9 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen. 156

B.30. Datos de estadística descriptiva para el problema ZDT2 empleando $iMOACO_{\mathbb{R}}$ con diferentes combinaciones de parámetros. 157

B.31. Intervalos de confianza generados por la prueba Tukey en el problema ZDT2. Aquellos intervalos que intersectan el valor cero indican que no hay una diferencia estadísticamente significativa entre las combinaciones pareadas. 158

B.32. Datos de estadística descriptiva para el problema ZDT6 empleando $iMOACO_{\mathbb{R}}$ con diferentes combinaciones de parámetros. 158

B.34. Datos de estadística descriptiva para el problema DTLZ2 en tres dimensiones empleando iMOACO _ℝ con diferentes combinaciones de parámetros.	159
B.33. Intervalos de confianza generados por la prueba Tukey en el problema ZDT6. Aquellos intervalos que intersectan el valor cero indican que no hay una diferencia estadísticamente significativa entre las combinaciones pareadas.	159
B.35. Intervalos de confianza generados por la prueba Tukey en el problema DTLZ2 en tres dimensiones. Aquellos intervalos que intersectan el valor cero indican que no hay una diferencia estadísticamente significativa entre las combinaciones pareadas.	160
B.36. Datos de estadística descriptiva para el problema DTLZ5 en tres dimensiones empleando iMOACO _ℝ con diferentes combinaciones de parámetros.	160
B.37. Datos de estadística descriptiva para el problema DTLZ7 en tres dimensiones empleando iMOACO _ℝ con diferentes combinaciones de parámetros.	161
B.38. Intervalos de confianza generados por la prueba Tukey en el problema DTLZ7 en tres dimensiones. Aquellos intervalos que intersectan el valor cero indican que no hay una diferencia estadísticamente significativa entre las combinaciones pareadas.	161
B.39. Datos de estadística descriptiva para el problema WFG2 en tres dimensiones empleando iMOACO _ℝ con diferentes combinaciones de parámetros.	162
B.40. Intervalos de confianza generados por la prueba Tukey en el problema WFG2 en tres dimensiones. Aquellos intervalos que intersectan el valor cero indican que no hay una diferencia estadísticamente significativa entre las combinaciones pareadas.	162
B.41. Datos de estadística descriptiva para el problema WFG3 en tres dimensiones empleando iMOACO _ℝ con diferentes combinaciones de parámetros.	163
B.42. Intervalos de confianza generados por la prueba Tukey en el problema WFG3 en tres dimensiones. Aquellos intervalos que intersectan el valor cero indican que no hay una diferencia estadísticamente significativa entre las combinaciones pareadas.	163
B.43. Datos de estadística descriptiva para el problema WFG4 en tres dimensiones empleando iMOACO _ℝ con diferentes combinaciones de parámetros.	164
B.44. Intervalos de confianza generados por la prueba Tukey en el problema WFG4 en tres dimensiones. Aquellos intervalos que intersectan el valor cero indican que no hay una diferencia estadísticamente significativa entre las combinaciones pareadas.	164

B.45. Datos de estadística descriptiva para el problema WFG5 en tres dimensiones empleando iMOACO_R con diferentes combinaciones de parámetros. 165

B.46. Intervalos de confianza generados por la prueba Tukey en el problema WFG5 en tres dimensiones. Aquellos intervalos que intersectan el valor cero indican que no hay una diferencia estadísticamente significativa entre las combinaciones pareadas. 165

B.47. Datos de estadística descriptiva para el problema WFG6 en tres dimensiones empleando iMOACO_R con diferentes combinaciones de parámetros. 166

B.48. Intervalos de confianza generados por la prueba Tukey en el problema WFG6 en tres dimensiones. Aquellos intervalos que intersectan el valor cero indican que no hay una diferencia estadísticamente significativa entre las combinaciones pareadas. 166

B.49. Datos de estadística descriptiva para el problema WFG7 en tres dimensiones empleando iMOACO_R con diferentes combinaciones de parámetros. 167

B.50. Intervalos de confianza generados por la prueba Tukey en el problema WFG7 en tres dimensiones. Aquellos intervalos que intersectan el valor cero indican que no hay una diferencia estadísticamente significativa entre las combinaciones pareadas. 167

B.51. Datos de estadística descriptiva para el problema WFG8 en tres dimensiones empleando iMOACO_R con diferentes combinaciones de parámetros. 168

B.52. Intervalos de confianza generados por la prueba Tukey en el problema WFG8 en tres dimensiones. Aquellos intervalos que intersectan el valor cero indican que no hay una diferencia estadísticamente significativa entre las combinaciones pareadas. 168

B.53. Datos de estadística descriptiva para el problema WFG9 en tres dimensiones empleando iMOACO_R con diferentes combinaciones de parámetros. 169

B.54. Intervalos de confianza generados por la prueba Tukey en el problema WFG9 en tres dimensiones. Aquellos intervalos que intersectan el valor cero indican que no hay una diferencia estadísticamente significativa entre las combinaciones pareadas. 169

Índice de tablas

3.1. Conjunto de aplicaciones de OCH listadas de acuerdo con el tipo de problema. Tabla extraída de la obra de Dorigo y Stützle [62].	56
5.1. Características de los conjuntos de problemas de prueba ZDT, DTLZ y WFG.	74
5.2. Configuración adoptada para el conjunto de problemas de prueba WFG.	75
5.3. Puntos de referencia empleados en el cálculo del hipervolumen.	76
5.4. Parámetros comunes entre las metaheurísticas.	79
5.5. Combinaciones resultantes de los niveles establecidos para los parámetros q y ξ	88
5.6. Valor medio del hipervolumen para el problema DTLZ2 resuelto por iMOACO _R incrementando el número máximo de generaciones en múltiplos de 416. El mejor valor del hipervolumen para este problema corresponde al NSGA-III con 7.421715 (ver tabla B.2.)	92
5.7. Clasificación general de los optimizadores considerados en el experimento de comparación en cada conjunto de problemas de prueba por cada indicador de desempeño.	95
A.1. Resumen de características del conjunto de problemas de prueba ZDT.	100
A.2. Resumen de características del conjunto de problemas de prueba DTLZ. N/A significa «no aplica»	106
A.3. Resumen de características del conjunto de problemas de prueba WFG.	115
B.1. Comparación de iMOACO _R con respecto a SMS-EMOA, MOEA/D, NSGA-III y MOACO _R en los problemas ZDT con 2 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de iMOACO _R en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor.	130

- B.2. Comparación de iMOACO_ℝ con respecto a SMS-EMOA, MOEA/D, NSGA-III y MOACO_ℝ en los problemas DTLZ con 3 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de iMOACO_ℝ en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor. 131
- B.3. Comparación de iMOACO_ℝ con respecto a SMS-EMOA, MOEA/D, NSGA-III y MOACO_ℝ en los problemas DTLZ con 4 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de iMOACO_ℝ en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor. 132
- B.4. Comparación de iMOACO_ℝ con respecto a SMS-EMOA, MOEA/D, NSGA-III y MOACO_ℝ en los problemas DTLZ con 5 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de iMOACO_ℝ en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor. 132
- B.5. Comparación de iMOACO_ℝ con respecto a SMS-EMOA, MOEA/D, NSGA-III y MOACO_ℝ en los problemas DTLZ con 6 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de iMOACO_ℝ en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor. 133
- B.6. Comparación de iMOACO_ℝ con respecto a SMS-EMOA, MOEA/D, NSGA-III y MOACO_ℝ en los problemas DTLZ con 7 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de iMOACO_ℝ en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor. 133
- B.7. Comparación de iMOACO_ℝ con respecto a SMS-EMOA, MOEA/D, NSGA-III y MOACO_ℝ en los problemas DTLZ con 8 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de iMOACO_ℝ en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor. 134
- B.8. Comparación de iMOACO_ℝ con respecto a SMS-EMOA, MOEA/D, NSGA-III y MOACO_ℝ en los problemas DTLZ con 9 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de iMOACO_ℝ en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor. 134

B.9. Comparación de iMOACO_ℝ con respecto a SMS-EMOA, MOEA/D, NSGA-III y MOACO_ℝ en los problemas DTLZ con 10 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de iMOACO_ℝ en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor. 135

B.10. Comparación de iMOACO_ℝ con respecto a SMS-EMOA, MOEA/D, NSGA-III y MOACO_ℝ en los problemas WFG con 3 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de iMOACO_ℝ en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor. 141

B.11. Comparación de iMOACO_ℝ con respecto a SMS-EMOA, MOEA/D, NSGA-III y MOACO_ℝ en los problemas WFG con 4 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de iMOACO_ℝ en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor. 142

B.12. Comparación de iMOACO_ℝ con respecto a SMS-EMOA, MOEA/D, NSGA-III y MOACO_ℝ en los problemas WFG con 5 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de iMOACO_ℝ en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor. 143

B.13. Comparación de iMOACO_ℝ con respecto a SMS-EMOA, MOEA/D, NSGA-III y MOACO_ℝ en los problemas WFG con 6 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de iMOACO_ℝ en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor. 144

B.14. Comparación de iMOACO_ℝ con respecto a SMS-EMOA, MOEA/D, NSGA-III y MOACO_ℝ en los problemas WFG con 7 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de iMOACO_ℝ en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor. 145

B.15. Comparación de iMOACO_ℝ con respecto a SMS-EMOA, MOEA/D, NSGA-III y MOACO_ℝ en los problemas WFG con 8 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de iMOACO_ℝ en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor. 146

- B.16. Comparación de $iMOACO_{\mathbb{R}}$ con respecto a SMS-EMOA, MOEA/D, NSGA-III y $MOACO_{\mathbb{R}}$ en los problemas WFG con 9 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de $iMOACO_{\mathbb{R}}$ en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor. 146
- B.17. Comparación de $iMOACO_{\mathbb{R}}$ con respecto a SMS-EMOA, MOEA/D, NSGA-III y $MOACO_{\mathbb{R}}$ en los problemas WFG con 10 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de $iMOACO_{\mathbb{R}}$ en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor. 147

Capítulo 1

Introducción

El mundo real presenta una amplia gama de problemas en los que es necesario optimizar varios objetivos (normalmente en conflicto entre sí) simultáneamente. Para este tipo de problemas de optimización multi-objetivo (POMs) no hay una solución única sino que tienen varias soluciones que representan los mejores compromisos posibles entre los objetivos.

Debido a la gran complejidad de los POMs, se han propuesto a lo largo de los últimos años una serie de técnicas estocásticas denominadas metaheurísticas para su resolución. De entre esta clase de métodos, se encuentran los algoritmos de optimización evolutiva multi-objetivo (AEMOs), los cuales simulan el proceso evolutivo natural y han mostrado su aplicabilidad en diversos dominios durante los últimos 30 años [32].

Por otra parte, existen igualmente metaheurísticas basadas en el comportamiento social de aves, peces o insectos, englobadas en el amplio concepto de Inteligencia Colectiva. La interacción social de las hormigas ha dado origen a una técnica denominada optimización mediante colonias de hormigas (OCH) que ha demostrado su efectividad en problemas de optimización combinatoria y, en menor medida, en optimización en espacios de búsqueda continuos.

Se han propuesto muy pocos algoritmos de optimización multi-objetivo basados en OCH (AOMCHs) para espacios continuos en la literatura especializada, y los existentes presentan un desempeño que está por debajo del de un AEMO. En esta tesis, se desea proponer un AOMCH para espacios continuos que sea competitivo con respecto a AEMOs del estado del arte y, adicionalmente, que pueda resolver POMs en espacios de alta dimensionalidad en el espacio de las funciones objetivo.

El presente capítulo está organizado de la siguiente manera. La sección 1.1 presenta los antecedentes y motivación de este trabajo. La sección 1.2 establece el planteamiento del problema y la propuesta de solución. El objetivo general de la tesis se presenta en la sección 1.3. Las contribuciones principales de este trabajo se indican en la sección 1.4. La sección 1.5 describe brevemente el trabajo previo realizado. Finalmente, la organización de esta tesis se detalla en la sección 1.6.

1.1. Antecedentes y motivación

En años recientes, las técnicas de solución de POMs han tomado un papel esencial en escenarios de toma de decisiones en diversos planos incluyendo el científico, industrial y económico, entre otros. Las técnicas de programación matemática [143] fueron la primera vía de solución de este tipo de problemas. No obstante, estas técnicas producen en algunas ocasiones resultados muy pobres o simplemente fallan. Debido a esto, las metaheurísticas bioinspiradas [32] tales como los AEMOs, han surgido como una alternativa de resolución a este tipo de problemas extremadamente complejos.

El éxito de los AEMOs se ha hecho notar en la solución de POMs con dos o tres funciones objetivo, empleando principalmente como mecanismo de selección la dominancia de Pareto [82]. Desafortunadamente, cuando la dimensión del espacio de las funciones objetivo es mayor o igual a 4, este mecanismo de selección comienza a deteriorarse haciendo que los AEMOs tengan una convergencia muy pobre, que para 10 o más objetivos resulta peor que la de una selección puramente aleatoria [112, 118]. En consecuencia, estas dificultades han motivado a la comunidad científica especializada a desarrollar AEMOs que sean capaces de resolver POMs en altas dimensiones del espacio de las funciones objetivo.

OCH es una metaheurística bioinspirada para la resolución de problemas de optimización combinatoria mono-objetivo, propuesta por Dorigo [55] en su tesis doctoral. La fuente de inspiración biológica de OCH proviene del comportamiento social entre las hormigas de una colonia en busca de alimento. Cada hormiga de la colonia deposita feromonas en su camino del nido a la fuente de alimento y viceversa, dando como resultado la formación de trazos de feromona. Las hormigas son capaces de detectar las feromonas por lo que tienden a seguir aquellos caminos donde haya una mayor concentración de esta sustancia química. Dichos caminos son, en efecto, los más cortos hacia la fuente de alimento. Este comportamiento emergente entre la colonia se denomina *stigmergy*¹ donde las feromonas actúan como un medio indirecto de comunicación.

Inicialmente, OCH fue diseñado para resolver problemas combinatorios, siendo el problema del viajero (TSP, por sus siglas en inglés) su mayor exponente [60, 180]. Por otra parte, la aplicación de OCH hacia problemas de optimización en espacios de búsqueda continuos se ha extendido recientemente, habiendo una gran cantidad de propuestas [127]. Bilchev y Parmee [19] propusieron el primer OCH para dominios continuos y a partir de éste se han generado varias propuestas entre las que Leguizamón y Coello [127] mencionan las siguientes: [63, 31, 121, 131, 171]. Cabe resaltar que a pesar de la gran cantidad de OCHs para espacios continuos, aún no se cuenta con una propuesta que sea considerada como unificadora [127], aunque $ACO_{\mathbb{R}}$ [171] parece ser el mejor candidato para ello hasta el momento.

A diferencia del caso mono-objetivo, en la literatura especializada se han reportado muy pocos AOMCHs para POMs en espacios continuos [32, 127]. Leguizamón y Coello [127] mencionan únicamente dos algoritmos multi-objetivo de es-

¹Una forma de comunicación indirecta regulada por modificaciones en el ambiente.

te tipo: *Population-based ACO algorithm for Multi-Objective Function Optimization* (PACO-MOFO) [4] y el *Multi-Objective Ant Colony Optimizer for Continuous Spaces* (MOACO_ℝ) [77]. Ambos usan como motor de búsqueda ACO_ℝ [171] y emplean como criterio principal de selección de soluciones la dominancia de Pareto. A causa del uso de este esquema de selección, el funcionamiento de estos AOMCHs no es escalable en alta dimensionalidad del espacio objetivo, lo cual deja abierta la extensión de la metaheurística OCH para la solución de problemas de optimización con muchas funciones objetivo.

1.2. Planteamiento del problema y propuesta de solución

El desarrollo continuo de la ciencia ha motivado la creación de nuevas y mejores técnicas de resolución de problemas. En el área de las ciencias computacionales esto ha dado origen a diversos algoritmos determinísticos y estocásticos enfocados a resolver problemas altamente complejos como son los POMs. Entre este tipo de técnicas se encuentran los AEMOs y AOMCHs que se distinguen por ser métodos estocásticos. La directriz de esta tesis se orienta específicamente a la resolución de POMs (sin restricciones) extendiendo el motor de búsqueda ACO_ℝ a una versión multi-objetivo capaz de escalar a altas dimensiones en el espacio de las funciones objetivo.

Un problema de optimización multi-objetivo [32] se centra en la optimización (minimización o maximización) simultánea de m funciones objetivo. La solución a un POM, a diferencia de la de un problema de optimización mono-objetivo, no es única ya que involucra un conjunto de soluciones que presentan diferentes compromisos entre los objetivos. Estas soluciones son vectores de decisión que satisfacen un cierto conjunto de restricciones, en su caso general. Las funciones objetivo describen matemáticamente criterios de desempeño de los objetivos en conflicto entre sí. Cabe destacar que en la literatura especializada los POMs también son denominados problemas de optimización vectorial o multi-criterio. A continuación se presenta la definición formal de un POM sin restricciones.

$$\min_{x \in \Omega} F : \mathbb{R}^n \rightarrow \mathbb{R}^m \quad (1.1)$$

donde $\Omega \subset \mathbb{R}^n$, $F(x) = (f_1(x), \dots, f_k(x))^T$ y $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ es la i -ésima función objetivo.

Es imperativo enfatizar que para los fines de este trabajo se asumirá minimización de todas las funciones objetivo. En caso de requerir maximizar alguna de las funciones objetivo, éste se puede transformar a un problema de minimización usando: $\min f_i(x) \equiv -\max(-f_i(x))$.

En este trabajo se extiende la metaheurística OCH para resolver POMs empleando el motor de búsqueda ACO_ℝ [171] y usando un esquema de selección de soluciones basado en el indicador R2 [28], denominado clasificación-R2, propuesto por Hernández y Coello [100]. Gracias al uso de este esquema de selección, la propuesta

de solución presentada es capaz de escalar su funcionamiento en espacios objetivo de alta dimensionalidad. Aquí se propone un nuevo AOMCH para espacios de búsqueda continuos nombrado *indicator-based Multi-Objective Ant Colony Optimizer for Continuous Search Spaces* (iMOACO_ℝ), del cual se analiza su desempeño con respecto a tres AEMOs del estado del arte y un AOMCH para dominios continuos. En cuanto a los AEMOs se emplean los siguientes: NSGA-III [45], basado en dominancia de Pareto y puntos de referencia uniformemente distribuidos en el espacio objetivo; MOEA/D [191], basado en descomposición; SMS-EMOA [18], el cual emplea el indicador hipervolumen [192, 27] en su mecanismo de estimación de densidad (se utiliza el algoritmo HypE [10] para aproximar el valor del hipervolumen, puesto que su costo computacional es exponencial con respecto al número de objetivos). Con respecto al AOMCH, se utiliza MOACO_ℝ [77] que se basa en ACO_ℝ y dominancia de Pareto.

1.3. Objetivo general

Desarrollar un nuevo algoritmo de optimización multi-objetivo mediante la metaheurística colonias de hormigas (fundamentado en ACO_ℝ) para resolver problemas en espacios continuos basado en esquemas alternativos de selección con el fin de escalar su funcionamiento a altas dimensiones en el espacio de las funciones objetivo. El algoritmo propuesto deberá presentar un desempeño competitivo con respecto a algoritmos multi-objetivo del estado del arte, empleando conjuntos de problemas de prueba e indicadores de evaluación de calidad estándar de la literatura especializada.

1.4. Contribuciones

Las principales contribuciones de esta tesis se resumen a continuación:

- Un nuevo AOMCH para espacios de búsqueda continuos, basado en ACO_ℝ, capaz de resolver POMs en alta dimensionalidad en el espacio de las funciones objetivo.
- Un estudio comparativo detallado de la propuesta con respecto a otros AEMOs del estado del arte y un AOMCH para espacios continuos.
- Se presenta un análisis de sensibilidad de los parámetros específicos del motor de búsqueda ACO_ℝ para la resolución de POMs.
- Un artículo aceptado para el congreso internacional *Parallel Problem Solving from Nature* 2016 (PPSN'2016) [66].
- Un artículo en proceso de revisión para la revista internacional *Swarm Intelligence*.

1.5. Trabajo previo

En esta sección se revisa brevemente el motor de búsqueda $ACO_{\mathbb{R}}$, así como los AOMCHs existentes para espacios de búsqueda continuos y los principales esquemas de selección basados en indicadores.

Como se indicó anteriormente, $ACO_{\mathbb{R}}$ [171] es la propuesta que parece tener la mayor aceptación como trabajo unificador de OCH para espacios continuos [127]. Esta propuesta se basa en un modelado probabilístico de las zonas potenciales de búsqueda empleando funciones de densidad de probabilidad (FDPs) multimodales gaussianas. $ACO_{\mathbb{R}}$ usa una matriz de feromonas donde se almacenan las k mejores soluciones halladas hasta el momento. Por cada variable de decisión se genera una FDP con k funciones gaussianas empleando las soluciones en el modelo de feromonas. Las hormigas muestrean las FDPs para generar incrementalmente una solución. De acuerdo con los autores, $ACO_{\mathbb{R}}$ es la única propuesta que conserva las características distintivas de un OCH.

A pesar de que existen varias propuestas mono-objetivo de OCH para espacios continuos, no sucede lo mismo con las versiones multi-objetivo de OCH para espacios continuos [127]. Coello y Leguizamón [127] mencionan solamente dos propuestas multi-objetivo. En primera instancia, Angus formuló PACO-MOFO basándose en *Crowding Population-based Ant Colony Optimization* CPACO [3] y $ACO_{\mathbb{R}}$. PACO-MOFO emplea FDPs en una forma similar a $ACO_{\mathbb{R}}$. Además, utiliza un operador de reemplazo basado en un torneo de selección restringida [96] (RTS, por sus siglas en inglés) con el fin de mantener diversidad de soluciones y compartición de aptitud (*fitness sharing*) como mecanismo de selección orientado a asegurar un muestreo uniforme en el espacio de las funciones objetivo. No obstante, PACO-MOFO no pudo superar al *Nondominated Sorting Genetic Algorithm II* (NSGA-II) [44]. El segundo OCH multi-objetivo se denomina MOACO $_{\mathbb{R}}$ y fue propuesto por García-Najera y Bullinaria [77]. MOACO $_{\mathbb{R}}$ es una extensión directa de $ACO_{\mathbb{R}}$. Su diferencia fundamental yace en emplear el concepto de profundidad de dominancia [44] con el objeto de agrupar las soluciones en diferentes frentes, prefiriendo las más externas (*v.g.* las más cercanas al verdadero frente de Pareto). Emplea también distancia de agrupamiento (*crowding distance*) [32] como medio de admisión al archivo de soluciones. MOACO $_{\mathbb{R}}$ fue comparado contra NSGA-II [44] y el *Multi-Objective Particle Swarm Optimizer* (MOPSO) [34], superando ampliamente a este último pero no a NSGA-II. Ambas propuestas al hacer uso de la dominancia de Pareto como mecanismo primario de selección, no pueden resolver POMs con muchas funciones objetivo.

La optimización de muchas funciones objetivo (*many-objective optimization*) es un área reciente con gran auge debido a la ineficacia de los AEMOs basados en jerarquización de Pareto en altas dimensiones en el espacio de las funciones objetivo. Ishibuchi et al. [112] indican que los AEMOs basados en dominancia de Pareto sufren de deterioro en su funcionamiento al resolver POMs con espacios objetivo con cardinalidad mayor o igual a cuatro. Entre las principales dificultades se encuentran: (1) disminución de la presión de selección debido a la gran cantidad de soluciones no

dominadas, (2) incremento exponencial del número de soluciones para aproximar el frente de Pareto y (3) dificultad de visualización de las soluciones. Con el fin de escalar el funcionamiento de los AEMOs a altas dimensiones en el espacio de las funciones objetivo, Ishibuchi et al. [112] mencionan que se puede realizar lo siguiente: (1) modificar la dominancia de Pareto para aumentar la presión de selección, (2) introducir diferentes tipos de rangos para reducir el número de soluciones no dominadas y (3) uso de indicadores de calidad en el esquema de selección. Esta última es considerada probablemente como la propuesta más popular en la actualidad [112, 18].

En torno al paradigma que propone usar indicadores para formular esquemas de selección se encuentran los siguientes algoritmos. En 2004, Zitzler y Künzli [196] propusieron un algoritmo denominado *Indicator-based Evolutionary Algorithm* (IBEA) que tiene por objeto expresar, por medio de la inclusión de indicadores, diferentes preferencias del tomador de decisiones. IBEA transforma el problema de optimización del vector de funciones objetivo en la optimización de un indicador de calidad binario dado. IBEA demostró resultados competitivos con respecto a NSGA-II [44] y el *Strength Pareto Evolutionary Algorithm 2* (SPEA2) [198].

Por otra parte, un indicador ampliamente usado es el hipervolumen [192], del cual se ha demostrado matemáticamente que su maximización conduce a soluciones Pareto óptimas a lo largo del frente de Pareto, si bien éstas no son necesariamente uniformes. Además, es el único indicador unario que se sabe que presenta un comportamiento monótono. Basándose en este indicador, Beume et al. desarrollaron el *S Metric Selection Evolutionary Multi-Objective Algorithm* (SMS-EMOA) [18]. SMS-EMOA emplea el ordenamiento no dominado de NSGA-II [44] como forma de asignación de rangos a las soluciones y utiliza el hipervolumen como criterio de selección para descartar al individuo cuya contribución al hipervolumen sea mínima para el frente con el más alto rango. SMS-EMOA supera a algoritmos como el NSGA-II [44] o el SPEA2 [198]. Empero, una seria desventaja de este indicador es que su complejidad computacional es exponencial con respecto al número de objetivos, lo cual lo hace muy costoso en POMs en altas dimensiones del espacio objetivo.

Otro indicador ampliamente usado en la literatura especializada es R2 [28] que requiere una aproximación al verdadero frente de Pareto, un conjunto de funciones de utilidad y un conjunto de vectores de pesos convexos bien distribuidos. Con base en este indicador, Hernández-Gómez y Coello [100] propusieron la *Many-Objective Metaheuristic Based on the R2 Indicator* (MOMBI). La componente fundamental de MOMBI es la formulación de un esquema de asignación de rangos basado en R2 como base de su esquema de selección. Cabe resaltar que MOMBI no usa en lo absoluto la relación de dominancia de Pareto. Se propone el uso de la función ponderada de Tchebycheff [143] como función de utilidad. MOMBI fue comparado con respecto al *Multiobjective Evolutionary Algorithm based on Decomposition* (MOEA/D) [191] y SMS-EMOA [18]. Superó al primero en la mayoría de los casos y requirió menor costo computacional que el segundo. Ahora bien, los mismos autores han desarrollado MOMBI-II [101] cuya diferencia con respecto a MOMBI es la utilización de las funciones escalares por logros (ASF, por sus siglas en inglés) [143] como funciones de

utilidad y un esquema de actualización de los vectores de referencia basado en estadísticas de la población. De acuerdo con los autores, MOMBI-II supera a MOMBI, Δ_p -Differential Evolution (Δ_p -DDE) [159], NSGA-III [45], *R2-Many-Objective Genetic Algorithm* (R2-MOGA) [53] y R2-IBEA [181] en casi todos los problemas de prueba adoptados.

1.6. Estructura de la tesis

Además de este capítulo introductorio, esta tesis está constituida por seis capítulos y dos apéndices.

El capítulo 2 provee una serie de conceptos y definiciones que son necesarios como base teórica para el resto de la tesis.

El capítulo 3 introduce brevemente la optimización por colonias de hormigas aplicada tanto a espacios discretos como continuos.

La propuesta de solución se describe detalladamente en el capítulo 4.

El capítulo 5 muestra los resultados obtenidos al comparar la propuesta con respecto a tres AEMOs del estado del arte y un AOMCH para espacios continuos. Se emplean problemas de prueba estándar y medidas de evaluación de desempeño de la literatura especializada. Además, se estudia la sensibilidad de iMOACO _{\mathbb{R}} con respecto a dos de sus parámetros más importantes empleando el análisis de varianza (ANOVA, por sus siglas en inglés).

Las conclusiones obtenidas, así como también algunas posibles vertientes para trabajo futuro son presentadas en el capítulo 6.

El apéndice A contiene una descripción completa de los problemas de prueba estándar empleados para probar la metaheurística propuesta.

Finalmente, el apéndice B engloba los resultados numéricos producidos por los experimentos.

La implementación de la propuesta, así como también el estudio completo pueden ser descargados del sitio: <http://computacion.cs.cinvestav.mx/~jfalcon/imoacor.html>

Capítulo 2

Conceptos de optimización multi-objetivo

La aparición natural en diversas disciplinas de problemas con un múltiple número de objetivos a ser optimizados simultáneamente ha dado origen a una gran cantidad de investigación. Estos problemas son generalmente denominados problemas de optimización multi-objetivo (POMs). La optimización multi-objetivo, también llamada optimización multi-criterio, es la encargada de resolver los POMs.

Este capítulo tiene la intención de otorgar al lector una introducción a los términos básicos de la optimización multi-objetivo. El resto del capítulo está organizado de la siguiente manera. La sección 2.1 formula matemáticamente los problemas de optimización multi-objetivo. La sección 2.2 presenta la noción de óptimo de Pareto y de dominancia que son las más comúnmente usadas en espacios m dimensionales y, adicionalmente, se mencionan algunas características importantes que deben tener las soluciones generadas por optimizadores multi-objetivo. La sección 2.3 define algunos vectores especiales empleados usualmente como puntos de referencia. En la sección 2.4 se explica la función del tomador de decisiones en el proceso de solución. Las dificultades presentes en las funciones vectoriales que definen los problemas de optimización multi-objetivo así como también el tipo de geometrías asociadas a las hipersuperficies generadas por los optimizadores son introducidas en la sección 2.5. Existen diferentes técnicas que tienen por objeto resolver problemas de optimización multi-objetivo, los cuales se exponen brevemente en la sección 2.6. La sección 2.7 tiene como finalidad introducir una línea de investigación que ha tenido un gran impacto recientemente y tiene que ver con la escalabilidad de los optimizadores para resolver problemas multi-objetivo con más de tres funciones objetivo. Relacionado con el tema anterior, en la sección 2.8 se reseña el uso de los indicadores de calidad en mecanismos de selección de soluciones para ayudar a los optimizadores a escalar a altas dimensiones en el espacio de las funciones objetivo. Finalmente, en la sección 2.9 se resume el capítulo.

2.1. Optimización multi-objetivo

En una gran variedad de disciplinas se pueden encontrar de forma natural problemas que involucran la optimización simultánea de varias funciones objetivo que se encuentran en conflicto entre sí. Tradicionalmente, este tipo de problemas de optimización multi-objetivo (POMs) han sido manejados mediante una agregación (o suma) de los objetivos para así definir un problema de optimización mono-objetivo [143]. Sin embargo, esta forma de manejar un POM por lo general no otorga una buena visión del propio problema. En consecuencia, la optimización multi-objetivo (también llamada optimización multi-criterio u optimización vectorial) se presenta como un área de investigación reciente y cuyo avance va en aumento para la solución de problemas del mundo real.

De manera precisa, los POMs se definen como aquellos problemas donde la finalidad es optimizar simultáneamente m funciones objetivo que se encuentran en conflicto mutuamente. Esto puede involucrar la minimización o maximización de los m objetivos o una combinación de minimización y maximización. A continuación se define formalmente un MOP.

Definición 2.1.1 (Problema de optimización multi-objetivo) *Un problema de optimización multi-objetivo con restricciones puede ser escrito de la siguiente forma:*

$$\begin{aligned} \min_{\vec{x}} \quad & \vec{F} : \mathbb{R}^n \rightarrow \mathbb{R}^m \\ \text{tal que} \quad & g(\vec{x}) \leq 0 \\ & h(\vec{x}) = 0 \end{aligned} \tag{2.1}$$

donde $g : \mathbb{R}^n \rightarrow \mathbb{R}^q$, $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$, $\vec{F}(\vec{x}) = (f_1(\vec{x}), \dots, f_k(\vec{x}))^T$ y $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ es la i -ésima función objetivo.

$\vec{F} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ es nombrado «vector objetivo» y contiene $m(\geq 2)$ funciones objetivo f_i . El vector $\vec{x} = (x_1, x_2, \dots, x_n)^T$ se denomina «vector de decisión» y pertenece a una región factible $\Omega \subset \mathbb{R}^n$ definida por las funciones de restricción $g_i, h_j : \mathbb{R}^n \rightarrow \mathbb{R}$, para toda $i = 1, \dots, q, j = 1, \dots, p$. Estas funciones de restricción representan condicionamientos del ambiente o de recursos, propios del POM que se esté intentando resolver. Se debe enfatizar que el número de restricciones de igualdad p debe ser menor a n , porque en caso contrario ($p \geq n$) el POM estaría sobre-restringido, dado que habría más ecuaciones que incógnitas.

A diferencia de un problema de optimización mono-objetivo donde existe una única solución, un POM cuenta con un conjunto de soluciones que denotan los mejores compromisos entre los objetivos. Las soluciones de un POM son incomparables entre ellas, lo cual implica que todas ostentan la misma calidad y, muchas veces, son *incomensurables* (es decir, se encuentran medidas en diferentes unidades).

Por simplicidad, se presupone que todos los objetivos son minimizados. En caso de contar con problemas de maximización en el POM, éstos se pueden transformar a

problemas de minimización usando la siguiente identidad:

$$\text{máx } f_i(\vec{x}) := - \text{mín } (-f_i(\vec{x})) \quad (2.2)$$

La noción acerca de cuándo una solución es óptima es integral para la teoría y resolución de POMs. En la literatura se encuentran diferentes definiciones de optimalidad de soluciones [182]. Sin embargo, en la siguiente sección se introducirá el concepto de solución óptima más ampliamente utilizado en la literatura especializada.

2.2. Optimalidad de Pareto

Debido a que los POMs no tienen una única solución sino un conjunto de soluciones que mapean los diferentes compromisos entre los objetivos, una noción de orden que permita determinar la optimalidad de una solución debe ser establecida. Francis Ysidro Edgeworth [65] presentó en 1881 el concepto de óptimo en un espacio multidimensional, mientras que Vilfredo Pareto desarrolló, en mayor medida en 1896, el concepto a su forma actualmente aceptada [150, 149]. Debido a esto, la noción de óptimo empleada usualmente en POMs se denomina «óptimo de Pareto»¹. A continuación se presentan un conjunto de definiciones formales que apoyan el concepto de optimalidad de Pareto.

Definición 2.2.1 (Dominancia de Pareto) Una solución \vec{x} **domina** a una solución \vec{y} (denotado por $\vec{x} \prec \vec{y}$), si y sólo si \vec{x} es parcialmente menor que \vec{y} ; es decir, $\forall i \in \{1, \dots, m\}, x_i \leq y_i \wedge \exists j \in \{1, \dots, m\}$ tal que $x_j < y_j$.

Definición 2.2.2 (Dominancia débil de Pareto) Una solución \vec{x} **domina débilmente** a una solución \vec{y} (denotado por $\vec{x} \preceq \vec{y}$), si y sólo si $\forall i \in \{1, \dots, m\}, x_i \leq y_i$.

Definición 2.2.3 (Dominancia estricta de Pareto) Una solución \vec{x} **domina estrictamente** a una solución \vec{y} (denotado por $\vec{x} \prec_s \vec{y}$), si y sólo si $\forall i \in \{1, \dots, m\}, x_i < y_i$.

En la figura 2.1 se ejemplifican las definiciones de dominancia. Primeramente, la dominancia estricta se presenta en sólo un caso, $a \prec_s d$, puesto que se mejoran ambos objetivos simultáneamente. Por otra parte, la dominancia de Pareto se aprecia en los siguientes casos: $a \prec b$ y $a \prec c$ debido a que a mejora para al menos un índice j algún objetivo.

Definición 2.2.4 (Optimalidad de Pareto) Un vector de variables de decisión $\vec{x} \in \Omega$ es **Pareto-óptimo** si no existe otro vector de decisión $\vec{y} \in \Omega$ tal que $f_i(\vec{y}) \leq f_i(\vec{x})$ para todo $i = 1, \dots, m$ y $f_j(\vec{y}) < f_j(\vec{x})$ para al menos un índice j .

¹Inicialmente, se denominaba *óptimo de Edgeworth-Pareto*.

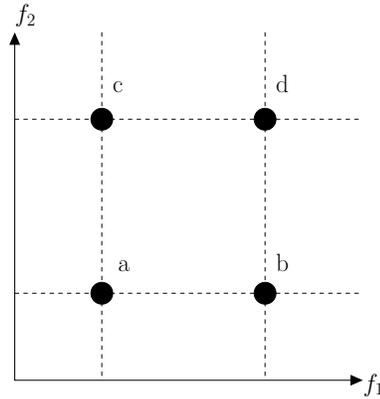


Figura 2.1: Ilustración del concepto de dominancia de Pareto.

Definición 2.2.5 (Conjunto óptimo de Pareto) *El conjunto óptimo de Pareto \mathcal{P}^* se define como:*

$$\mathcal{P}^* := \{x \in \Omega \mid \vec{x} \text{ es Pareto-óptimo}\}$$

Definición 2.2.6 (Frente óptimo de Pareto) *El frente óptimo de Pareto \mathcal{PF}^* está definido por:*

$$\mathcal{PF}^* := \{F(\vec{x}) \in \mathbb{R}^m \mid \vec{x} \in \mathcal{P}^*\}$$

Definición 2.2.7 (Optimalidad débil de Pareto) *Una solución $\vec{x}^* \in \Omega$ es un óptimo débil de Pareto si no existe otro $\vec{x} \in \Omega$ tal que $f_i(\vec{x}) < f_i(\vec{x}^*)$, para $i = 1, \dots, m$.*

Definición 2.2.8 (Optimalidad estricta de Pareto) *Un punto $\vec{x}^* \in \Omega$ se dice que es un óptimo estricto de Pareto si no existe ningún $\vec{x} \in \Omega$, $\vec{x} \neq \vec{x}^*$ tal que $f_i(\vec{x}) \leq f_i(\vec{x}^*)$, para $i = 1, \dots, m$.*

En otras palabras, la definición 2.2.4 indica que \vec{x} es Pareto-óptimo siempre y cuando no exista algún otro vector factible \vec{y} que produzca una mejora en algún objetivo sin haber causado simultáneamente un deterioro en al menos un criterio. Las soluciones Pareto-óptimas también son llamadas soluciones «no inferiores, admisibles» o «eficientes» [32].

Dado un POM, se desea encontrar todas las soluciones Pareto-óptimas que conforman el denominado conjunto de óptimos de Pareto (\mathcal{P}^*), que satisfagan la ecuación (2.1). La imagen de \mathcal{P}^* en el espacio objetivo es denominada frente óptimo de Pareto (\mathcal{PF}^*). En la figura 2.2, se presenta el conjunto de Pareto y su correspondiente frente de Pareto para un POM bi-objetivo.

La caracterización analítica de la variedad² geométrica que representa \mathcal{PF}^* es, en general, muy difícil de obtener y en la mayoría de los casos suele ser imposible hacerlo.

²Una variedad geométrica (en inglés denominado *manifold*) es una generalización del concepto de curva (1-variedad), superficie (2-variedad) a cualquier dimensión.

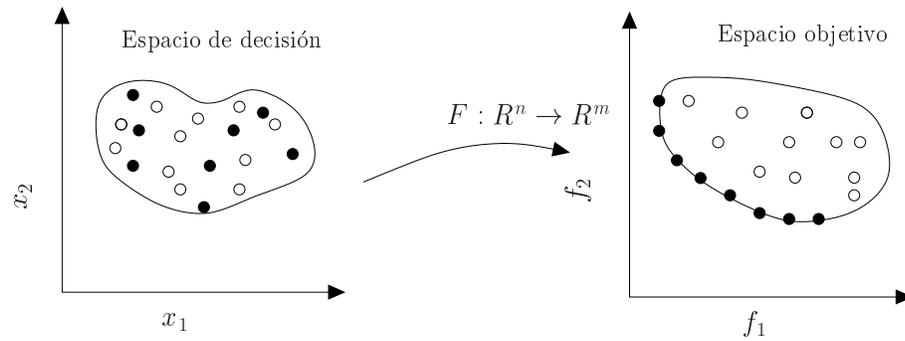


Figura 2.2: Transformación del espacio de decisión al espacio objetivo. Los puntos en tono más oscuro en el espacio de decisión corresponden a \mathcal{P}^* y aquellos en el espacio objetivo representan \mathcal{PF}^* .

Asimismo, otro problema involucrado en la obtención de \mathcal{PF}^* es que se compone de una infinidad de puntos, lo cual es imposible de procesar con una computadora digital. En consecuencia, una forma usual de obtener el frente de Pareto es mediante una aproximación finita, denotada mediante PF_{true} [182], que contenga un número suficiente de puntos que permitan describir \mathcal{PF}^* .

Con respecto a las aproximaciones al frente de Pareto, Zitzler et al. [195] han sugerido tres características deseables que deben ostentar estos conjuntos:

1. **Convergencia:** debe minimizarse la distancia de PF_{true} hacia \mathcal{PF}^* .
2. **Distribución:** es deseable contar con una buena (en muchos casos uniforme) distribución de las soluciones sobre el espacio objetivos.
3. **Cobertura:** las soluciones en PF_{true} deben tratar de maximizar el área cubierta en el espacio objetivo, es decir, por cada objetivo, debe cubrirse un amplio rango de valores por las soluciones en la aproximación.

Las características descritas previamente se muestran en la figura 2.3, donde la curva representa \mathcal{PF}^* y el conjunto de puntos simboliza PF_{true} . Primeramente, se tienen soluciones muy próximas a \mathcal{PF}^* pero totalmente dispersas. En segundo lugar, se tiene una aproximación no convergente, aunque bien distribuida y cubriendo mayormente \mathcal{PF}^* . En el tercer caso, se presentan soluciones convergentes y bien distribuidas, aunque se cubre sólo una pequeña región del frente. Finalmente, el cuarto caso ejemplifica las tres características que debe tener idealmente PF_{true} .

2.3. Puntos de referencia

En esta sección se definen tres puntos especiales en el espacio objetivo que algunos algoritmos de optimización multi-objetivo (AOMs) suelen emplear como soluciones de referencia. Tales vectores se muestran gráficamente en la figura 2.4.

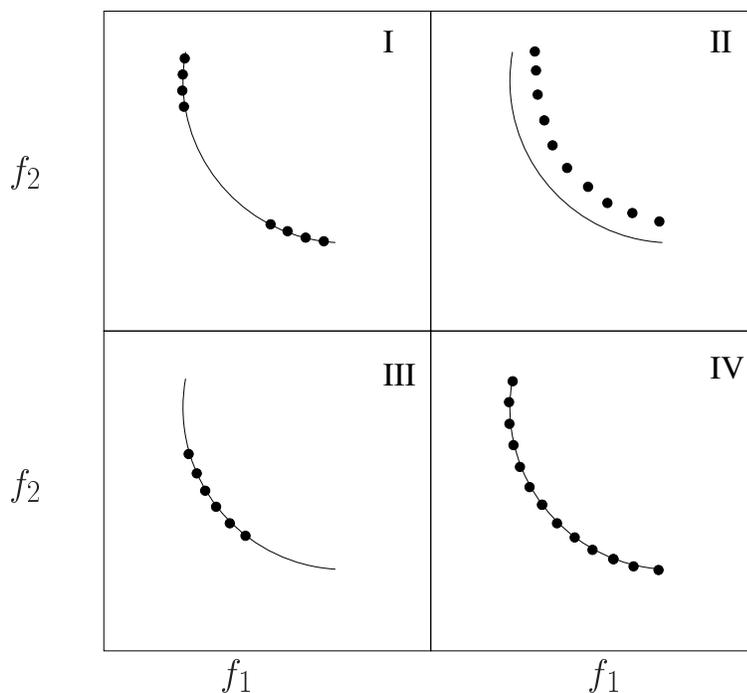


Figura 2.3: Características deseables de una aproximación al frente de Pareto.

El vector que contiene los valores que minimizan cada función objetivo considerada de forma independiente es llamado vector ideal (o perfecto) y se define como sigue:

Definición 2.3.1 (Vector objetivo ideal) *El vector objetivo ideal, denotado como $\bar{z}^* = (z_1^*, \dots, z_m^*)$ minimiza cada una de las funciones objetivo por separado. La i -ésima componente está dada por $z_i^* = \min_{\bar{x} \in \Omega} f_i(\bar{x})$.*

Es claro que si \bar{z}^* fuera factible, el frente de Pareto se vería reducido a este único punto. Sin embargo, esto normalmente no ocurre debido al conflicto entre los objetivos. Un aspecto a resaltar acerca de este punto, es que determina los límites inferiores de \mathcal{PF}^* por cada función objetivo.

Algunos AOMs requieren de un punto de referencia que sea estrictamente mejor, es decir, que domine a todas las soluciones Pareto-óptimas. Dado esto, se define a continuación el vector objetivo utópico.

Definición 2.3.2 (Vector objetivo utópico) *El vector objetivo utópico es un vector objetivo infactible cuyos componentes están formados por $\bar{z}^{**} = \bar{z}^* - \epsilon_i$ para todo $i \in \{1, \dots, m\}$, donde ϵ_i es un valor escalar positivo pequeño.*

Otro punto de referencia importante que ayuda a delimitar el frente de Pareto es el «vector objetivo de nadir», también llamado «vector objetivo imperfecto». Este vector es la contraparte de \bar{z}^* , ya que establece los límites superiores de \mathcal{PF}^* , es decir, los máximos valores por cada función objetivo. Aunado a esto, dicho vector puede ser factible o infactible.

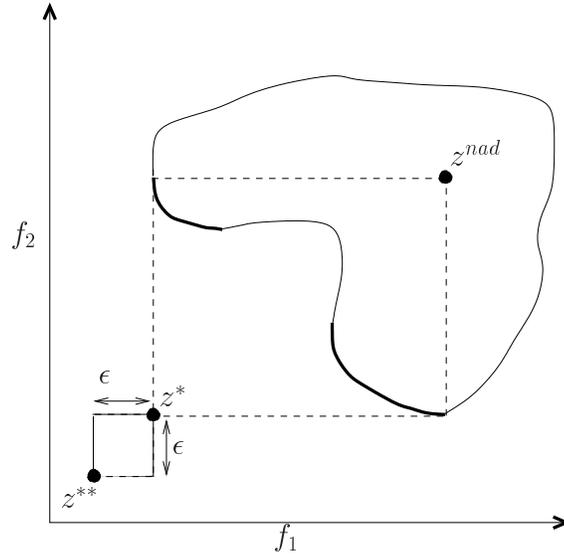


Figura 2.4: Puntos de referencia: vector ideal (z^*), vector de nadir (z^{nad}) y vector utópico (z^{**}).

Definición 2.3.3 (Vector objetivo de nadir) *El vector objetivo de nadir $\bar{z}^{nad} = (z_1^{nad}, \dots, z_m^{nad})^T$ es construido usando los peores valores de las funciones objetivo en \mathcal{PF}^* . Cada i -ésima componente está definida como $z_i^{nad} = \max_{\bar{x} \in \mathcal{P}^*} f_i(\bar{x})$.*

El vector de nadir usualmente es difícil de encontrar, particularmente para espacios objetivo de dimensión mayor que dos. Miettinen [143] menciona que las componentes del vector pueden ser aproximadas usando una tabla que es formada mediante el uso de los vectores objetivo obtenidos cuando se calcula \bar{z}^* . Para esto, la fila i de la tabla muestra los valores de todas las funciones objetivo calculadas en el punto en el que f_i obtuvo su valor mínimo. Por tanto, z_i^* está en la diagonal principal de la tabla. El valor máximo de la columna i en la tabla puede ser seleccionado como un estimado del límite superior para el objetivo f_i para $i = 1, \dots, m$ sobre el conjunto óptimo de Pareto.

Es necesario enfatizar que las estimaciones encontradas a partir del método descrito anteriormente no necesariamente son iguales a las componentes reales de \bar{z}^{nad} . Como resultado de esto, el valor de la aproximación del vector de nadir puede ser bajo o muy alto.

Finalmente, en algunas ocasiones es aconsejable hacer un ajuste de la escala de los valores objetivo, esto es, normalizar los objetivos para manejar aproximadamente una misma magnitud. En este caso, las componentes de \bar{z}^* y \bar{z}^{nad} son de gran valor para reemplazar cada función objetivo $f_i(\bar{x})$ ($i = 1, \dots, m$) por la función

$$f'_i(\bar{x}) = \frac{f_i(\bar{x}) - z_i^*}{z_i^{nad} - z_i^*} \quad (2.3)$$

donde el nuevo rango de cada función objetivo es $[0, 1]$.

2.4. Tomador de decisiones

En el sentido matemático, cada punto Pareto-óptimo es una solución igualmente aceptable puesto que entre ellos son incomparables [143]. No obstante, se suele elegir generalmente una única solución de entre todo el conjunto de Pareto que represente ciertas aspiraciones o metas. Para seleccionar una solución se involucran aspectos que no se encuentran presentes íntegramente en las funciones objetivo, sino que provienen de un conocimiento más específico del problema.

El responsable de seleccionar una solución es una persona (o grupo de personas) denominado «tomador de decisiones» (TD). El TD cuenta con tres características principales: (1) tiene un conocimiento profundo acerca del POM que se resuelve, (2) puede expresar preferencias sobre las soluciones existentes y (3) en él reside la responsabilidad por la decisión final tomada.

La resolución de un POM requiere de la cooperación del TD y un analista. Por *analista* se hace alusión a una persona o un programa de computadora responsable del procesamiento matemático para encontrar una solución. El analista procesa el POM y genera información que es enviada al TD de tal forma que pueda seleccionar una solución de acuerdo con sus preferencias.

Durante el proceso de solución, diferentes tipos de información son solicitados al TD. Esta información puede estar relacionada con, por ejemplo, niveles deseables o aceptables en los valores de las funciones objetivo. Estos valores (factibles o no) son de especial interés e importancia para el TD.

Definición 2.4.1 (Niveles de aspiración) *Los valores de las funciones objetivo que son satisfactorios o deseables para el TD son llamados **niveles de aspiración** y se denotan por $\bar{z}_i, i = 1, \dots, m$. El vector $\bar{z} \in \mathbb{R}^m$, que consiste de niveles de aspiración, es denominado un punto de referencia.*

Desde el punto de vista del TD, la solución a un POM involucra encontrar un vector de decisión factible tal que sea Pareto-óptimo y satisfaga sus necesidades y requerimientos. Asumiendo que tal solución existe, ésta es llamada «solución final». Sin embargo, en ocasiones puede ser difícil para el TD distinguir entre soluciones buenas y óptimas en problemas reales. Si este es el caso, se debe hacer énfasis en encontrar buenas soluciones (y algunas veces, sólo en encontrar soluciones).

2.5. Características de los problemas multi-objetivo

En la literatura especializada, la región de decisión factible Ω y la región objetivo factible \mathcal{Z} son conocidas como **espacio de búsqueda** y **espacio de aptitud**, respectivamente. Con base en esto, es claro que $\mathcal{P}^* \subset \Omega$ y $\mathcal{PF}^* \subset \mathcal{Z}$. La transformación del espacio de búsqueda al espacio de aptitud es llamado **paisaje de aptitud**.

En las siguientes dos subsecciones, se presentan las características del paisaje de aptitud y las posibles geometrías del frente óptimo de Pareto. La información a continuación presentada se basa principalmente en [104].

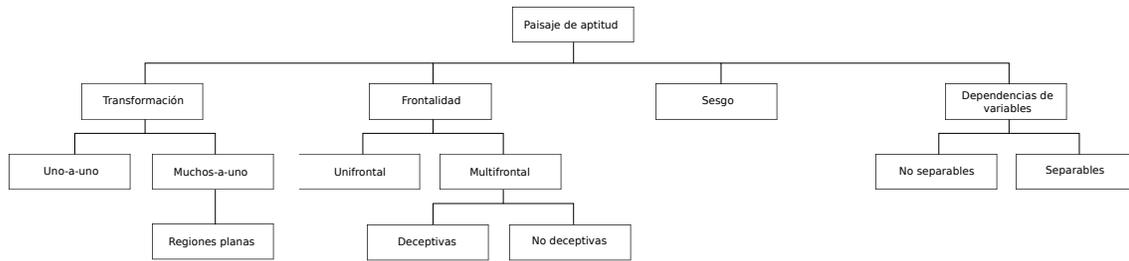


Figura 2.5: Características del paisaje de aptitud para un POM [99].

2.5.1. Paisaje de aptitud

En la figura 2.5 se presenta una clasificación del paisaje hecha por Hernández [99]. Puesto que tales características no son exclusivas, es posible que un POM pueda tener una combinación de ellas.

Dado que los paisajes de aptitud definen una transformación entre un dominio y un co-dominio, ésta puede ser **uno-a-uno** (inyectiva) o **muchos-a-uno** (sobreyectiva). Un paisaje de aptitud sobreyectivo da más dificultades para un optimizador ya que debe decidir varios vectores de decisión que se mapean a un mismo vector objetivo. Igualmente, la transformación entre el conjunto óptimo de Pareto y el frente óptimo de Pareto puede ser inyectivo o sobreyectivo. En estos casos, se dice que el POM es Pareto-inyectivo o Pareto-sobreyectivo.

Las **regiones planas** son un caso especial de una transformación sobreyectiva, donde un conjunto abierto del espacio de parámetros mapea a un conjunto unitario³, esto es, pequeñas perturbaciones en el espacio de decisión no cambian los valores objetivo. Los optimizadores pueden tener dificultades con las regiones planas debido a la falta de información del gradiente. En la figura 2.6(a) se ilustra esta característica para un problema con un solo objetivo.

Si la mayor parte del paisaje de aptitud presenta regiones planas, el optimizador no podrá obtener información de utilidad para localizar los óptimos de Pareto. En consecuencia, los óptimos de Pareto se dice que son **óptimos aislados** [41]. Los POMs con óptimos aislados son muy difíciles de resolver.

Otra característica importante de los paisajes de aptitud es la **frontalidad**. Una función objetivo \vec{F} es **multifrontal** si cuenta con varios frentes que se consideran localmente óptimos, mientras que cuando cuenta con un solo frente óptimo se denomina **unifrontal**. Un problema con una función objetivo multifrontal es llamado **problema multifrontal**. Este tipo de problemas pueden hacer que el optimizador quede atascado en alguno de los frentes localmente óptimos, sin opción a poder salir fácilmente. La frontalidad es la extensión de la modalidad en el caso mono-objetivo. En este ámbito, una función objetivo f es multimodal si cuenta con varias soluciones localmente óptimas y un óptimo global como se aprecia en la figura 2.6(b), mientras que si f sólo tiene el óptimo global, ésta es unimodal.

³Un conjunto unitario (*singleton* en inglés) es un conjunto que contiene un único elemento [176]

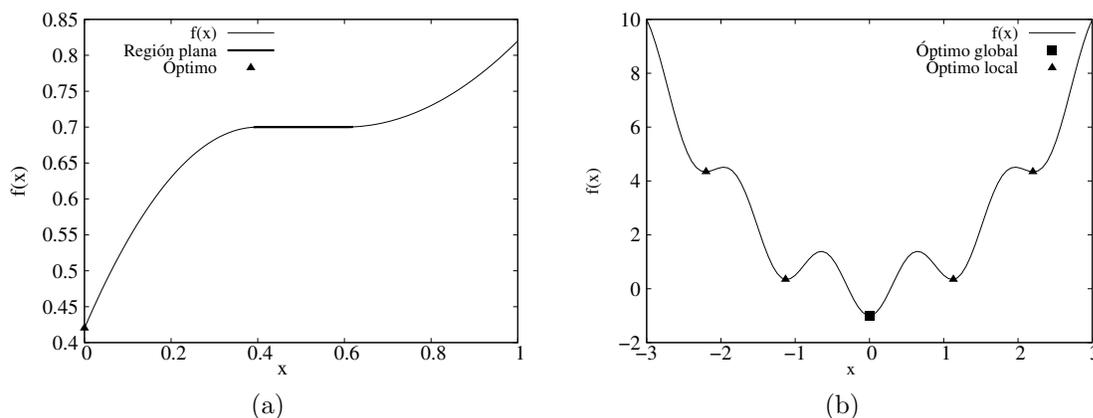


Figura 2.6: (a) Región plana dentro de la función $f(x) = 0.7 + 1.75 \min(0, x - 0.4)(0.4 - x) - 0.75 \min(0, 0.6 - x)(x - 0.6)$, donde $x \in [0, 1]$. (b) Ejemplo de una función multimodal $f(x) = x^2 + \sin\frac{5}{3}\pi x - \frac{\pi}{2}$, donde $x \in [-3, 3]$.

Las funciones objetivo **deceptivas** tienen un tipo especial de multifrentalidad. De acuerdo con Deb [41], para que una función objetivo sea deceptiva, debe tener al menos dos tipos de frentes: un frente óptimo verdadero y un frente deceptivo, aunque la mayoría del espacio de búsqueda debe favorecer al frente deceptivo. Un problema con esta característica es denominado **problema deceptivo**. Estos problemas explotan en mayor medida las dificultades de un problema multifrental al colocar el frente óptimo global en un lugar poco probable, no obstante que todas las regiones restantes conducen al frente deceptivo. Un ejemplo de este tipo de problemas es mostrado en la figura 2.7(a) para el caso mono-objetivo.

Otra característica del paisaje de aptitud es si una muestra uniformemente distribuida de vectores de parámetros en el espacio de búsqueda se asigna a un conjunto uniformemente distribuido de vectores objetivo en el espacio de aptitud. Se espera alguna variación, pero se está especialmente interesado en variaciones significativas, lo cual es conocido como **sesgo**. El sesgo tiene un impacto natural en el proceso de búsqueda, particularmente cuando la transformación del conjunto óptimo de Pareto al frente óptimo de Pareto está sesgada. Cuando sólo se puede identificar un subconjunto representativo del espacio distribuido y la transformación entre espacios presenta sesgo, el TD tiene que decidir usualmente entre tener una distribución uniforme de soluciones con respecto al espacio de búsqueda o al espacio de aptitud. En la figura 2.7(b) se pueden apreciar los efectos del sesgo.

Con el fin de juzgar si un problema está sesgado, se debe analizar la variación de densidad de las soluciones en el espacio de aptitud, dado un reparto equilibrado de soluciones en el espacio de decisión. Actualmente, no hay una definición matemática consensuada de sesgo (el lector puede revisar el trabajo de Da Fonseca [89] para obtener una posible definición). El sesgo tal vez es más claro observarlo mediante una

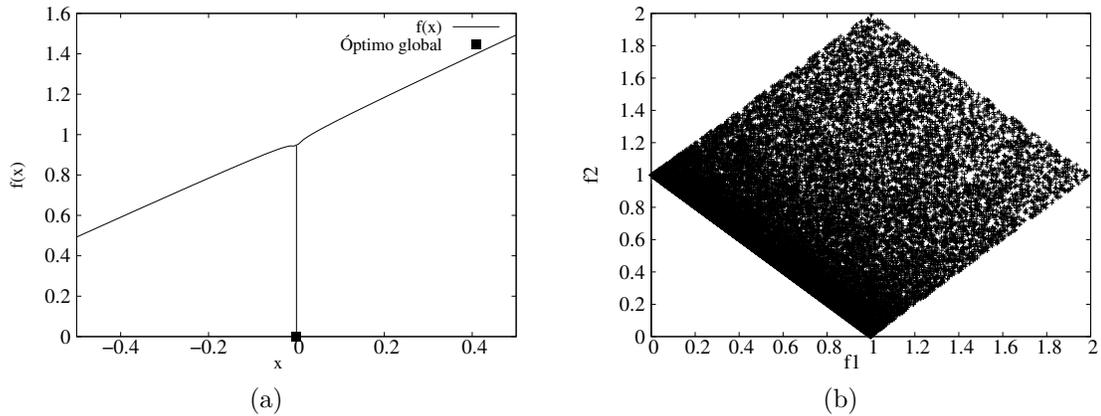


Figura 2.7: (a) Ejemplo de una función deceptiva $f(x) = x + |x|^{0.001}$, donde $x \in [-0.5, 0.5]$. (b) Los vectores objetivo de 40,000 vectores de decisión seleccionados aleatoriamente. La función se define como sigue: $F(x, y) = (x^5 + y, x^5 + 1 - y)$, donde $x, y \in [0, 1]$. Se puede notar claramente el sesgo hacia la región del frente óptimo de Pareto.

gráfica del espacio de aptitud del problema.

Las dependencias entre variables de decisión son un aspecto importante de un problema. Dado un único objetivo O , un vector de decisión \vec{x} , y un índice i , se define el problema derivado $P_{O, \vec{x}, i}$ como el problema de optimizar O mediante la variación única de x_i . Esto es un problema con un solo objetivo y un único parámetro. Se define también $P_{O, \vec{x}, i}^*$ como el conjunto global de los óptimos (en el espacio de decisión) para cada subproblema. Si $P_{O, \vec{x}, i}^*$ es el mismo para todos los valores de \vec{x} , se dice que x_i es **separable** en O . De otra forma, x_i es **no separable** en O .

Si la mayoría de los parámetros en O son separables, entonces O es un **objetivo separable**. De lo contrario, O es un **objetivo no separable**. Similarmente, si cada objetivo de un problema P es separable, entonces P es un **problema separable**. En caso contrario, P es un **problema no separable**.

Los objetivos separables pueden ser optimizados considerando cada variable de decisión en turno, independientemente de las demás. El conjunto resultante de vectores de decisión óptimos globalmente es el producto cruz de los conjuntos óptimos para cada parámetro optimizado individualmente. En el sentido multi-objetivo, esto significa que los puntos ideales para objetivos separables pueden ser determinados considerando sólo una variable de decisión a la vez. Un problema no separable es caracterizado por dependencias de variables lo cual le otorga mayor dificultad y lo hace más representativo del tipo de problemas que se presentan en el mundo real.

2.5.2. Geometrías del frente óptimo de Pareto

Debido a que los frentes óptimos de Pareto para problemas multi-objetivo representan hipersuperficies, a diferencia de los problemas con un solo objetivo que tienen un único punto óptimo, los frentes óptimos de Pareto tienen una gran cantidad de geometrías.

En esta sección se describen algunas de las posibles geometrías asociadas a los frentes de Pareto. Empero, se requieren previamente una serie de conceptos útiles [99] que se definen a continuación.

Definición 2.5.1 (Conjunto convexo) *Un conjunto X es **convexo** si el segmento de línea entre cualesquiera dos puntos en X está en X , v.g., si para cualesquiera $x^1, x^2 \in X$ y para cualquier $\lambda \in [0, 1]$, se tiene:*

$$\lambda x^1 + (1 - \lambda)x^2 \in X \quad (2.4)$$

Si no se cumple la definición anterior, se dice que X es **no convexo**.

Definición 2.5.2 (Función convexa) *Una función $f : X \rightarrow \mathbb{R}$ es denominada **convexa** en X si para cualesquiera dos puntos $x^1, x^2 \in X$:*

$$f(\lambda x^1 + (1 - \lambda)x^2) \leq \lambda f(x^1) + (1 - \lambda)f(x^2) \quad (2.5)$$

donde $0 \leq \lambda \leq 1$.

Definición 2.5.3 (Función estrictamente convexa) *Una función $f : X \rightarrow \mathbb{R}$ es denominada **estrictamente convexa** en X si para cualesquiera dos puntos $x^1, x^2 \in X$:*

$$f(\lambda x^1 + (1 - \lambda)x^2) < \lambda f(x^1) + (1 - \lambda)f(x^2) \quad (2.6)$$

donde $0 < \lambda < 1$.

Por otra parte, una función f es (estrictamente) concava si $-f$ es (estrictamente) convexa.

Definición 2.5.4 (Cierre convexo) *El **cierre convexo** o **envoltura convexa** de un conjunto X , denotado como $CH(X)$, es el conjunto de todas las combinaciones convexas de puntos en X :*

$$CH(X) = \left\{ \sum_{i=1}^k \lambda_i x^i \mid x^i \in X, (\forall i : \lambda_i \geq 0), \sum_{i=1}^k \lambda_i = 1 \right\} \quad (2.7)$$

El cierre convexo $CH(X)$ representa el conjunto convexo más pequeño que contiene a X . En la figura 2.8 se ejemplifica la definición del cierre convexo.

Otra definición de un conjunto convexo en términos de su cierre convexo, se da a continuación.

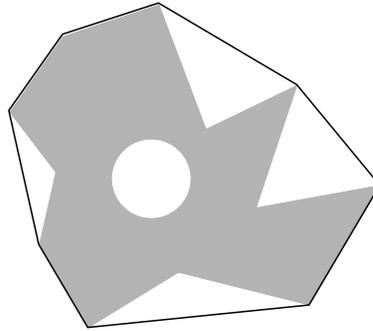


Figura 2.8: Ejemplo del cierre convexo.

Definición 2.5.5 (Conjunto convexo) *Un conjunto es convexo si y sólo si éste cubre su cierre convexo.*

Definición 2.5.6 (Conjunto cóncavo) *Un conjunto es cóncavo si y sólo si es cubierto por su cierre convexo.*

Definición 2.5.7 (Conjunto estrictamente convexo) *Un conjunto es estrictamente convexo si es convexo y no cóncavo.*

Definición 2.5.8 (Conjunto estrictamente cóncavo) *Un conjunto es estrictamente cóncavo si es cóncavo y no convexo.*

Definición 2.5.9 (Conjunto/Función lineal) *Un conjunto o función lineal es tanto convexa y cóncava pero no estrictamente convexa ni estrictamente cóncava.*

Por tanto, un POM es convexo (respectivamente cóncavo o lineal) si todos las funciones objetivo son convexas (respectivamente cóncavas o lineales) [42].

Un **frente mixto** es aquel con subconjuntos conexos donde cada uno es estrictamente convexo, estrictamente cóncavo, o lineal, pero no todos al mismo tiempo.

Adicionalmente, un **frente degenerado** es una hipersuperficie con dimensión $d < m - 1$, donde m es la dimensión del espacio de aptitud. Por ejemplo, un frente representado por un segmento de línea en un espacio de aptitud tridimensional, es degenerado.

Los frentes óptimos de Pareto degenerados pueden causar problemas a algunos algoritmos. Por ejemplo, algunos métodos empleados para fomentar un reparto equilibrado de soluciones a través del frente óptimo de Pareto podrían funcionar de manera diferente si el frente es una variedad geométrica con $d < m - 1$.

Otra geometría importante se expresa por medio de un **discontinuo**, es decir, un frente representado por un **conjunto desconectado**. Asimismo, los conjuntos óptimos de Pareto también pueden ser desconectados. Pese a que los conjuntos de Pareto desconectados usualmente se asignan a frentes de Pareto desconectados, esto no siempre es el caso. En la figura 2.9 se presentan algunas de estas geometrías.

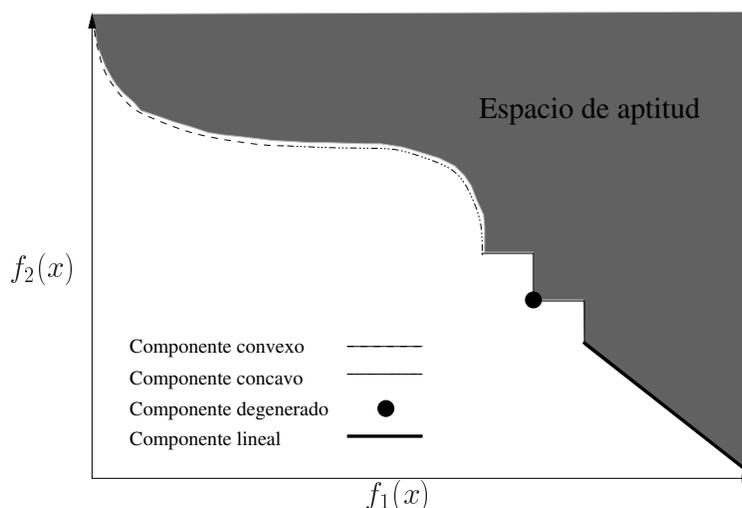


Figura 2.9: Ejemplo de un frente mixto y desconectado conteniendo un componente convexo, un componente cóncavo, un punto degenerado y un componente lineal.

Finalmente, otro concepto comúnmente usado en optimización multi-objetivo es la **rodilla**:

Definición 2.5.10 La **rodilla** es un punto en un frente óptimo de Pareto convexo (cóncavo) que ostenta la distancia más pequeña hacia el vector ideal (de nadir).

Los problemas que son de alta dimensionalidad, discontinuos, multimodales, y/o **NP-Completo** [9, 84] son nombrados **irregulares** [124]. En la siguiente sección se describen algunos métodos para lidiar con este tipo de problemas.

2.6. Métodos para resolver problemas multi-objetivo

Existe una gran gama de métodos capaces de resolver POMs. De acuerdo con Coello et al. [32], las técnicas de búsqueda general y optimización se pueden clasificar principalmente en tres sectores: enumerativas, determinísticas y estocásticas. En la figura 2.10, se muestra esta clasificación junto con algunos ejemplos por cada tipo.

En primera instancia, los esquemas enumerativos o exhaustivos son los más rudimentarios puesto que evalúan cada posible solución de un espacio de búsqueda finito. Para espacios de búsqueda pequeños esto no resulta ningún obstáculo, sin embargo, al considerar, por ejemplo, una instancia del problema del viajero (un problema NP-Completo) con 50 nodos, el espacio de búsqueda contiene 10^{62} posibles recorridos, lo que convierte a este esquema en prohibitivo.

En segundo lugar se encuentran los algoritmos determinísticos de búsqueda. A diferencia del esquema enumerativo, este tipo de algoritmos usan ciertas reglas heurísticas, es decir, usan el conocimiento acerca del problema para limitar el espacio de

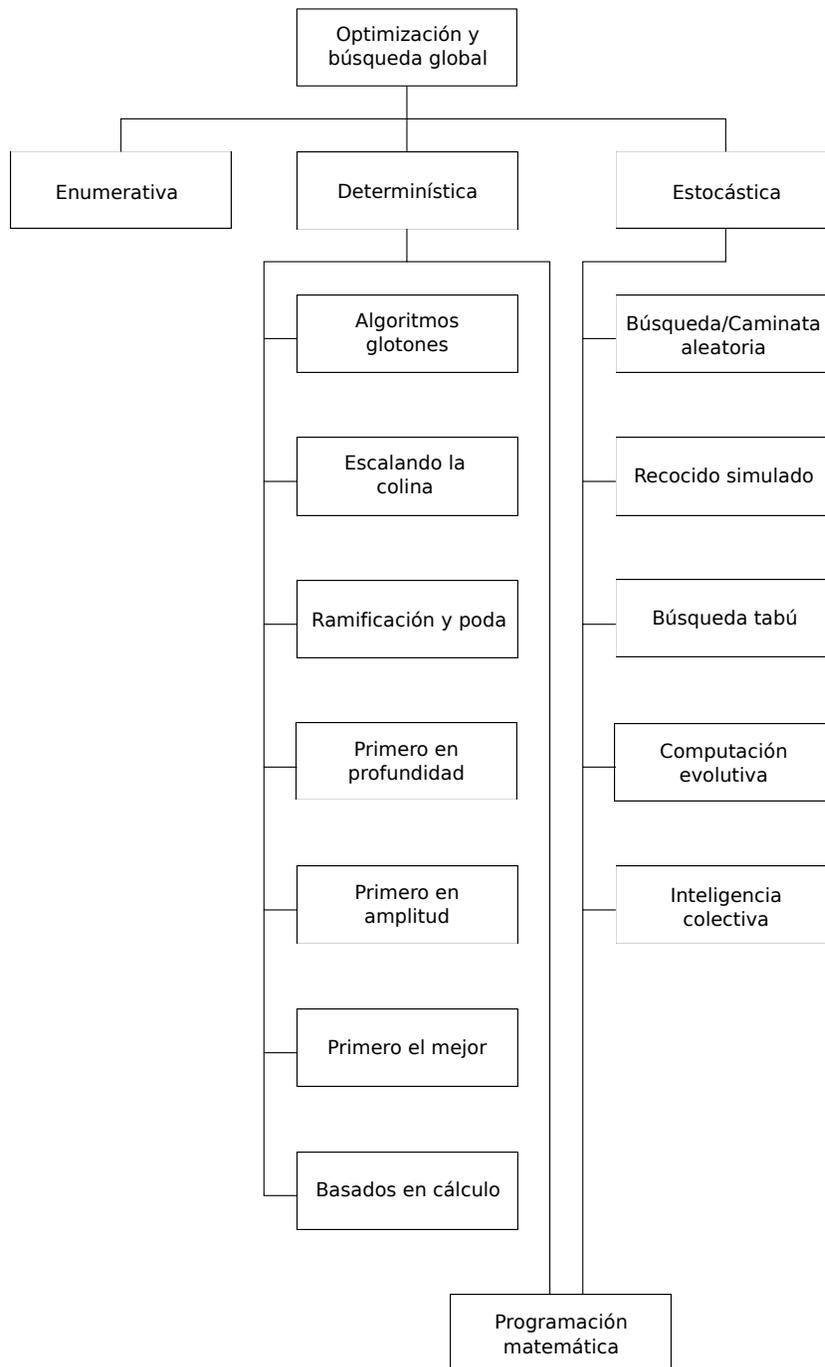


Figura 2.10: Técnicas de búsqueda y optimización global.

búsqueda. Gracias al empleo de las heurísticas⁴, se pueden encontrar soluciones aceptables en un tiempo razonable. Enseguida se describen algunos de los métodos de esta clasificación.

- **Algoritmos glotones.** Este tipo de algoritmos se basa en una heurística que elige en cada etapa la solución localmente óptima con la esperanza de construir una solución globalmente óptima. Para esto se asume que en efecto las soluciones localmente óptimas son parte del óptimo global [26]; en caso de que esto no se cumpla, el algoritmo falla.
- **Algoritmos escalando la colina.** Dada una única solución inicial de búsqueda, este tipo de algoritmos, en cada iteración, refinan tal solución moviéndose en la dirección de ascenso (descenso en caso de minimización) más empinada [146]. Esta clase de técnicas se desempeñan bien en funciones unifrонтales, pero la presencia de óptimos locales, puntos de silla o crestas en el paisaje de aptitud reducen su efectividad [160].
- **Técnicas de ramificación y poda.** Utilizan heurísticas o algoritmos de decisión específicos al problema para poder reducir su espacio de búsqueda [152, 78]. Este tipo de técnicas calculan el límite en un nodo dado que determina si el nodo es prometedor; posteriormente, los límites de varios nodos son comparados y el algoritmo elige el nodo más satisfactorio.
- **Búsqueda primero en profundidad.** Estos algoritmos son ciegos o no informados de que el orden de búsqueda es independiente de la localización de la solución (excepto para la terminación de la búsqueda). En cada iteración se expande un nodo, se generan todos los sucesores, se expande un sucesor y así sucesivamente. Si la búsqueda se ve bloqueada por alguna razón, se aplica *retroceso* hacia el padre del nodo bloqueado [88].
- **Búsqueda primero en amplitud.** Este esquema de búsqueda también es no informado. Difiere de la búsqueda primero en profundidad en las acciones tomadas después de la expansión del nodo a causa de la exploración progresiva del grafo una capa a la vez.
- **Búsqueda primero el mejor.** Usa información heurística para colocar valores numéricos en un nodo; el nodo con el valor más grande es examinado primero.
- **Métodos de búsqueda basados en cálculo.** Esta clase de buscadores aplican técnicas del cálculo y por ende requieren continuidad en el dominio de las variables para poder encontrar algún valor óptimo [7, 6].

Ante las dificultades que implica manejar problemas irregulares, las heurísticas de los algoritmos determinísticos son incapaces de limitar adecuadamente el espacio de

⁴Las heurísticas son criterios, métodos o principios para decidir cuál podría ser la solución (de entre un amplio conjunto de alternativas) más efectiva, con el objeto de alcanzar alguna meta [152].

búsqueda. En consecuencia, diversos métodos de búsqueda y optimización estocástica han sido propuestos como alternativas para la resolución de problemas irregulares. Sin embargo, una desventaja que presentan es que eventualmente podrían encontrar una solución y ésta no se puede asegurar que sea óptima. En general, estas técnicas proveen buenas soluciones a un amplio rango de problemas de optimización que son difíciles para los buscadores determinísticos [82, 105]. Ejemplos de estos métodos incluyen las siguientes metaheurísticas⁵:

- **Búsqueda/caminata aleatoria.** Un búsqueda aleatoria es la estrategia de búsqueda aleatoria más simple posible pues sencillamente evalúa un cierto número de soluciones elegidas aleatoriamente. Una caminata aleatoria es muy similar, excepto que la siguiente solución es aleatoriamente seleccionada usando la última solución evaluada como punto de inicio. Como en el caso enumerativo, estas técnicas no son eficientes para la resolución de POMs ya que no reducen el espacio de búsqueda con la incorporación de conocimiento del problema.
- **Recocido simulado.** Es una metaheurística explícitamente basada en el proceso de recocido, donde, por ejemplo, el agua es calentada y después gradualmente enfriada hasta obtener una configuración baja en energía. Este algoritmo consiste en hacer un movimiento aleatorio en lugar del mejor movimiento como en los algoritmos escalando la colina. Si el movimiento mejora el óptimo actual, se ejecuta; de lo contrario el movimiento se realiza con una probabilidad $p < 1$. Esta probabilidad decrece exponencialmente ya sea por tiempo o con la cantidad con la cual el óptimo ha sido empeorado. La analogía que usa esta metaheurística es que si la probabilidad de movimiento decrece lo suficientemente lento, el óptimo global podrá encontrarse.
- **Búsqueda tabú.** Esta técnica es considerada una meta-estrategia desarrollada para evitar quedar atrapado en soluciones localmente óptimas. Se mantiene un registro tanto de las soluciones visitadas como de los caminos que llevaron hasta ellas en diferentes memorias. Esta información restringe la elección de soluciones a evaluar subsecuentemente. La búsqueda tabú se suele integrar con otros métodos de optimización [81].
- **Computación evolutiva (CE).** Es un término genérico para varios métodos de búsqueda estocástica que simulan computacionalmente el proceso natural de la evolución. CE engloba técnicas como algoritmos genéticos, estrategias evolutivas, y programación evolutiva. Su principal fuente de inspiración es la evolución natural y el concepto Darwiniano de la supervivencia del más apto.
- **Inteligencia colectiva.** Engloba un conjunto de metaheurísticas inspiradas en el comportamiento colectivo de sistemas descentralizados y auto-organizados

⁵El término **metaheurística**, se deriva de la composición de dos palabras griegas. Heurística se deriva del verbo *heuriskein* que significa «encontrar», mientras que el sufijo *meta* significa «más allá, en un nivel más alto». En general, las metaheurísticas son estrategias de alto nivel que guían heurísticas subordinadas en el proceso de búsqueda [22].

naturales o artificiales. Inspirados por la naturaleza, especialmente por ciertos sistemas biológicos, los sistemas de inteligencia colectiva están típicamente formados por una población de agentes simples que interactúan localmente entre ellos y con su medio ambiente. Los agentes siguen reglas simples y, aunque no existe una estructura de control centralizado que dictamine el comportamiento de cada uno de ellos, las interacciones locales entre los agentes conducen a un comportamiento global complejo. Ejemplos de este tipo de sistemas son las colonias de hormigas (ver capítulo 3 para mayor referencia), la optimización mediante cúmulos de partículas, y el crecimiento bacteriano, entre otros.

La complejidad de los POMs y las dificultades de los métodos de búsqueda determinísticos causaron que la comunidad de **Investigación de Operaciones** [102] promoviera la creación de diversas técnicas de optimización bajo el nombre de **programación matemática**. Estos métodos son clasificados de acuerdo con la complejidad del POM. La **programación lineal** está diseñada para resolver problemas con funciones objetivo y restricciones lineales. Por el contrario, las técnicas de **programación no lineal** [146] tratan algunos POMs con funciones objetivo no lineales y funciones de restricción convexas. Finalmente, la **programación estocástica** es usada cuando los parámetros son valores aleatorios y las funciones objetivo están sometidas a ruido estocástico. Dependiendo del tipo de variables del problema, existen diversas variantes de estos métodos (v.g., discretas, enteras, binarias, programación entera mixta).

A continuación se presenta una clasificación de este tipo de métodos de acuerdo con el papel del tomador de decisiones en el proceso de solución [106]. Para mayor referencia, se recomienda al lector revisar la obra de Miettinen [143].

- **Métodos sin preferencia.** En esta clase de métodos, las opiniones del TD no son tomadas en consideración. El POM es resuelto usando métodos relativamente simples para finalmente presentar la solución al TD. El TD puede aceptar o rechazar la solución. Este tipo de métodos son útiles cuando el TD no tiene alguna preferencia especial y sólo busca obtener una solución óptima. Algunos métodos relevantes son: el método del criterio global y el método multi-objetivo del haz próximo.
- **Métodos a posteriori.** Estos métodos también pueden ser llamados **métodos para generar soluciones Pareto-óptimas**. Después de que el conjunto óptimo de Pareto (o una parte de él) ha sido generado, es presentado al TD, quien selecciona la solución con mayor preferencia entre todas las alternativas. El inconveniente es que el proceso de generación es usualmente costoso y en ocasiones, al menos, difícil. Por otra parte, es igualmente complicado para el TD elegir una opción entre un gran número de alternativas. Una diligencia aún más importante es cómo presentar o desplegar las alternativas al TD en una manera efectiva. Algunos métodos representativos son: el método de ponderación, el método de restricciones ϵ , el método de las métricas ponderadas y las funciones de escalarización por logros.

- **Métodos a priori.** En este caso, el TD debe especificar sus preferencias, esperanzas y opiniones antes del proceso de solución. La dificultad inherente es que el TD no necesariamente conoce de antemano qué esperar del problema ni sabe tampoco qué tan realistas son sus preferencias. Ejemplos de este tipo de técnicas son: el método de la función de valor, orden lexicográfico y programación por metas.
- **Métodos interactivos.** Suponiendo que el TD tiene el suficiente tiempo y capacidades para la cooperación, estos métodos pueden producir resultados más satisfactorios. Particularmente, sólo una parte de los puntos Pareto-óptimos tiene que ser generada y evaluada para que el TD pueda especificar y corregir sus preferencias y selecciones conforme el proceso de solución continúa. Esto también implica que el TD no necesita conocimiento previo del problema para estructurar sus preferencias. Sin embargo, la información manejada en el proceso debe ser realmente de utilidad y fácil de entender. Algunas instancias de esta clase de técnicas son: el método de Tchebycheff, el método del punto de referencia, la búsqueda por rayo de luz y el método de la dirección de referencia, entre otros [143].

Estos métodos clásicos resuelven el problema usando, por ejemplo, la información del gradiente de las funciones objetivo. Por ende, se basan en ciertas suposiciones acerca de la continuidad y convexidad de las funciones objetivo. Usualmente, la idea principal de estos métodos es transformar el POM en un problema con un solo objetivo usando restricciones, un proceso de jerarquización de los objetivos o funciones de agregación.

2.7. Optimización de muchas funciones objetivo

En las últimas décadas, los AEMOs han demostrado gradualmente su efectividad y eficiencia para resolver POMs [32]. No obstante, la mayor parte de estos algoritmos han sido típicamente probados empleando POMs con dos o tres funciones objetivo, aunque los problemas del mundo real suelen tener más de tres objetivos.

Recientemente, se han realizado estudios donde se ha probado que los AEMOs basados en optimalidad de Pareto tienen un desempeño muy pobre cuando resuelven POMs con cuatro o más objetivos [112, 118]. Farina y Amato [67] fueron los primeros en descubrir que la optimalidad y dominancia de Pareto son ineficientes para modelar y simular el proceso de decisión humana en problemas con muchas funciones objetivo. Ellos introdujeron el término *many-objective*⁶ (muchas funciones objetivo), que fue extendido posteriormente a *many-objective optimization* (optimización de muchas funciones objetivo) por Purshouse y Fleming [153]. Actualmente, en el área de la optimización multi-objetivo esta línea de investigación ha cobrado una alta relevancia.

⁶Un POM cumple con este término cuando contiene cuatro o más funciones objetivo.

De acuerdo con Ishibuchi et al. [112], existen principalmente tres problemas cuando se aplican AEMOs basados en optimalidad de Pareto a POMs con muchas funciones objetivo. A continuación se describen tales inconvenientes.

1. **Deterioro de la habilidad de búsqueda.** Al incrementar la dimensión del espacio objetivo, casi todas las soluciones se convierten en no dominadas. Esto provoca que la presión de selección se diluya, es decir, los mecanismos de convergencia basados en optimalidad de Pareto pierden todo efecto en la selección de soluciones. Evidentemente, esto provoca que la propiedad de convergencia de los AEMOs se vea sumamente comprometida.
2. **Crecimiento exponencial del número de soluciones requeridas para aproximar el frente de Pareto.** Los conjuntos de soluciones producidos por los AEMOs deben cubrir idealmente tanto como sea posible del frente de Pareto verdadero. Dado que el frente óptimo de Pareto es a lo más una $m - 1$ -variedad geométrica, el número de soluciones requeridas para su aproximación es del orden $\mathcal{O}(mr^{m-1})$, donde r es la resolución o número de hipercubos en los que se divide cada dimensión [168]. Por tanto, son necesarias miles de soluciones para poder aproximar \mathcal{PF}^* en un problema con $m \geq 4$ funciones objetivo.
3. **Dificultad en la visualización de las soluciones.** Un aspecto importante al finalizar el proceso de solución de un POM es la elección de una solución por parte del TD basado en sus preferencias. Al tener cuatro o más objetivos, la visualización de las soluciones se torna cada vez más complicada al no existir una forma directa de hacerlo. En consecuencia, el TD tendrá más problemas para elegir la solución final.

Con respecto al deterioro de la habilidad de búsqueda, Farina y Amato [67] mencionan que una solución bajo el esquema de optimalidad de Pareto puede encontrarse en uno de los tres siguientes conjuntos: mejores soluciones (b), peores soluciones (w) y soluciones equivalentes (e). La figura 2.11 muestra tales conjuntos para POMs con 2 y 3 objetivos. Los autores discuten que en un espacio objetivo en m dimensiones la porción del espacio con soluciones equivalentes crece de acuerdo con la siguiente regla: $e = (2^m - 2)/2^m$. Cuando $m \rightarrow \infty$, entonces $e \rightarrow 1$ lo cual implica que todo el espacio objetivo contendrá soluciones mutuamente no dominadas. A partir de esto, se puede observar claramente que los mecanismos de convergencia basados en optimalidad de Pareto se deterioran al seleccionar soluciones casi en forma aleatoria. En efecto, Mostaghim y Schmeck [144] han mostrado que un optimizador por búsqueda aleatoria obtiene mejores resultados que el NSGA-II [44] en un problema con 10 objetivos.

Otra cuestión que interfiere en que los AEMOs puedan escalar su funcionamiento a altas dimensiones en el espacio objetivo se debe al incremento del número de soluciones resistentes a la dominancia [93, 120, 48] (SRDs). Las SRDs son soluciones con un mal valor en al menos un objetivo, pero con valores cercanos a los óptimos en los objetivos restantes. En otras palabras, son soluciones no dominadas, aunque lejanas del frente óptimo de Pareto.

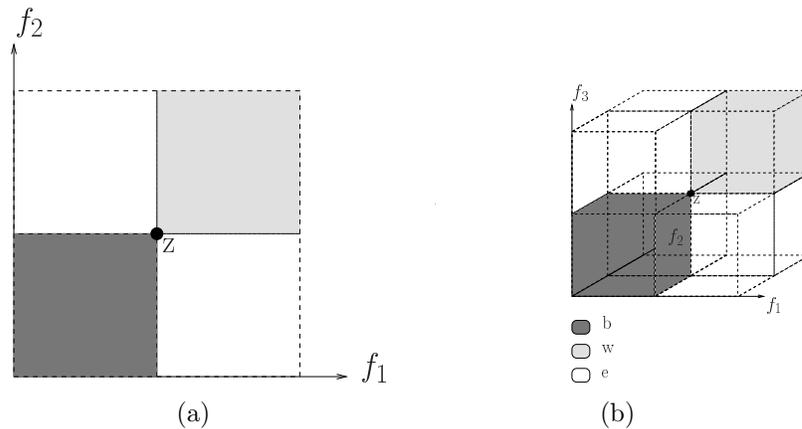


Figura 2.11: (a) conjuntos de soluciones en un problema bi-dimensional, (b) conjuntos de soluciones en un problema tri-dimensional. Las buenas soluciones se denotan por ‘b’, las malas soluciones por ‘w’ y las soluciones equivalentes mediante ‘e’.

De acuerdo con Bentley et al. [17], el número de vectores no dominados de dimensión m en un conjunto de tamaño N es del orden $\mathcal{O}(\ln^{m-1} N)$. Como resultado de esto, en problemas con muchas funciones objetivo, la selección de soluciones es llevada a cabo casi en forma aleatoria o guiada por los mecanismos de diversidad. A propósito de lo anterior, Knowles y Corne [118] demostraron que, para POMs con más de diez objetivos, una búsqueda puramente aleatoria obtenía mejores resultados comparado con un AEMO.

Como posibles soluciones a los problemas mencionados, Ishibuchi [112] y Li et al. [129] describen los siguientes:

1. **Redefinir relaciones de dominancia.** La forma directa para mejorar el escalamiento de los AEMOs es la generación de nuevas relaciones de dominancia que permitan aumentar la presión de selección hacia \mathcal{PF}^* , es decir, reduzcan el número de soluciones no dominadas en cada población. Una gran cantidad de estudios han sido realizados bajo esta directriz, de entre los que destacan los siguientes: dominancia ϵ [125], control del área de dominancia [162], dominancia de malla [130], clasificación por orden de preferencia [52], optimalidad k [68] y la optimalidad de Pareto difusa [97].
2. **Nuevos mecanismos de evaluación de aptitud.** En lugar de evaluar la aptitud de las soluciones basado en un esquema de Pareto, una propuesta que ha cobrado relevancia en años recientes es el uso de AEMOs basados en indicadores donde las métricas de evaluación de calidad (tales como el hipervolumen [192]) son usadas para evaluar cada solución [184, 111].
3. **Métodos basados en descomposición.** Este tipo de métodos descomponen un POM en un conjunto de subproblemas los cuales son optimizados en una

forma colaborativa. Se debe notar que este concepto es muy general, al grado en que tanto funciones de agregación o POMs más simples [133] pueden ser usados para formar los subproblemas. Dado que los vectores de pesos que definen niveles de aspiración están ampliamente distribuidos, se espera que las soluciones obtenidas cubran el frente de Pareto y se encuentren bien distribuidas. Un ejemplo de este tipo de métodos es MOEA/D [191].

4. **Mejorar los mecanismos de manejo de diversidad.** Los mecanismos de diversidad son usados generalmente como segundo criterio de selección en los AEMOs. Cuando los mecanismos de convergencia se ven afectados, los mecanismos de diversidad entran en acción. Lamentablemente, no mucho trabajo se ha realizado en la mejora de este tipo de mecanismos para espacios objetivo de alta dimensionalidad [129].
5. **Manejo de preferencias.** La información de preferencias es usada para concentrar la búsqueda en una pequeña región del frente de Pareto [161, 47, 46]. Debido a esto, las dificultades mencionadas en optimización de muchas funciones objetivo se ven aliviadas ya que se reduce el tamaño del espacio de búsqueda.
6. **Reducción de objetivos.** Se basa en la suposición de la existencia de objetivos redundantes en optimización de muchas funciones objetivo. Para esto, la idea es reducir la dimensión del espacio objetivo usando técnicas como el análisis de componentes principales [163] con el fin de identificar el frente de Pareto embebido. Consecuentemente, los AEMOs tradicionales (o sea, basados en jerarquización de Pareto) para dos y tres objetivos pueden ser fácilmente aplicados.

Finalmente, con respecto a la visualización, diferentes técnicas para mostrar soluciones en m dimensiones han sido propuestas en la literatura [147, 190] donde los vectores objetivo son trasladados a un espacio de más baja dimensión para su visualización. Un gran número de técnicas de visualización de vectores en altas dimensiones han sido también propuestas en el área llamada toma de decisiones multicriterio (*Multiple Criteria Decision Making*) [143].

2.8. Indicadores de calidad como mecanismos de selección

Muchos AEMOs emplean generalmente dos mecanismos de selección de soluciones. El primero de ellos busca converger hacia el frente óptimo de Pareto mediante el uso de la optimalidad de Pareto, mientras que el segundo tiene por objeto mejorar la diversidad del conjunto solución basado en el cálculo de distancias euclidianas en el espacio objetivo. Este tipo de mecanismos han probado su efectividad en POMs con dos y tres funciones objetivo, sin embargo, cuando se incrementa la dimensión del espacio objetivo su desempeño se deteriora notablemente [112].

Los indicadores de calidad (también denotados como indicadores de desempeño, medidas de calidad o simplemente indicadores) [32, 94, 117], empleados principalmente para medir la calidad de las salidas de los optimizadores multi-objetivo, han sido usados recientemente como una alternativa para guiar la búsqueda en espacios objetivo de altas dimensiones.

2.8.1. Propiedades

Los indicadores de calidad pueden ser usados para cumplir diferentes metas: comparar numéricamente las salidas de los optimizadores multi-objetivo, guiar el proceso de búsqueda siendo aplicados en mecanismos de selección y, finalmente, como criterios de paro. A continuación se presentan algunas definiciones formales acerca de estas medidas de desempeño.

Definición 2.8.1 *Un indicador de desempeño (unario) es una función $I : \mathcal{Z} \subset \mathbb{R}^m \rightarrow \mathbb{R}$ que asigna a cada conjunto de aproximación un número real. Donde \mathcal{Z} es el espacio de aptitud.*

Definición 2.8.2 *Un indicador I se dice ser estrictamente monótono si y sólo si cada vez que una aproximación al frente de Pareto domina totalmente a otra, entonces el valor del indicador del frente dominante será también mejor (considerando maximización del indicador). Formalmente, esto se puede expresar como:*

$$\forall A, B \in \mathcal{Z} : A \prec B \Rightarrow I(A) > I(B)$$

donde \prec representa la dominancia de Pareto subyacente del conjunto.

Definición 2.8.3 *Un indicador I se dice ser débilmente monótono si y sólo si para cada aproximación al frente de Pareto que es comparada con otra aproximación se cumple que al ser al menos tan buenos en términos de la relación de dominancia implica tener valores del indicador al menos tan buenos. Formalmente, esto se puede expresar como:*

$$\forall A, B \in \mathcal{Z} : A \preceq B \Rightarrow I(A) \geq I(B)$$

donde \preceq representa la dominancia débil de Pareto subyacente del conjunto.

Un indicador que cumpla con la definición anterior se denomina **Pareto compatible**. En el contexto de la teoría del orden, un indicador Pareto compatible I es una función que preserva el orden de (\mathcal{Z}, \preceq) a (\mathbb{R}, \geq) .

Los indicadores Pareto compatibles definen refinamientos del orden parcial inducido por la dominancia débil de Pareto. Se debe observar que muchos de los indicadores que son empleados en la literatura especializada de optimización multi-objetivo no son Pareto compatibles. Varios indicadores populares están diseñados para evaluar sólo un aspecto de un conjunto de aproximación, tal como la convergencia hacia el

frente óptimo de Pareto, o su cobertura en el espacio objetivo. Este tipo de indicadores de calidad, a veces referidos como indicadores **funcionalmente independientes** son, por definición, Pareto no compatibles.

Definición 2.8.4 (Indicador Pareto no compatible) *Cualquier indicador que pueda producir para cualesquiera conjuntos de aproximación $A, B \in \mathcal{Z}$ una preferencia por A sobre B cuando B es preferible a A con respecto a la dominancia débil de Pareto ($B \preceq A \wedge \neg A \preceq B$) es **Pareto no compatible**.*

2.8.2. Hipervolumen

El indicador de hipervolumen (HV), también llamado métrica S , fue primeramente propuesto y usado por Zitzler y Thiele [199, 200], siendo denotado como el **tamaño del espacio cubierto**. HV es el único indicador unario conocido que es estrictamente monótono con respecto a la dominancia de Pareto. Como resultado de lo anterior, se garantiza que \mathcal{PF}^* ostenta el máximo valor posible de hipervolumen, mientras que cualquier conjunto de aproximación tendrá un valor peor del indicador. Se ha demostrado que mediante la maximización del hipervolumen se producen conjuntos de soluciones cuya cobertura es máxima a lo largo del frente óptimo de Pareto (aunque esto no necesariamente significa que tales soluciones estén uniformemente distribuidas a lo largo de \mathcal{PF}^*).

El hipervolumen mide el tamaño de la porción del hiperespacio objetivo que es dominado por un conjunto de aproximación [193] (véase figura 2.12). A continuación se define matemáticamente HV.

Definición 2.8.5 (Indicador hipervolumen) *Basado en un punto de referencia anti-óptimo \vec{r} , el hipervolumen de un conjunto de aproximación A está definido como*

$$HV(A, \vec{r}) = \mathcal{L} \left(\bigcup_{\vec{a} \in A} \{ \vec{a}' \mid \vec{a} \prec \vec{a}' \prec \vec{r} \} \right) \quad (2.8)$$

donde $\mathcal{L}(\cdot)$ denota la medida de Lebesgue de un conjunto [11]. Cuanto más grande sea el valor del indicador, mejor será el conjunto de aproximación.

La contribución al hipervolumen de una solución refleja la influencia de un único punto en la calidad del conjunto de aproximación. Este concepto se define formalmente como sigue.

Definición 2.8.6 (Contribución al hipervolumen) *La contribución al hipervolumen de una solución $\vec{a} \in A$ está definida como*

$$C_{HV}(\vec{a}, A, \vec{r}) = HV(A, \vec{r}) - HV(A \setminus \{\vec{a}\}, \vec{r}) \quad (2.9)$$

Aunque este indicador cuenta con buenas propiedades matemáticas, se ha demostrado que no existe algún algoritmo que pueda calcularlo en orden polinomial. Por el

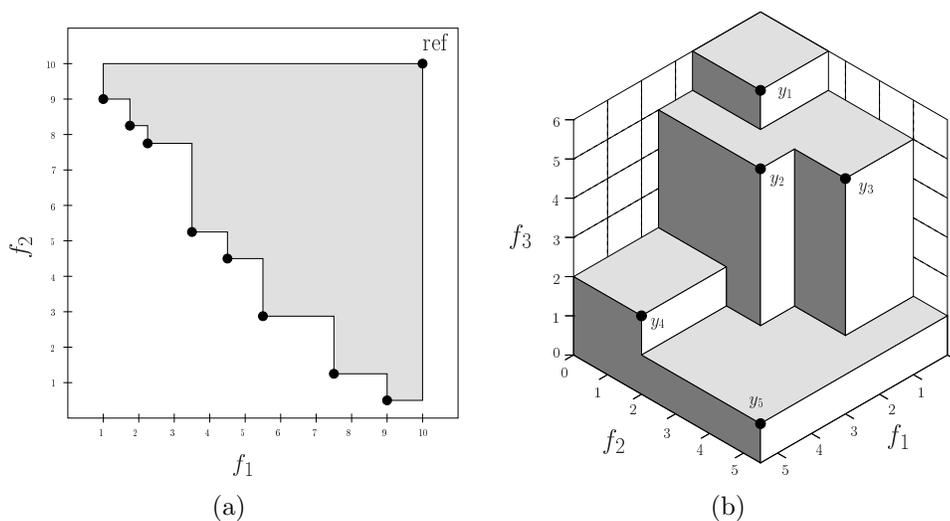


Figura 2.12: Ejemplos del hipervolumen dominado en dos y tres dimensiones. La figura de la izquierda muestra un caso de minimización, mientras que la de la derecha presenta un caso de maximización.

contrario, los algoritmos conocidos son de orden exponencial con respecto al número de objetivos m . Sin embargo, HV ha sido un detonante para la generación de AEMOs que lo empleen en su mecanismo de selección. En las siguientes subsecciones, se describen dos algoritmos relevantes: IBEA y SMS-EMOA.

2.8.2.1. IBEA

Zitzler y Künzli [196] propusieron el *Indicator-Based Evolutionary Algorithm* (IBEA) el cual es un AEMO general que puede integrar diferentes indicadores de calidad como mecanismos de selección. La idea principal de IBEA es formalizar las preferencias en términos de generalizaciones continuas de la relación de dominancia. IBEA permite la adaptación ante información de preferencia arbitraria y a escenarios de optimización; adicionalmente, no requiere ningún mecanismo de preservación de diversidad. Comparado con el trabajo de Knowles [119], IBEA es más general puesto que el tamaño de población puede ser arbitrario porque sólo compara pares de individuos y no conjuntos de aproximación enteros. En síntesis, IBEA transforma el problema de optimización general (ver ecuación 2.1) a un problema de optimización de un indicador I dado.

Zitzler y Künzli [196] integraron HV en el mecanismo de asignación de aptitud de IBEA dando como resultado el algoritmo IBEA-HV. Los resultados fueron comparados con respecto a SPEA2 [198] y NSGA-II [44], de lo cual se observa que IBEA-HV se desempeña significativamente mejor.

2.8.2.2. SMS-EMOA

S metric Selection-Evolutionary Multi-Objective Algorithm (SMS-EMOA), propuesto por Beume et al. [18], tiene como ideas principales: (1) el uso de un mecanismo de selección basado en HV para descartar el individuo cuya contribución al hipervolumen sea la menor y (2) la aplicación del ordenamiento no dominado del NSGA-II como criterio de asignación de rangos.

SMS-EMOA garantiza que el hipervolumen cubierto por una población no pueda decrementarse en futuras generaciones. Además, las soluciones producidas están bien distribuidas en el frente óptimo de Pareto, siendo las regiones limítrofes y la rodilla del frente las más favorecidas. Una característica importante de SMS-EMOA es que puede aproximar conjuntos de Pareto haciendo uso de un pequeño número de individuos. Sin embargo, su costo computacional es muy alto cuando se resuelven problemas con muchas funciones objetivo.

2.8.3. R2

La familia de indicadores R [95] está basada en el uso de funciones de utilidad que transforman un vector $\vec{y} \in \mathbb{R}^m$ a un valor de utilidad escalar $u \in \mathbb{R}$ para evaluar la calidad relativa de dos aproximaciones al frente de Pareto [28].

Dentro de la familia R se puede encontrar el indicador R2 que es una propuesta recomendada [117] para POMs con muchas funciones objetivo. Las características distintivas de R2 son las siguientes: (1) cumple con ser débilmente monótono, v.g., $I_{R2}(A) \geq I_{R2}(B)$ en caso de que $A \preceq B$, (2) simultáneamente evalúa todos los aspectos deseables de una aproximación al frente de Pareto, (3) las soluciones que genera están usualmente uniformemente distribuidas, y (4) es mucho menos costoso computacionalmente que HV. A continuación se dan las definiciones formales de R2.

Definición 2.8.7 *Para un conjunto U de funciones de utilidad generales, una distribución de probabilidad p sobre U , y un conjunto de referencia R , el indicador R2 de un conjunto solución A está definido como la utilidad esperada:*

$$R2(R, A, U, p) = \int_{u \in U} \max_{\vec{r} \in R} \{u(\vec{r})\} p(u) du - \int_{u \in U} \max_{\vec{a} \in A} \{u(\vec{a})\} p(u) du \quad (2.10)$$

Definición 2.8.8 *Para un conjunto discreto y finito U y una distribución uniforme p sobre U , el indicador R2 puede ser escrito como:*

$$R2(R, A, U) = \frac{1}{|U|} \sum_{u \in U} \left(\max_{\vec{r} \in R} \{u(\vec{r})\} - \max_{\vec{a} \in A} \{u(\vec{a})\} \right) \quad (2.11)$$

Puesto que el primer sumando es una constante si se asume que R es constante, se elimina el primer sumando y se le llama también R2 al indicador unario resultante.

Definición 2.8.9 Para un conjunto de referencia constante, el indicador R2 puede ser definido como un indicador unario

$$R2(A, U) = -\frac{1}{|U|} \sum_{u \in U} \max_{\vec{a} \in A} \{u(\vec{a})\} \quad (2.12)$$

Existen diferentes opciones con respecto a las funciones de utilidad requeridas, p. ej., suma ponderada, métrica de Tchebycheff ponderada, función escalar por logros (ASF, por sus siglas en inglés), etc. [143]. Estas funciones de utilidad tienen asociado un conjunto de vectores de pesos convexos⁷ Λ uniformemente distribuidos y un punto de referencia \vec{z} , para mantener diversidad.

Definición 2.8.10 El indicador R2 de un conjunto solución A para un conjunto dado de vectores de pesos convexos Λ , usando ASF y un punto de referencia \vec{z} se define como

$$R2(A, \Lambda, \vec{z}) = \frac{1}{|\Lambda|} \sum_{\vec{\lambda} \in \Lambda} \min_{\vec{a} \in A} \left\{ \max_{i \in \{1, \dots, m\}} \frac{|a_i - z_i|}{\lambda_i} \right\} \quad (2.13)$$

Finalmente, la contribución de una solución $\vec{a} \in A$ al indicador R2 se define a continuación:

$$C_{R2}(\vec{a}, A, \Lambda, \vec{z}) = R2(A, \Lambda, \vec{z}) - R2(A \setminus \{\vec{a}\}, \Lambda, \vec{z}). \quad (2.14)$$

En las siguientes subsecciones se introducen brevemente dos AEMOs basados en R2: MOMBI (en sus dos versiones) y R2-MOGA.

2.8.3.1. MOMBI y MOMBI-II

Hernández y Coello [100] propusieron la *Many-Objective Metaheuristic Based on the R2 Indicator* (MOMBI) en 2013, cuya idea principal se basa en la creación de un mecanismo de asignación de rangos, similar al del NSGA-II, pero fundamentado en el indicador R2, denominado «clasificación-R2». MOMBI agrupa las soluciones que optimizan el conjunto de funciones de utilidad elegidas asignándoles el mejor grado de preferencia. Tales puntos son inmediatamente removidos y un segundo grado de preferencia es asignado en la misma manera. El proceso continúa hasta que todas las soluciones han sido asignadas con un rango. MOMBI fue comparado con MOEA/D y SMS-EMOA, superando al primero y teniendo un desempeño competitivo con el último aunque con un menor costo computacional.

Los mismos autores, en 2015, propusieron una mejora a MOMBI denominada MOMBI-II [101]. Un mecanismo de aproximación estadística del vector ideal y del vector de nadir, junto con el uso de la ASF [143] como función de utilidad permitió que MOMBI-II pudiera superar significativamente a MOMBI, Δ_p -DDE [159], NSGA-III [45], R2-MOGA [53] y R2-IBEA [181] en la mayoría de las funciones de prueba empleadas.

⁷Para que un vector $\vec{v} = (v_1, v_2, \dots, v_n)^T$ sea convexo se debe cumplir que $\sum_{i=1}^n v_i = 1$.

2.8.3.2. R2-MOGA y R2-MODE

Díaz-Manríquez et al. [53], propusieron un esquema de asignación de rangos basado en el indicador R2, denominado ordenamiento rápido R2. Este esquema de ordenación es una versión modificada del ordenamiento no dominado del NSGA-II. La idea principal detrás del algoritmo de ordenamiento es acomodar las soluciones con base en su contribución al indicador R2, siendo aquellas con mayor contribución las que obtengan el mejor rango. Posteriormente, tales soluciones son removidas y se vuelve a repetir el proceso hasta que no haya más soluciones por clasificar.

Los autores integraron este algoritmo de ordenamiento en dos motores de búsqueda lo cual dio origen a dos AEMOs: el *R2-Multi-Objective Genetic Algorithm* (R2-MOGA) y el *R2-Multi-Objective Differential Evolution* (R2-MODE). Estos AEMOs fueron comparados con respecto a NSGA-II [44], MOEA/D [191] y SMS-EMOA [18], obteniendo resultados competitivos en POMs con dos y tres objetivos. Para el caso de problemas con muchas funciones objetivo, R2-MOGA y R2-MODE fueron comparados con SMS-EMOA y una versión modificada de éste, denominada SMS-EMOA2 que aproxima el valor del hipervolumen. Los resultados, en este caso, indicaron que las propuestas obtuvieron mejores resultados tanto en las medidas de calidad adoptadas como también en el costo computacional.

2.9. Resumen

Este capítulo introduce los conceptos básicos de la optimización multi-objetivo, que ha sido ampliamente usada en diferentes disciplinas. Primeramente, se define matemáticamente un POM, del cual es indispensable que los objetivos se encuentren en conflicto entre sí. Posteriormente, se introducen los conceptos de dominancia normal, débil y estricta de Pareto para dar paso así a la optimalidad de Pareto y los conceptos fundamentales de conjunto óptimo y frente óptimo de Pareto que representan la salida de un optimizador multi-objetivo. Adicionalmente, se describen las características ideales con que debe contar una aproximación al frente de Pareto: convergencia, distribución y cobertura.

Se definieron tres puntos especiales del espacio objetivo: el vector ideal, el vector de nadir y el vector utópico. Estos vectores usualmente son empleados como puntos de referencia para poder acotar el frente óptimo de Pareto. Asimismo, se estableció la función que desempeña el tomador de decisiones como única persona capaz de dictar preferencias acerca de las soluciones generadas por un optimizador multi-objetivo, con el fin de elegir una solución final al POM.

Se discutieron diferentes características del paisaje de aptitud de los POMs junto con algunas geometrías posibles a adoptar. También se presentaron en una taxonomía los métodos empleados en el área de búsqueda y optimización. Estas técnicas fueron clasificadas en: enumerativas (evalúan completamente el espacio de búsqueda), determinísticas (reducen el espacio de búsqueda aplicando heurísticas) y estocásticas (alternativas para solucionar POMs irregulares mediante metaheurísticas). Además,

las técnicas de programación matemática clasificadas por la acción del TD se describieron.

Finalmente, la optimización de muchas funciones objetivo, mejor conocida como *many-objective optimization*, fue expuesta como un área de gran interés para la comunidad de optimización multi-objetivo. Se discutieron los problemas y retos que presenta esta línea de investigación, así como también algunas vías de solución como lo es el uso de indicadores de calidad en los mecanismos de selección para poder reestablecer la presión de selección perdida por el uso de la optimalidad de Pareto en estos POMs de alta dimensionalidad.

Capítulo 3

Optimización mediante colonias de hormigas

El estudio constante de las colonias de insectos ha demostrado que su estructura social tiene propiedades interesantes como lo es la auto-organización. Gracias a esto, no sólo la biología se ha visto beneficiada, sino que también lo han sido las ciencias computacionales con el modelado de poderosas herramientas para la transferencia de conocimiento que han dado paso a áreas como el **diseño de sistemas inteligentes**.

La expresión **inteligencia de colectiva** fue usada por primera vez por Beni, Hackwood y Wang [14, 15, 16] en el contexto de sistemas robóticos celulares. No obstante, a lo largo de los últimos años esta expresión se ha extendido para incluir el diseño de algoritmos o dispositivos distribuidos que resuelven problemas inspirados en el comportamiento colectivo de insectos y otras sociedades de animales [24, 116].

Este capítulo tiene como finalidad describir una metaheurística inspirada en el comportamiento social de las hormigas en su búsqueda de alimento, llamada OCH. En la sección 3.1 se presenta la inspiración biológica que dio lugar a la construcción de la metaheurística. La sección 3.2 describe detalladamente el esquema general de la metaheurística. Algunas propuestas de OCH para la solución de problemas de optimización mono-objetivo en espacios de búsqueda discretos y continuos se muestran en la sección 3.3 y, así mismo, la sección 3.4 introduce las versiones multi-objetivo de OCH para ambos tipos de problemas. Finalmente, se realiza un resumen del capítulo en la sección 3.5.

3.1. Metáfora biológica

Los insectos que viven en colonias tales como las hormigas, abejas, termitas o avispas han sido materia de estudio de los naturalistas y, recientemente, una fuente de inspiración para las ciencias de la computación. Las colonias de insectos sociales son sistemas distribuidos que a pesar de la simplicidad de sus individuos, presentan una estructura social altamente organizada [61]. En efecto, una colonia de insectos sociales es indudablemente un sistema descentralizado con la capacidad de solucionar

problemas debido a la interacción de los individuos que la componen [24]. Una de las características más importantes de los insectos sociales es que ellos pueden resolver estos problemas de una manera muy flexible y robusta: la flexibilidad permite la adaptación ante ambientes cambiantes, mientras que la robustez dota a la colonia con la habilidad de funcionar aún cuando algunos individuos fallen al realizar sus labores.

En una colonia de insectos sociales, un individuo (agente) no se encarga de ejecutar todos los tipos de tareas, sino que se especializa en un conjunto de actividades de acuerdo con su morfología, edad u oportunidades. Gracias a esta división de labores, se pueden ejecutar diferentes faenas simultáneamente por grupos de agentes especializados. Este esquema de trabajo se cree más eficiente que ejecutar las tareas secuencialmente por individuos no especializados [158, 114].

De acuerdo con Bonabeau et al. [24], la idea principal detrás del sistema de comportamiento altamente coordinado de las colonias de insectos es la **auto-organización** (AO). AO es un conjunto de mecanismos dinámicos por los cuales surgen estructuras en un nivel global del sistema debido a la interacción de sus componentes de bajo nivel. Las reglas que especifican las interacciones entre las unidades constituyentes del sistema son ejecutadas con base en información puramente local, sin referencia al patrón global, el cual es una propiedad emergente del sistema en lugar de una propiedad impuesta por algún tipo de influencia externa.

AO en los insectos sociales a menudo requiere de la interacción entre ellos: tales interacciones pueden ser directas o indirectas. Las interacciones directas se pueden ver de manera obvia en acciones como intercambio de comida o líquidos, contacto mediante antenas, mandíbulas, visual o químico (el olor de otros insectos del nido), entre otros. Sin embargo, la interacción indirecta es más sutil: dos individuos interactúan indirectamente cuando uno de ellos modifica el ambiente y el otro responde a este cambio ambiental en un tiempo posterior. Este tipo de interacción es un ejemplo de *stigmergy*¹. Grassé [87, 86] introdujo el término *stigmergy* para explicar la coordinación de tareas y regulación en el contexto de la construcción de nidos de termitas del género *Macrotermes*. *Stigmergy* provee un mecanismo general donde el comportamiento individual modifica el ambiente, lo que en turno propicia la modificación del comportamiento de otros individuos de la colonia [24]. De hecho, muchos comportamientos a nivel colonia observados en insectos sociales pueden ser explicados por modelos simples donde únicamente la comunicación está presente.

La auto-organización y *stigmergy* son los dos pilares centrales en los que diferentes algoritmos han sido construidos. En este caso, un tipo de *stigmergy* artificial coordina las sociedades de agentes artificiales. Para los fines de este trabajo, se centra la atención en colonias de hormigas.

En el caso de las hormigas, la comunicación indirecta entre individuos se da a través del uso de una sustancia química producida por estos insectos. Esta sustancia es denominada **feromona**. Particularmente importante para algunas especies de hormigas es la **feromona de rastro**. Este tipo de feromona es usada por algunos tipos

¹El término *stigmergy* proviene del griego *stigma*: picadura y *ergon*: trabajo.

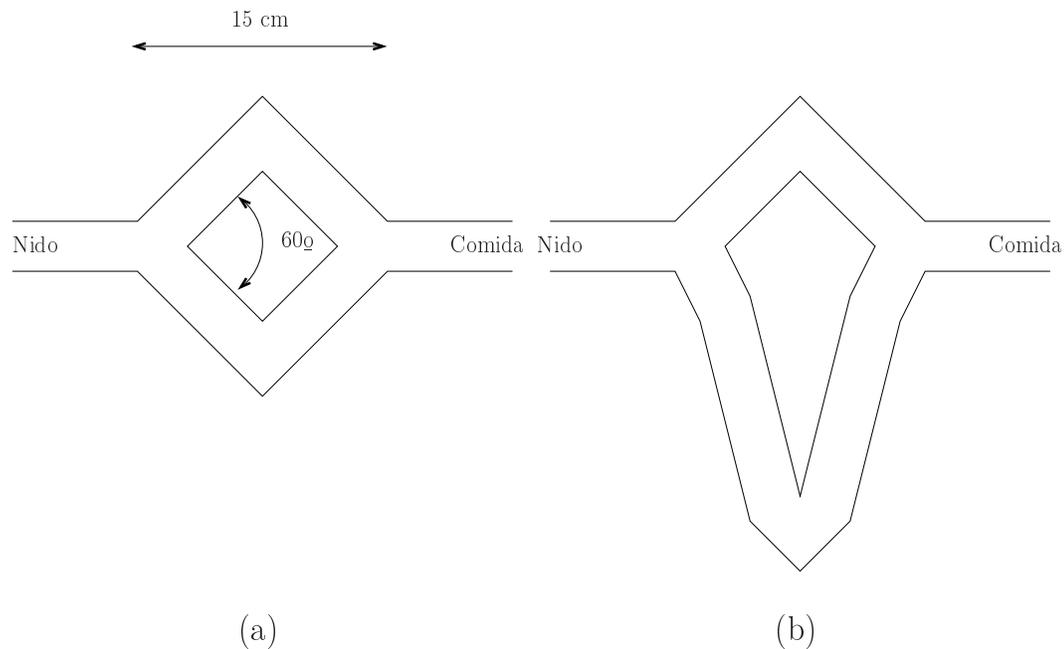


Figura 3.1: Configuración del experimento del doble puente. (a) Los brazos del puente tienen la misma longitud. (b) Los brazos tienen longitud diferente.

de hormigas como la *Lasius niger* o la hormiga argentina *Iridomyrmex humilis* [85] para la construcción de caminos entre una fuente de comida y el nido. Mediante la detección de los rastros de feromonas, las hormigas pueden descubrir el camino a la fuente de alimento dejado por otras compañeras de nido. Este comportamiento colectivo por el cual una hormiga se ve influenciada por el rastro químico dejado por otras hormigas es la fuente principal de inspiración de la optimización mediante colonias de hormigas [61].

El comportamiento de búsqueda de comida de algunas hormigas está mediado por la comunicación indirecta a través de feromonas. En su camino del nido a la fuente de alimento y viceversa, las hormigas depositan feromona sobre el suelo, dando paso así a caminos de rastros de feromona. Las hormigas detectan esta feromona y tienden a elegir, con mayor probabilidad, aquellos caminos con una mayor concentración de esta sustancia. Para probar esta hipótesis, Deneubourg et al. [85, 51] diseñaron un experimento donde unían un nido y una fuente de alimento mediante un puente de dos conexiones (ver figura 3.1) para después liberar hormigas y observar su comportamiento.

En la primera fase del experimento (figura 3.1(a)) ambas conexiones eran del mismo tamaño, observándose que al inicio se realizaban elecciones aleatorias sobre qué camino tomar. Sin embargo, después de cierto tiempo, las hormigas usaban indistintamente uno u otro brazo del puente con mayor probabilidad. Este comportamiento se puede explicar como sigue. Al inicio, dado que no hay rastros de feromona, las hormigas al no tener preferencias eligen aleatoriamente uno u otro camino. Por cues-

tiones de fluctuaciones aleatorias, un poco más de hormigas tenderán a elegir un brazo sobre el otro. En consecuencia, éste tendrá una mayor concentración de feromonas y estimulará en mayor medida a las hormigas hasta lograr convergencia. Este proceso **auto-catalítico** o de **retroalimentación positiva** es, en efecto, una instancia de comportamiento auto-organizado: un patrón macroscópico (la convergencia en la elección de un brazo del puente) emerge de las interacciones que se dan lugar a nivel microscópico (interacciones locales entre individuos de la colonia) [30, 92, 145]

En la segunda fase del experimento se alteraron las longitudes de los brazos del puente como se muestra en la figura 3.1(b), de tal forma que uno de los brazos tuviera el doble de largo. El resultado más importante observado es que después de un tiempo las hormigas tendían a elegir con mucha mayor probabilidad el camino más corto hacia la fuente de alimento. La razón de este comportamiento yace en que las hormigas que eligen el camino más corto llegan más rápidamente a la fuente de alimento y, por ende, son también las primeras en regresar al nido. Por este motivo, las siguientes hormigas que salen del nido ven sesgada su decisión por la mayor concentración de feromonas en el camino más corto. Por tanto, la cantidad de feromona que se acumula en el camino más corto comienza a ser mucho mayor y, eventualmente, éste será usado por la mayoría de las hormigas debido al proceso auto-catalítico. No obstante, no todas las hormigas usan el camino más corto. Deneubourg et al. [51] observaron en el experimento que un pequeño porcentaje tomaba el camino más largo. Esto podría ser interpretado como un comportamiento puramente exploratorio.

En un experimento adicional, se estudió qué pasaba con una colonia cuando se le ofrece un nuevo camino más corto después de alcanzar convergencia. En este caso, Deneubourg et al. [51] conectaban únicamente el nido y la fuente de alimento con el camino más largo y liberaban a las hormigas. Después de media hora, añadían el camino más corto al puente. En la gran mayoría de los experimentos, los autores notaron que las hormigas no podían olvidar el primer camino para converger así al camino más corto añadido posteriormente. El camino más corto sólo era elegido esporádicamente, por lo que la colonia quedaba atrapada en una solución subóptima. Esto se explica por la gran concentración de feromona en el camino más largo y lo lento del proceso de evaporación de esta sustancia, que no permitía romper con el proceso de retroalimentación positiva. La evaporación de feromonas, que puede favorecer el descubrimiento de nuevos caminos, es un proceso muy lento: el tiempo de vida de la feromona es tan largo como el tiempo de recorrido del camino [85]. Lo anterior significa que la feromona se evapora tan lentamente que no permite a la colonia «olvidar» el camino subóptimo y así «aprender» el nuevo camino más corto.

Como conclusión final de estos experimentos, el puente doble muestra claramente que las colonias de hormigas tienen una capacidad de optimización intrínseca. A través del uso de modelos probabilísticos basados en información local, las hormigas pueden encontrar los caminos más cortos entre dos puntos en su medio ambiente. En consecuencia, OCH fue inicialmente empleado para problemas de optimización combinatoria.

3.2. Descripción de la metaheurística

La optimización mediante colonias de hormigas (OCH) [61] es una metaheurística en la cual una colonia de hormigas artificiales cooperan con el fin de encontrar buenas soluciones a problemas difíciles. La cooperación es un aspecto clave en el diseño de estos algoritmos y se da a través de la comunicación indirecta (*stigmergy*) de los agentes.

Los algoritmos basados en OCH son procedimientos de búsqueda estocástica. El componente central de este tipo de algoritmos es el **modelo de feromonas**, el cual es usado para probabilísticamente muestrear el espacio de búsqueda [61, 24, 56]. El modelo de feromonas puede ser derivado de un «modelo» del problema de optimización a resolver. Inicialmente, OCH fue orientado a resolver problemas combinatorios, sin embargo, en esta sección, sin pérdida de generalidad, el problema podrá ser considerado como discreto o continuo sin distinción alguna.

Definición 3.2.1 (Modelo de un problema de optimización) *Un modelo $\mathcal{P} = (\Omega, \psi, f)$ de un problema de optimización (ya sea discreto o continuo) está constituido por los siguientes componentes:*

- *un espacio de búsqueda Ω definido sobre un conjunto finito de variables discretas/continuas de decisión y un conjunto ψ de restricciones entre las variables;*
- *una función objetivo $f : \Omega \rightarrow \mathbb{R}$ a ser minimizada.*

Una solución factible $s^* \in \Omega$ es llamada **solución globalmente óptima** (u óptimo global), si $f(s^*) \leq f(s) \forall s \in \Omega$. El conjunto de soluciones globalmente óptimas es denotado por $\Omega^* \subset \Omega$. Para resolver el problema de optimización se debe encontrar una solución $s^* \in \Omega^*$.

En el algoritmo 3.1 se muestra el marco general de un algoritmo básico basado en OCH. Éste funciona como sigue. En cada iteración t , N hormigas probabilísticamente construyen una solución para \mathcal{P} , explotando el modelo de feromonas. Después, se ejecutan opcionalmente operaciones centralizadas que no pueden ser ejecutadas por las hormigas, tales como búsqueda local, o activación de información global para mejorar el modelo de feromonas. Finalmente, antes de que inicie la siguiente iteración, algunas soluciones son empleadas para realizar el proceso de actualización del modelo de feromonas. A continuación se describe este algoritmo general con mayor detalle.

- *InicializarFeromonas()*. Al inicio del algoritmo, los valores de las feromonas son inicializados haciendo un muestreo aleatorio del espacio de búsqueda.
- *ConstruirSoluciones()*. Esta actividad involucra la liberación de una colonia independiente de hormigas artificiales para generar incrementalmente nuevas soluciones. Este es uno de los dos componentes principales de todo OCH. Se emplea un mecanismo heurístico, es decir, se utiliza el conocimiento almacenado en el modelo de feromonas con el objeto de construir nuevas soluciones.

Algoritmo 3.1 Marco general de la metaheurística OCH

```
1:  $t \leftarrow 0$ 
2: InicializarFeromonas()
3: while  $t < t_{max}$  do
4:   ConstruirSoluciones()
5:   AccionesDemonio() {Opcional}
6:   ActualizarFeromonas()
7:    $t \leftarrow t + 1$ 
```

Cada hormiga inicia el proceso de construcción con una solución parcial vacía s . Posteriormente, en cada paso de la construcción, la solución s es extendida añadiendo un componente factible tomando una decisión estocástica local.

- *AccionesDemonio*(). Dado que las hormigas por sí solas no pueden llevar a cabo algunas labores centralizadas, muchos OCHs incluyen algunas operaciones específicas denominadas **acciones demonio**. Un ejemplo clásico de este tipo de actividades es la activación de buscadores locales con el fin de mejorar las soluciones construidas en el paso anterior. La ejecución de este tipo de acciones es opcional.
- *ActualizarFeromonas*(). La experiencia adquirida por la colonia en cada iteración es considerada en esta actividad. El objetivo de este mecanismo es incrementar los valores de feromona en aquellas soluciones de mayor calidad. Como en la naturaleza, un proceso de **evaporación de feromona** toma lugar (las implementaciones usuales de este proceso decrementan el valor de feromona de las soluciones). Desde un punto de vista práctico, la evaporación de feromona es necesaria para evitar la convergencia prematura del algoritmo hacia regiones subóptimas. Gracias a este efecto de evaporación, se implementa una forma útil para «olvidar» soluciones y así favorecer la exploración de nuevas áreas en el espacio de búsqueda.

Dada la descripción anterior del esquema general, es claro que un algoritmo basado en OCH puede ser aplicado a cualquier modelo de problema de optimización siempre y cuando se pueda definir lo que a continuación Bonabeau et al. [24] indican:

1. Una representación apropiada del problema que permita a las hormigas generar incrementalmente nuevas soluciones haciendo uso del conocimiento guardado en el modelo de feromonas mediante reglas de transición probabilística e información heurística local.
2. Un método de satisfacción de restricciones que fomente la generación de soluciones factibles.
3. Reglas de actualización de feromonas (reglas de aprendizaje) que especifiquen cómo actualizar el modelo de feromonas.

En las siguientes dos secciones se presentan diferentes variantes de OCH para optimización mono- y multi-objetivo tanto para problemas combinatorios como continuos.

3.3. OCHs para optimización mono-objetivo

El comportamiento de las hormigas debido a la búsqueda de alimento ha inspirado toda una nueva gama de propuestas de algoritmos de optimización denominados OCH, siendo Dorigo [55] el primero en proponer esta metaheurística en su tesis doctoral bajo el nombre *Ant System* [35, 59, 58]. Debido a la inspiración biológica y a experimentos como el doble puente, OCH tuvo sus orígenes en la resolución de problemas de optimización combinatoria, específicamente el problema del viajero (TSP, por sus siglas en inglés) [8, 115, 156]. Los resultados de AS en el TSP fueron inicialmente alentadores, pero no pudieron superar a los algoritmos del estado del arte de esa época. En consecuencia, una cantidad sustancial de investigación fue realizada en OCH para extender AS y ofrecer algoritmos más competitivos.

Por otra parte, debido al éxito posterior de las variantes de AS y su extensión para resolver diferentes tipos de problemas combinatorios, hubo un gran interés por extrapolar OCH para resolver problemas de optimización continua. El trabajo de Bilchev y Parmee [19] fue el primer OCH para espacios de búsqueda continuos, donde la idea principal fue realizar una discretización del espacio continuo para que las hormigas pudieran realizar movimientos aleatorios a partir del nido con el objeto de alcanzar el punto óptimo (en el mejor caso). De acuerdo con Leguizamón y Coello [127], existen muchas propuestas de OCH para espacios continuos, sin embargo, ninguna de ellas es considerada unificadora además de que Dorigo y Stützle [62] consideran que ninguno de estos algoritmos cumplen totalmente con el marco de un OCH. No obstante, una propuesta que posteriormente puede ser considerada unificadora [127] fue realizada por Socha y Dorigo [171].

En las siguientes subsecciones se describirán algunas propuestas de OCH para optimización combinatoria y continua.

3.3.1. Ant System

El primer OCH propuesto en la literatura fue el *Ant System* (AS). De hecho, AS fue originalmente un conjunto de tres algoritmos llamados *ant-cycle*, *ant-density* y *ant-quantity*. Estos algoritmos fueron propuestos por Dorigo en su tesis doctoral [55] y fueron probados usando pequeñas instancias del TSP con hasta 75 ciudades.

De acuerdo con Dorigo y Stützle [62], mientras que en *ant-density* y *ant-quantity* las hormigas actualizan las feromonas directamente después de moverse de una ciudad a otra adyacente, en *ant-cycle* la actualización de las feromonas es hecha sólo después de que todas las hormigas han construido los recorridos y la cantidad de feromona depositada por cada hormiga ha sido establecida en función de la calidad del recorrido.

Dado que *ant-cycle* se desempeña mejor que las otras dos variantes, es simplemente llamado AS y las otras dos no han sido estudiadas a mayor profundidad.

El objetivo del TSP es encontrar un recorrido cerrado de longitud mínima que conecte n ciudades. Una ciudad debe ser visitada una y sólo una vez, es decir se busca un ciclo hamiltoniano [88]. Sea d_{ij} la distancia entre la ciudad i y la j . El problema puede ser definido en un espacio euclidiano, para lo cual se requieren las coordenadas de las ciudades, o generalmente usando un grafo $G = (N, E)$, donde N es el conjunto de nodos que representan las ciudades y las conexiones entre ellas se agrupan en el conjunto E de aristas.

En AS, las hormigas construyen soluciones para el TSP moviéndose en el grafo del problema de una ciudad a otra hasta que se complete el recorrido. En cada iteración $t < t_{max}$ (donde t_{max} es el máximo número de iteraciones) del algoritmo, cada hormiga $k = 1, \dots, m$ construye un recorrido ejecutando $n = |N|$ pasos en los cuales se aplica una regla de probabilidad de transición.

De acuerdo con Dorigo et al. [61], para cada hormiga, la transición de una ciudad i a otra ciudad j en la iteración t depende de lo siguiente:

1. Cada hormiga k cuenta con una memoria (llamada **lista tabú**) que le indica el conjunto de ciudades J_i^k que aún debe visitar cuando está en la ciudad i . A través de la explotación de J_i^k , la hormiga k puede evitar visitar una ciudad más de una vez.
2. Un valor heurístico denominado $\eta_{ij} = 1/d_{ij}$ que determina la preferencia de elegir la ciudad j estando en la ciudad i .
3. La cantidad de rastro de feromona $\tau_{ij}(t)$ en la arista $e_{ij} \in E$ que conecta las ciudades i y j . Este valor varía con respecto al tiempo (iteraciones) porque trata de expresar la **preferencia aprendida** de escoger la ciudad j estando en la ciudad i .

La **regla aleatoria proporcional de transición**, es decir, la probabilidad de que la hormiga k vaya de la ciudad i a la ciudad j mientras construye su t -ésimo recorrido se define como sigue:

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in J_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta} \quad (3.1)$$

donde α y β son parámetros del algoritmo que controlan el peso específico de la intensidad del rastro de feromona $\tau_{ij}(t)$ y el valor heurístico η_{ij} . Conforme $\alpha \rightarrow 0$, las ciudades más cercanas tienen mayor probabilidad de ser seleccionadas (similar a lo que haría un algoritmo glotón), mientras que si $\beta \rightarrow 0$, se harán rápidas selecciones de recorrido que posiblemente no sean óptimos. Es evidente entonces que debe haber un balance entre estos parámetros.

Después de haber completado la construcción de un recorrido, cada hormiga k deja un rastro de feromona $\Delta\tau_{ij}^k(t)$ en cada arista e_{ij} que ha empleado y tal valor está en función de qué tan bueno ha sido el recorrido. Este valor se define como sigue:

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q/L^k(t) & \text{si } e_{ij} \in T^k(t) \\ 0 & \text{si } e_{ij} \notin T^k(t) \end{cases} \quad (3.2)$$

donde $T^k(t)$ es el recorrido hecho por la hormiga k en la iteración t , $L^k(t)$ es su longitud, y Q es un parámetro que, en el mejor de los casos, debe ser aproximadamente del mismo orden de la longitud de un recorrido óptimo.

AS no se podría desempeñar correctamente sin un método de evaporación de feromonas. El objetivo de reducir los valores de feromona es el promover la habilidad de exploración del espacio de búsqueda y así tratar de evitar la convergencia prematura. El decaimiento del rastro de feromona se implementa mediante la introducción de un coeficiente de evaporación $\rho \in [0, 1)$. Con base en esto, la regla de actualización de feromonas (aplicada a todas las aristas) está dada por la siguiente fórmula:

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (3.3)$$

donde $\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t)$ y m es el número de hormigas. Al principio del algoritmo, se presupone que todas las aristas tienen asociado un valor de feromona pequeño τ_0 .

En el algoritmo 3.2 se presenta un esquema de alto nivel del AS junto con los valores de los parámetros usados por Dorigo en [58, 60]. De acuerdo con Dorigo y Stützle [62], la complejidad computacional de AS es $\mathcal{O}(t \cdot n^2 \cdot m)$, donde t es el número de iteraciones.

AS fue comparado con respecto a: recocido simulado [1, 29], búsqueda tabú [81, 79, 80] y un algoritmo genético (AG) [103] en instancias del TSP usando desde 30 hasta 75 ciudades. Los resultados mostrados por Dorigo et al. [58] fueron interesantes y decepcionantes al mismo tiempo. En la instancia con 30 ciudades, AS pudo superar al AG y fue competitivo con respecto a las dos metaheurísticas restantes. No obstante, para instancias más complejas, AS nunca obtuvo la mejor solución conocida.

En la siguiente sección se describe una versión extendida de AS que emplea un método simple de búsqueda local y presenta resultados mejores.

3.3.2. Ant Colony System

Dorigo et al. propusieron una mejora a AS denominada *Ant Colony System* (ACS) [54, 57]. ACS está fundamentado en cuatro diferencias principales con respecto a AS: una regla diferente de transición, una regla diferente de actualización de rastros de feromona, el uso de actualizaciones locales de feromona con el objeto de favorecer la exploración, y el empleo de una lista de candidatos para restringir la decisión de la siguiente ciudad a visitar.

Algoritmo 3.2 Descripción de alto nivel de AS para el TSP

-
- 1: $\alpha \leftarrow 1, \beta \leftarrow 5, \rho \leftarrow 0.5, m \leftarrow n, Q \leftarrow 100$ y $\tau_0 = 10^{-6}$
 - 2: **for all** $e_{ij} \in E$ **do**
 - 3: $\tau_{ij}(0) \leftarrow \tau_0$
 - 4: **for** $k = 1$ hasta m **do**
 - 5: Colocar la hormiga k en una ciudad aleatoria
 - 6: Sea T^+ el camino más corto al inicio y L^+ su longitud
 - 7: **for** $t = 1$ hasta t_{max} **do**
 - 8: **for** $k = 1$ hasta m **do**
 - 9: Construir el recorrido $T^k(t)$ escogiendo $n - 1$ aristas mediante la regla de transición $p_{ij}^k(t)$ (ecuación (3.1))
 - 10: **for** $k = 1$ hasta m **do**
 - 11: Calcular $L^k(t)$ para $T^k(t)$
 - 12: **if** se ha encontrado un recorrido mejor que T^+ **then**
 - 13: Actualizar T^+ y L^+
 - 14: **for all** $e_{ij} \in E$ **do**
 - 15: Actualizar rastros de feromona mediante la ecuación (3.3)
-

La Regla de transición fue modificada para permitir explícitamente la exploración. Una hormiga k en la ciudad i escoge la siguiente ciudad j con la siguiente regla:

$$j = \begin{cases} \arg \max_{u \in J_i^k} \{ \tau_{iu}(t) \cdot [\eta_{iu}]^\beta \} & \text{si } q \leq q_0 \\ J & \text{si } q > q_0 \end{cases} \quad (3.4)$$

donde q es una variable aleatoria uniformemente distribuida sobre el rango $[0, 1]$, $q_0 \in [0, 1]$ es un parámetro dado por el usuario, y $J \in J_i^k$ es una ciudad elegida con probabilidad:

$$p_{iJ}^k(t) = \frac{[\tau_{iJ}] \cdot [\eta_{iJ}]^\beta}{\sum_{l \in J_i^k} [\tau_{il}] \cdot [\eta_{il}]^\beta} \quad (3.5)$$

la cual es muy similar a la probabilidad de transición de AS. Cuando $q \leq q_0$, se favorece la explotación del conocimiento disponible acerca del problema, mientras que cuando $q > q_0$, se estimula la exploración.

Con respecto a la actualización de los rastros de feromona, en AS todas las hormigas estaban habilitadas para depositar tal sustancia una vez completado su recorrido. En contraste, ACS sólo permite que la hormiga que haya generado el mejor recorrido pueda modificar globalmente los valores de feromona en las aristas. Por tanto, las hormigas son incitadas a buscar caminos en la vecindad de la mejor solución encontrada hasta el momento. La regla de actualización es la siguiente:

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}(t) \quad (3.6)$$

donde las parejas (i, j) corresponden con las aristas del mejor tour conocido T^+ hasta el momento, ρ es el parámetro que controla el decaimiento y $\Delta\tau_{ij}(t) = 1/L^+$,

siendo L^+ la longitud de T^+ . El procedimiento anterior únicamente permite reforzar globalmente el mejor recorrido conocido. Sin embargo, con el fin de que pueda surgir otra solución, se permiten realizar actualizaciones locales.

La actualización local de feromona se realiza de la siguiente manera: cuando una hormiga k se encuentra en la ciudad i y selecciona la ciudad $j \in J_i^k$, la concentración de feromona de la arista e_{ij} se actualiza mediante la fórmula:

$$\tau_{ij}(t) = (1 - \rho)\tau_{ij}(t) + \rho\tau_0 \quad (3.7)$$

donde τ_0 se mantiene constante. Se encontró experimentalmente que $\tau_0 = (n \cdot L_{nn})^{-1}$ ofrece buenos resultados siempre y cuando L_{nn} sea la longitud de un recorrido producido por la heurística del vecino más cercano y n el número de ciudades. Cuando una hormiga visita una arista, la aplicación de la regla local disminuye su valor de feromona. Esto provoca que las aristas visitadas sean cada vez menos atractivas y de manera indirecta impulsa a las hormigas a visitar las aristas no exploradas.

La lista de candidatos es una estructura de datos común empleada en la solución de instancias grandes del TSP. Esta lista guarda un conjunto de ciudades preferidas para visitar desde la ciudad i . En lugar de examinar todas las posibilidades, las ciudades no visitadas en la lista se examinan primeramente, y sólo cuando todas las ciudades de la lista han sido probadas, otras pueden ser tomadas en cuenta.

Otro aspecto integrado a ACS para la resolución de instancias grandes del TSP es la incorporación de un método de búsqueda local. Dorigo y Gambardella [54] incorporan este mecanismo para mejorar iterativamente una solución. Los autores decidieron emplear el buscador local **3-opt** [132].

ACS fue comparado con el algoritmo de red elástica [64], los mapas auto-organizados, recocido simulado, un algoritmo genético y programación evolutiva [71]. Las instancias empleadas fueron de 50 ciudades. ACS producía los mejores resultados en la mayoría de los casos (para más información al respecto se recomienda al lector revisar la obra de Bonabeau et al. [24]).

3.3.3. Metáfora OCH para espacios de diseño continuos

La aplicación de OCH en problemas de optimización continua no es del todo directa. Sin embargo, debido a su gran éxito en problemas de optimización combinatoria, han surgido diversas propuestas para problemas continuos. Bilchev y Parmee [19] propusieron el primer OCH para espacios de búsqueda continuos. La principal dificultad afrontada fue cómo modelar un espacio de búsqueda continuo usando estructuras discretas. Los autores resolvieron esto representando un número finito de direcciones de búsqueda como vectores que parten de un punto base (nido). Con el fin de cubrir potencialmente todo el espacio continuo, estos vectores evolucionan con respecto al tiempo de acuerdo con la aptitud de las hormigas (ver figura 3.2(a)).

En el algoritmo 3.3 se muestra la estructura del algoritmo de colonias de hormigas (ACA, por sus siglas en inglés). Antes de la ejecución del algoritmo se necesita que el usuario determine la localización del nido (Bilchev y Parmee recomiendan hacer

Algoritmo 3.3 Algoritmo de colonias de hormigas para espacios continuos [19]

Entrada: Localización del nido, valor del radio de búsqueda R , máximo número de iteraciones T

```

1:  $t \leftarrow 0$ 
2: Inicializar  $A(t)$ 
3: Evaluar  $A(t)$ 
4: while  $t < T$  do
5:    $t \leftarrow t + 1$ 
6:   AgregarRastro  $A(t)$ 
7:   EnviarHormigas  $A(t)$ 
8:   Evaluar  $A(t)$ 
9:   Evaporar  $A(t)$ 

```

uso de alguna heurística para encontrar un punto prometedor) y se define un radio de búsqueda R , el cual determina la extensión del subespacio a ser considerada en cada iteración. En la línea 2, se envían hormigas en varias direcciones en un radio no mayor que R (ver figura 3.2(b)) para que posteriormente **Evaluar** $A(t)$ evalúe las soluciones generadas con la función objetivo. La función **AgregarRastro** $A(t)$ añade una cantidad proporcional de feromonas a las direcciones de búsqueda de acuerdo con la calidad de soluciones halladas. Por su parte, en la línea 7, **EnviarHormigas** $A(t)$ envía a las hormigas en ciertas direcciones mediante selección de ruleta [82]. Esta selección toma en cuenta la cantidad de feromona en los vectores y realiza un salto aleatorio de la localización de la mejor hormiga que ha seleccionado la misma dirección anteriormente. **Evaporar** $A(t)$ decrementa el nivel del rastro de feromona. El salto aleatorio es implementado como:

$$\Delta(t, R) = R \cdot [1 - r^{(1-t/T)^b}] \quad (3.8)$$

donde R es el radio de búsqueda, r es un número aleatorio en el rango $[0, 1]$, T es el máximo número de iteraciones, y b es un parámetro del algoritmo que determina el grado de no uniformidad. $\Delta(t, R)$ regresa un valor en el rango $[0, R]$ tal que la probabilidad de que $\Delta(t, R)$ esté cerca de 0 se incrementa conforme t se incrementa.

Si ciertas direcciones no resultan en una mejora, ya no participan en el proceso de agregación de rastro de feromona y el proceso de evaporación las deja fuera de toda consideración.

Bilchev y Parmee probaron el algoritmo propuesto en un problema de diseño de ingeniería altamente restringido. El dominio del problema involucra el diseño preliminar del fuselaje y la definición de la trayectoria de vuelo para un cohete alado que alcanzará su órbita antes de regresar a la atmósfera para un aterrizaje convencional. El problema es altamente sensible a cinco restricciones no lineales y se centra en minimizar el peso en el vacío del vehículo a través de un espacio de siete variables continuas y una discreta.

El algoritmo demostró, debido a las dinámicas de la colonia de hormigas, control

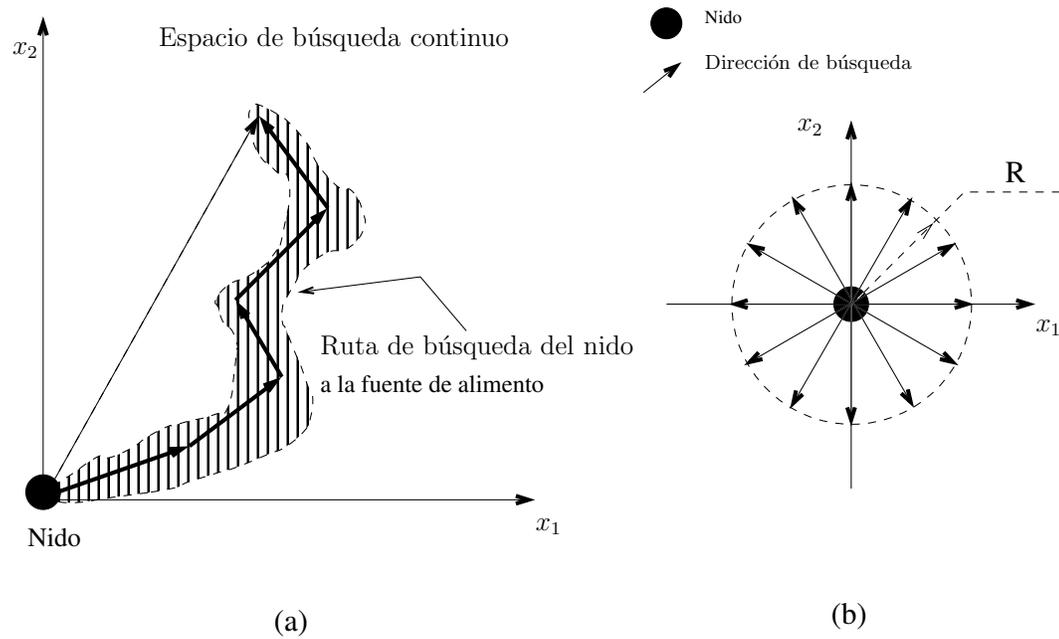


Figura 3.2: (a) Ejemplo de la discretización del espacio de búsqueda empleando vectores de dirección. (b) Modelo del vecindario de nido en un espacio bidimensional. Cada vector evoluciona con respecto al tiempo guiado por la aptitud de las soluciones halladas por las hormigas. El radio de búsqueda máximo es R .

sobre el balance de exploración y explotación del espacio de búsqueda. Los autores descubrieron diversas ventajas al usar ACA para encontrar buenas soluciones factibles:

- Muchas direcciones de búsqueda pueden ser consideradas en paralelo.
- Se puede combinar fácilmente con otras técnicas de búsqueda.
- Tiene un comportamiento mejor que un buscador local si el espacio de búsqueda contiene numerosos centros de atracción.

3.3.4. $\text{ACO}_{\mathbb{R}}$

En la literatura especializada, existen muchas variantes de algoritmos basados en OCH para espacios continuos, aunque ninguna es considerada como un estándar. De acuerdo con Leguizamón y Coello [127], la única propuesta que podría ser posteriormente el estándar de OCH en dominios continuos es el trabajo de Socha y Dorigo [171]. Su propuesta es denominada optimización basada en colonias de hormigas para dominios continuos ($\text{ACO}_{\mathbb{R}}$).

De acuerdo con los autores, $\text{ACO}_{\mathbb{R}}$ es una extensión directa de la metaheurística OCH que no requiere ningún cambio relevante en su estructura básica conceptual. La idea fundamental detrás de $\text{ACO}_{\mathbb{R}}$ es el cambio entre el uso de distribuciones discretas de probabilidad y el de funciones de densidad de probabilidad (FDPs) en el modelo de feromonas. En $\text{ACO}_{\mathbb{R}}$ en lugar de elegir aristas para completar un recorrido como en AS y ACS, se muestrea una FDP. La FDP más comúnmente usada es la función gaussiana, empero, a través de una sola FDP gaussiana no se podrían modelar zonas disjuntas del espacio de búsqueda. En consecuencia, los autores decidieron emplear una FDP con k funciones gaussianas por cada una de las n variables de decisión. Una FDP multimodal gaussiana (FDP-G) se define como sigue:

$$G^i(x) = \sum_{j=1}^k w_j g_j^i(x) = \sum_{j=1}^k w_j \frac{1}{\sigma_j^i \sqrt{2\pi}} \cdot e^{-\frac{(x-\mu_j^i)^2}{2\sigma_j^i{}^2}} \quad (3.9)$$

donde $i = 1, \dots, n$. Cada FDP-G depende de tres vectores de parámetros: \vec{w} es el vector de pesos, $\vec{\mu}^i$ es el vector de medias, y $\vec{\sigma}^i$ es el vector de desviaciones estándar. Un ejemplo de cómo luciría tal FDP-G se presenta en la figura 3.3.

El modelo de feromonas de $\text{ACO}_{\mathbb{R}}$ (figura 3.4) mantiene en un archivo \mathcal{T} , un registro de las k mejores soluciones encontradas hasta el momento. Cada solución j en \mathcal{T} guarda un vector de decisión $s_j = (s_1, \dots, s_n)^T$, un valor objetivo $u(s_j)$ y un peso w_j . Las soluciones son ordenadas por su calidad, v.g., $u(s_1) \leq u(s_2) \leq \dots \leq u(s_k)$, para un problema de minimización. Adicionalmente, cada solución tiene asociado un peso w_i proporcional a su calidad, es decir, $w_1 \geq w_2 \geq \dots \geq w_k$.

Las soluciones en \mathcal{T} son usadas para calcular los valores de los vectores \vec{w} , $\vec{\mu}^i$ y $\vec{\sigma}^i$. La complejidad de cada G^i depende de las k soluciones almacenadas en el archivo de feromonas. Para cada dimensión $i = 1, \dots, n$ hay una FPG-KG diferente. Es necesario

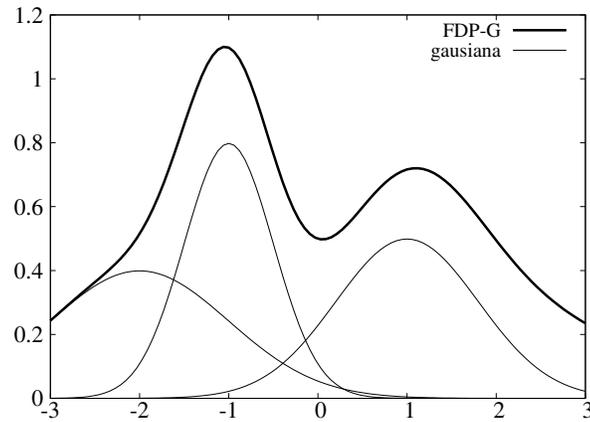


Figura 3.3: Ejemplo de cuatro FDPs gaussianas y su superposición para formar una FDP multimodal gaussianas.

s_1	s_1^1	s_1^2	$\cdot \cdot \cdot$	s_1^i	$\cdot \cdot \cdot$	s_1^n	$u(s_1)$	w_1
s_2	s_2^1	s_2^2	$\cdot \cdot \cdot$	s_2^i	$\cdot \cdot \cdot$	s_2^n	$u(s_2)$	w_2
	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot
	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot
	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot
s_j	s_j^1	s_j^2	$\cdot \cdot \cdot$	s_j^i	$\cdot \cdot \cdot$	s_j^n	$u(s_j)$	w_j
	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot
	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot
	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot
s_k	s_k^1	s_k^2	$\cdot \cdot \cdot$	s_k^i	$\cdot \cdot \cdot$	s_k^n	$u(s_k)$	w_k
	G^1	G^2		G^i		G^n		

Figura 3.4: Modelo de feromonas de ACO_R. Las soluciones se encuentran ordenadas con respecto a la calidad de la función objetivo $u(s_1) \leq u(s_2) \leq \dots \leq u(s_k)$, tomando en cuenta un problema de minimización. Además, cada solución i tiene asociado un peso w_i proporcional a su calidad. Por tanto, $w_1 \geq w_2 \geq \dots \geq w_k$. Cada variable de decisión se modela empleando una FDP-G.

mencionar que para construir G^i , sólo es necesaria la i -ésima variable de decisión de cada solución.

Los elementos de $\vec{\mu}^i$ para G^i corresponden con las i -ésimas variables de todas las soluciones:

$$\vec{\mu}^i = \{\mu_1^i, \dots, \mu_k^i\} = \{s_1^i, \dots, s_k^i\} \quad (3.10)$$

Para el cálculo de cada w_j se involucra el uso de un parámetro $q > 0$ que controla la diversificación del proceso de búsqueda. Cada w_j está definido como un valor de una función gaussiana de acuerdo con la siguiente fórmula:

$$w_j = \frac{1}{qk\sqrt{2\pi}} \cdot e^{-\frac{(\text{rank}(s_j)-1)^2}{2q^2k^2}} \quad (3.11)$$

donde $\text{rank}(\cdot)$ entrega el rango de la solución dentro del archivo de feromonas de acuerdo con el orden establecido. Conforme $q \rightarrow 0$, se prefieren las mejores soluciones, y cuando su valor es grande, los pesos tienden a estar uniformemente distribuidos.

Cada hormiga $i = 1, \dots, M$ realiza n pasos de construcción para generar una nueva solución. En el paso constructivo i , sólo la información acerca de la i -ésima solución es empleada con el fin de producir dinámicamente G^i .

El proceso de muestreo en el paso i es completado de la siguiente forma. Primeramente, los elementos de \vec{w} son calculados usando la ecuación (3.11). Una función gaussiana es seleccionado de entre los k que componen la FDP-G. La probabilidad de elegir la r -ésima función gaussiana está dada por:

$$p_r = \frac{w_r}{\sum_{l=1}^k w_l} \quad (3.12)$$

En el paso de construcción i debe conocerse exclusivamente la desviación estándar para la r -ésima función gaussiana elegido previamente. Por ende, no es necesario calcular el vector de desviaciones en su totalidad. El valor de la desviación estándar σ_r^i es calculada como sigue:

$$\sigma_r^i = \xi \sum_{j=1}^k \frac{|s_j^i - s_r^i|}{k-1} \quad (3.13)$$

donde $\xi > 0$ es un parámetro del algoritmo que controla la forma en que se usa la memoria a largo plazo, v.g., la velocidad de convergencia. Este parámetro es similar a la tasa de evaporación de feromona. Cuando ξ es grande, la velocidad de convergencia es más lenta y en caso de que su valor sea cercano a cero, la velocidad de convergencia se incrementa.

Posteriormente, la función gaussiana elegida es muestreado usando, por ejemplo, el método de Box-Müller [25]. El proceso anterior es repetido para cada dimensión con el objeto de obtener una nueva solución.

Una vez que las M hormigas han construido nuevas soluciones, se ejecuta el proceso de actualización de feromonas. El archivo de feromonas \mathcal{T} contiene las k mejores

soluciones encontradas hasta el momento, donde tal número es invariable a lo largo de todo el proceso de búsqueda. Al principio del algoritmo, \mathcal{T} se inicializa con soluciones elegidas aleatoriamente de una distribución uniforme. Sin embargo, cuando se realiza la actualización de feromonas, se debe elegir de entre las soluciones recientemente creadas aquellas que mejoren a las ya guardadas para, de esta manera, reemplazarlas. En consecuencia, el tamaño del archivo se mantendrá inalterado. Además, este proceso asegura que siempre se guarden las mejores soluciones.

ACO_R fue comparado con respecto a una gran cantidad de algoritmos que los autores clasificaron de la siguiente forma:

- Métodos de aprendizaje probabilístico - métodos que explícitamente modelan y muestrean una distribución de probabilidad.
- Métodos OCH - métodos que aseguran tomar inspiración del comportamiento de las hormigas.
- Otras metaheurísticas - originalmente pensadas para optimización combinatoria y luego adaptadas para dominios continuos.

De acuerdo con Socha y Dorigo, los resultados, producto de haber usado una gran cantidad de problemas de prueba, son competitivos. Cuando se comparó ACO_R con los métodos de aprendizaje probabilístico, obtuvo el mejor lugar en el 40 % de los problemas y en los restantes probaron que su desempeño no era significativamente peor que el de los algoritmos del estado del arte. Con respecto a los algoritmos basados en OCH, los autores prueban que ACO_R es la mejor propuesta para espacios continuos. Finalmente, al compararse contra las otras metaheurísticas, ACO_R fue sólo mejor en el 30 % de los problemas.

3.3.5. Aplicaciones en problemas mono-objetivo

En la actualidad, numerosas implementaciones exitosas de la metaheurística OCH (partiendo de AS) están disponibles y han sido aplicadas a una gran cantidad de problemas de optimización combinatoria. En la tabla 3.1 se resumen algunas de estas aplicaciones.

3.4. OCHs para optimización multi-objetivo

Los algoritmos de optimización mediante colonias de hormigas han demostrado ser estrategias efectivas de solución de problemas para una amplia gama de dominios, incluyendo problemas de optimización multi-objetivo. El enfoque de los OCHs multi-objetivo (AOMCHs) a la fecha ha sido mayormente la resolución de problemas combinatorios de optimización multi-objetivo (PCOMs). Esto es un efecto colateral claro del buen desempeño de OCH en problemas combinatorios mono-objetivo. En contraste, la comunidad de optimización evolutiva multi-objetivo se centra en mayor

Tabla 3.1: Conjunto de aplicaciones de OCH listadas de acuerdo con el tipo de problema.
 Tabla extraída de la obra de Dorigo y Stützle [62].

Tipo de problema	Nombre del problema	Autores	Año	Referencias
Enrutamiento	TSP	Dorigo et al. Dorigo y Gambardella Stützle y Hoos	1991, 1996 1997 1997, 2000	[58, 60] [54] [177, 179]
	TSP con ventanas de tiempo	López Ibáñez et al.	2009	[135]
	Ordenamiento secuencial	Gambardella et al. Gambardella et al.	2000 1999	[75] [74]
	Enrutamiento vehicular	Reimann et al. Favoretto et al.	2004 2007	[155] [69]
	Multicasting	Fuellerer et al. Hernández y Blum	2009 2009	[73] [98]
Asignación	Asignación cuadrática	Maniezzo Stützle y Hoos	1999 2000	[138] [179]
	Asignación de frecuencias	Maniezzo y Carbonaro	2000	[139]
	Horarios escolares	Socha et al.	2002, 2003	[172, 173]
	Coloración de grafos	Costa y Hertz	1997	[37]
Planificación	Planificación de proyectos	Merkle et al.	2002	[142]
	Tardanza ponderada	den Besten et al.	2000	[50]
	<i>Flow shop</i>	Merkle y Middendorf Stützle	2000 1997	[142] [178]
	<i>Open shop</i>	Rajendran, Ziegler	2004	[154]
	Secuenciamiento de automóviles	Blum Solnon	2005 2008	[20] [174]
	Subconjunto	Cobertura de conjunto	Lessing et al.	2004
Árboles de cardinalidad l		Blum y Blesa	2005	[21]
Mochilas múltiple		Leguizamón y Michalewicz	1999	[126]
Clique máximo		Solnon, Fenet	2006	[175]
Aprendizaje de máquina	Reglas de clasificación	Parpinelli et al. Martens et al. Otero et al.	2002 2006 2008	[151] [140] [148]
	Redes Bayesianas	Campos, Fernández-Luna	2002	[38, 39]
	Redes neuronales	Socha, Blum	2007	[170]
Bioinformática	Pliegue de proteínas	Shmygelska y Hoos	2005	[169]
	Acoplamiento molecular	Korb et al.	2006	[122, 123]
	Secuenciamiento de ADN	Blum et al.	2008	[23]
	Interferencia de haplotipo	Benedettini et al.	2008	[13]

medida en la solución de POMs en espacios de búsqueda continuos. Muchos de los AOMCHs tienden a ser extensiones de OCHs mono-objetivo con gran reconocimiento, tales como AS [60], ACS [54] y \mathcal{MMAS} [179].

El primer AOMCH reportado en la literatura especializada fue propuesto por Iredi et al. [107], quienes discuten diferentes alternativas para aplicar OCH a PCOMs y presentan sus resultados usando unas pocas variantes del problema de planificación bi-objetivo. A partir de entonces, una gran cantidad de estudios han probado diferentes propuestas. Los AOMCHs que se han sugerido pueden diferir en si emplean una o varias matrices de feromonas (una por cada objetivo), la cantidad de información heurística manejada, cómo se seleccionan las soluciones para depositar feromona, y si deben utilizarse una o varias colonias. Muchas combinaciones de estas posibilidades han sido estudiadas, por ejemplo, Alaya et al. [2] y López-Ibáñez et al. [136] muestran algunos de estos estudios. Para información más detallada acerca de los AOMCHs, se recomienda al lector referirse a los artículos de García-Martínez et al. [76], Angus y Woodward [5], y Leguizamón y Coello [127].

Finalmente, en lo que respecta a AOMCHs para resolver problemas multi-objetivo en espacios de búsqueda continuos, su extensión ha sido muy poco estudiada a comparación de su contraparte mono-objetivo. En efecto, Leguizamón y Coello [127] mencionan en su estudio únicamente dos propuestas de este tipo (que serán descritas en las secciones 3.4.3 y 3.4.4).

3.4.1. CPACO

El algoritmo *Population-based ACO algorithm* [90] (PACO) fue extendido por Angus [3] por medio de la integración de un esquema de reemplazo de población usando *crowding* para el TSP multiobjetivo, nombrándolo CPACO. Esta propuesta usa una matriz de feromonas y valores individuales heurísticos para cada objetivo. La idea básica del esquema de agrupamiento fue tomada del área de computación evolutiva. En cada iteración, la matriz de feromonas es actualizada de la siguiente manera: (1) el conjunto de soluciones en la población se clasifica usando el esquema de ordenamiento no dominado del NSGA-II [44], (2) todos los elementos de la matriz de feromonas son reinicializados a un valor τ_{init} , y (3) las soluciones clasificadas en la población producen un sesgo de actualización en la reinicialización de las feromonas usando el inverso del rango asignado. Por tanto, cuanto mejor sea el rango de una solución, habrá una mayor concentración de feromona en la entrada correspondiente en la matriz. La manera en la que CPACO combina la matriz de feromonas y los diversos valores heurísticos (uno por cada objetivo) sigue la propuesta de Barán y Schaerer [12].

CPACO fue estudiado en un conjunto de instancias de TSP con dos y cuatro objetivos. El estudio comparativo de CPACO con respecto a PACO mostró un desempeño mejorado de CPACO en los problemas bi-objetivo. Aunque los resultados no fueron tan buenos como para el primer conjunto de instancias, CPACO mostró un desempeño aceptable en los problemas con cuatro objetivos [127]. Los autores indicaron que la

actualización de feromonas y el mecanismo de recopilación de información heurística requieren mayor investigación.

3.4.2. MO-ant

Liu et al. [134] propusieron un algoritmo basado en OCH denominado (MO-ant) con el fin de generar frentes de Pareto para el problema multi-objetivo de la colocación de instalaciones de servicios de emergencia (ISEs). El concepto de jerarquización de Pareto [82] fue empleado para determinar cuáles eran las soluciones mejor calificadas para el problema. El área geográfica de estudio fue representada mediante un mapa reticular donde las hormigas debían encontrar los mejores sitios para colocar las ISEs. MO-ant no usa información heurística y gestiona una sola matriz de feromonas, en la cual cada ubicación representa la preferencia de colocar una ISE en la posición correspondiente en el área geográfica real.

Además, este algoritmo aplica en cada iteración una búsqueda local de dos fases donde se clasifican las soluciones de acuerdo con la jerarquización de Pareto. Durante la primera fase del buscador local se aplica la **búsqueda aleatoria de vecindario** (NRS, por sus siglas en inglés) que estimula a las hormigas a moverse aleatoriamente de una celda a otra dentro de una cierta distancia. Posteriormente, todas las soluciones vuelven a ser evaluadas y se obtienen las jerarquías de Pareto. La segunda fase consiste en la aplicación de la **búsqueda adaptable de enumeración de vecindario** (AENS, por sus siglas en inglés) a la primera solución en el conjunto previamente clasificado. AENS tiene como finalidad encontrar una mejor posición que la actual para cada hormiga en la colonia considerando todas las celdas dentro de una cierta distancia. Si la suma total de los objetivos es mejorada, la hormiga se mueve a la celda que produjo tal mejora. Este procedimiento es aplicado hasta que no se encuentren más cambios. Finalmente, las soluciones vuelven a ser procesadas usando la jerarquización de Pareto.

La matriz de feromonas es actualizada tomando en cuenta el rango de las soluciones halladas. De acuerdo con Leguizamón y Coello [127], la búsqueda del frente de Pareto es mayormente guiada por el procedimiento de búsqueda local así como también por los valores de las feromonas.

MO-ant fue aplicado en un problema del mundo real: la ubicación óptima de estaciones de bomberos en Singapur (considerando tres objetivos). Los resultados obtenidos fueron comparados con respecto a un conjunto de 10,000 soluciones generadas aleatoriamente y clasificadas de acuerdo con la optimalidad de Pareto.

3.4.3. PACO-MOFO

La metaheurística *Population-based ACO for Multi-objective Function Optimisation* (PACO-MOFO) fue propuesta por Angus [4]. PACO-MOFO es una extensión de trabajos previamente publicados los cuales incluyen $ACO_{\mathbb{R}}$ [171] y CPACO [91, 90]. Uno de los aspectos fundamentales de PACO-MOFO es el uso de técnicas de nicho como agrupamiento [40, 137] y compartición de aptitud [83] con el objeto de mantener

diversidad en las soluciones. Por otra parte, PACO-MOFO se puede definir como un método de articulación de preferencias a posteriori (ver sección 2.6 en la página 22).

PACO-MOFO reusa los componentes de $ACO_{\mathbb{R}}$ y CPACO. Dado el dominio del problema, la matriz de feromonas empleada es similar a la de $ACO_{\mathbb{R}}$. Además, una población discreta es utilizada y modificada a través de operaciones de comparación basadas en distancia de agrupamiento. La operación de reemplazo basada en agrupamiento asegura que la población retenga diversas soluciones del espacio objetivo, mientras que la selección con compartición de aptitud fomenta un muestreo uniforme del espacio objetivo.

Como se mencionó anteriormente, PACO-MOFO usa dos formas de creación de nichos: distancia de agrupamiento y compartición de aptitud para el reemplazo de población y como mecanismo de selección, respectivamente. La técnica de agrupamiento empleada es la selección de torneo restringida (STR) [96]. STR extrae un subconjunto de la población con el fin de comparar nuevas soluciones y así determinar la coincidencia más cercana. Si la nueva solución es mejor que la coincidencia más cercana, entonces se reemplaza; de lo contrario, se descarta la nueva solución. Dado esto, STR controla la diversidad que se encuentra en la solución. Por otra parte, PACO-MOFO usa compartición de aptitud para asegurar un muestreo balanceado de la población. Esta técnica trabaja agrupando los miembros de la población de acuerdo con su proximidad en el espacio objetivo y reduciendo su aptitud si pertenecen a una zona altamente densa de soluciones.

Los problemas seleccionados para probar la propuesta fueron Schaffer (SCH), Fonseca (FON), Poloni (POL) y Deb (DEB) [32]; todos los problemas tenían dos variables de decisión y dos objetivos. Los autores compararon PACO-MOFO únicamente con respecto a NSGA-II [44]. Salvo para el caso de SCH, en los tres problemas restantes la diferencia de resultados entre PACO-MOFO y NSGA-II no fue estadísticamente significativa.

3.4.4. MOACO $_{\mathbb{R}}$

García-Najera y Bullinaria [77] propusieron una extensión de $ACO_{\mathbb{R}}$ para resolver problemas de optimización multi-objetivo. Esta propuesta es denominada *Multi-Objective Ant Colony Optimizer for Continuous Domains* (MOACO $_{\mathbb{R}}$).

Puesto que al resolver POMs no existe una única solución sino un conjunto de ellas, el problema principal que afrontaron los autores fue determinar qué soluciones guardar. En $ACO_{\mathbb{R}}$, después de cada iteración se guardaban las mejores soluciones de acuerdo con el valor de la función objetivo. Sin embargo, para MOACO $_{\mathbb{R}}$ este esquema se modificó empleando el concepto de **profundidad de dominancia** [44, 197]. La profundidad de dominancia implica agrupar las soluciones a un POM en diferentes frentes de acuerdo con el nivel de dominancia de cada una, tal como se emplea en el NSGA-II. Una vez que las soluciones han sido agrupadas, MOACO $_{\mathbb{R}}$ prefiere guardar aquellas soluciones que se encuentran en el frente más exterior, es decir, aquel que presenta una mejor convergencia hacia el frente óptimo de Pareto.

Dado que se emplea un modelo de feromonas como el de $ACO_{\mathbb{R}}$, el archivo de feromonas es de tamaño constante y en caso de que al guardar las nuevas soluciones se exceda su tamaño, un mecanismo basado en información de densidad entra en acción. Este mecanismo decreta la probabilidad de una solución para ser seleccionada conforme la densidad de su vecindad aumenta. La información de densidad se calcula usando la distancia de agrupamiento [44]. $MOACO_{\mathbb{R}}$ conserva exactamente los mismos mecanismos de $ACO_{\mathbb{R}}$ para la generación de soluciones en los problemas multi-objetivo.

La propuesta fue probada usando los problemas de prueba SCH, DEB, POL, FON, Kursawe (KUR) [32] y el conjunto de problemas Zitzler-Deb-Thiele (ZDT) [194]. Se compararon resultados con los algoritmos NSGA-II [44] y el *Multi-Objective Particle Swarm Optimizer* [34] (MOPSO). Para los problemas SCH y DEB, los tres algoritmos obtuvieron resultados similares. En cuanto al problema POL, MOPSO y $MOACO_{\mathbb{R}}$ se comportaron aproximadamente igual pero no tan bien como el NSGA-II. En cuanto a los problemas FON y KUR, $MOACO_{\mathbb{R}}$ obtuvo los peores resultados. Finalmente, para los problemas del conjunto ZDT, NSGA-II y $MOACO_{\mathbb{R}}$ superaron a MOPSO en ZDT1, ZDT2 y ZDT3; mientras que en ZDT4, $MOACO_{\mathbb{R}}$ fue superado claramente al no poder converger al frente óptimo de Pareto; con respecto a ZDT6, $MOACO_{\mathbb{R}}$ obtuvo mejores resultados que los otros dos algoritmos.

3.5. Resumen

Este capítulo introdujo conceptos básicos de optimización mediante colonias de hormigas. En primera instancia, se describió la inspiración biológica que respalda la metaheurística. En este caso, el comportamiento social de ciertas clases de hormigas en búsqueda de comida da origen a un sistema descentralizado y auto-organizado que se comunica indirectamente (*stigmergy*) mediante la colocación de feromonas en el ambiente. Los rastros de feromonas dejados por las hormigas sesgan con cierta probabilidad la decisión de una hormiga que va del nido a la fuente de comida. Debido a esto, con el paso del tiempo el camino más corto a la fuente de alimento surge como una propiedad global del comportamiento social de las hormigas. Lo anterior se confirmó experimentalmente al liberar hormigas desde el nido a una fuente de alimento teniendo que cruzar un puente de dos vertientes de diferentes tamaños.

El esquema general de un OCH consta de tres procesos fundamentales: (1) la construcción incremental de soluciones por parte de cada hormiga de la colonia, (2) la actualización de los rastros de feromona en aquellas soluciones que son prometedoras junto con un esquema de evaporación de los rastros que favorezca la habilidad de exploración del espacio de búsqueda, y (3) actividades centralizadas, llamadas demonios, que no pueden ser ejecutadas individualmente por las hormigas.

El primer OCH propuesto fue el *Ant System* que estaba orientado en la solución del problema del viajero. Los resultados de AS no fueron del todo satisfactorios, aunque sí prometedores por lo que se realizaron una gran cantidad de extensiones entre las que destacan: *Ant Colony System* y *MAX-MIN Ant System*. Estas propuestas

sí entregan soluciones muy buenas, por lo que posteriormente la investigación en OCH abarcó una gran cantidad de problemas combinatorios con resultados muy satisfactorios. Además, algunos investigadores llevaron la metaheurística a la resolución de problemas continuos. En este ramo, los resultados han sido aceptables, aunque no se cuenta con una propuesta que sea del todo fiel a la inspiración original de OCH. A pesar de esto, $ACO_{\mathbb{R}}$ se erige como una propuesta muy prometedora.

Finalmente, OCH también ha participado en la resolución de problemas multi-objetivo. Como era de esperarse, ha tenido un mayor impacto en el área de problemas combinatorios multi-objetivo. No obstante, la incidencia en POMs continuos ha sido prácticamente nula. En consecuencia, la aplicación de AOMCHs a espacios de decisión continuos parece ser un área fértil de investigación.

Capítulo 4

Metaheurística iMOACO_ℝ

Este capítulo presenta el MOACO propuesto para espacios de búsqueda continuos, llamado *indicator-based Multi-Objective Ant Colony Optimizer for Continuous Search Spaces* (iMOACO_ℝ). Esta nueva metaheurística hace uso de ACO_ℝ (ver sección 3.3.4 en la página 52) como motor de búsqueda. Asimismo, iMOACO_ℝ utiliza un esquema de clasificación de soluciones basado en el indicador de desempeño R2 que permite escalar su funcionamiento en altas dimensiones en el espacio de las funciones objetivo. En la sección 4.1 se describe el algoritmo de asignación de rangos que depende de una función de utilidad y un conjunto de vectores de pesos, descritos en las secciones 4.2 y 4.3, respectivamente. La sección 4.4 tiene por objeto introducir un algoritmo de actualización de los vectores ideal y de nadir. En la sección 4.5, se describe detalladamente la integración de los diferentes componentes que permiten el funcionamiento de iMOACO_ℝ¹. Finalmente, en la sección 4.6 se provee un resumen del capítulo.

4.1. Algoritmo clasificación-R2

El esquema de ordenamiento no dominado propuesto por Deb et al. [44] para el NSGA-II, basado en dominancia de Pareto, marcó un punto importante en el desarrollo de los algoritmos evolutivos multi-objetivo. Sin embargo, se ha demostrado recientemente que los esquemas de asignación de rangos con base en dominancia de Pareto tienen un mal desempeño en problemas de optimización con muchas funciones objetivo [112]. Como consecuencia, Hernández en su tesis de maestría [99] propuso un esquema de asignación de rangos basado en el indicador R2 [28], denominado clasificación-R2².

Esta propuesta tiene por objeto producir un esquema de ordenamiento no dominado basado en un conjunto de funciones de utilidad. La idea básica yace en agrupar las soluciones que optimicen un conjunto de funciones de utilidad y colocarlas en el tope del ordenamiento de tal modo que obtengan el primer rango. Posteriormente,

¹El código fuente de iMOACO_ℝ está disponible para ser descargado en: <http://computacion.cs.cinvestav.mx/~jfalcon/imoacor.html>

²Este esquema fue publicado posteriormente por Hernández y Coello [100, 101].

tales puntos son removidos y un segundo rango es asignado de la misma manera. Este proceso continúa hasta que no haya más soluciones por clasificar. El esquema clasificación-R2, derivado de la ecuación (2.12), se define matemáticamente de la siguiente manera:

$$\text{rank}_k(A) = \bigcup_{\vec{u} \in U} u^* (\{A \setminus B_k(A)\}), \quad (4.1)$$

donde $B_k(A)$ es la unión de las soluciones con los rangos más bajos:

$$B_k(A) = \left\{ \bigcup_j \text{rank}_j(A) \mid k > 1, j \in [1, k] \right\}. \quad (4.2)$$

En el algoritmo 4.1 se muestra el esquema de asignación de rangos propuesto por Hernández y Coello en [101]. Para este algoritmo se asume que cada solución p de la población P tiene la siguiente estructura:

- $p.\vec{F}$: vector de funciones objetivo
- $p.\alpha$: valor actual de utilidad para el vector de pesos λ
- $p.u^*$: mejor valor de utilidad obtenido
- $p.rank$: rango otorgado por el algoritmo

Algoritmo 4.1 clasificación-R2 [101]

Entrada: Población P , conjunto de vectores de pesos Λ

Salida: Rangos de la población

- 1: $p.rank \leftarrow p.\alpha \leftarrow \infty \quad \forall p \in P$
 - 2: **for all** $\vec{u} \in U$ **do**
 - 3: **for all** $p \in P$ **do**
 - 4: $p.\alpha \leftarrow u_{ASF}(p.\vec{F} : \vec{0}, \vec{\lambda})$
 - 5: Ordenar P con respecto a los valores de α y L_2 en forma ascendente
 - 6: $rank \leftarrow 1$
 - 7: **for all** $p \in P$ **do**
 - 8: **if** $rank < p.rank$ **then**
 - 9: $p.rank = rank$
 - 10: $p.u^* = p.\alpha$
 - 11: $rank \leftarrow rank + 1$
-

En la línea 1 se inicializan los valores del rango y valor de utilidad de cada solución. Por cada vector de pesos $\vec{\lambda}$, se evalúan todas las soluciones en P usando la función de utilidad ASF [143] (descrita en la siguiente sección) y dicho valor se almacena en $p.\alpha$ (líneas 3 y 4). En la línea 5, las soluciones en P son ordenadas en forma creciente con respecto a dos criterios: el valor de α , y en caso de haber un empate se rompe con la

comparación de las normas L_2 de las soluciones. Finalmente, las líneas 6 a 11 realizan la asignación de rangos. De acuerdo con los autores, la complejidad computacional de este algoritmo es $\mathcal{O}(|\Lambda||P|(\log |P| + m))$.

4.2. Función de utilidad

El concepto de utilidad fue desarrollado por Von Neumann y Morgenstern en la teoría de la utilidad axiomática [183], el cual asume que el tomador de decisiones puede elegir entre las alternativas disponibles de tal forma que el nivel de satisfacción derivado sea tan alto como sea posible. Esto, por supuesto, implica que el tomador de decisiones esté consciente de las alternativas disponibles y que sea capaz de evaluarlas. Además, se asume que toda la información perteneciente a los varios niveles de aspiración de los objetivos puede ser capturada por la función de utilidad.

Matemáticamente, una función de utilidad, $u : \mathbb{R}^m \rightarrow \mathbb{R}$, es un modelo de las preferencias del tomador de decisiones que asigna a cada punto en el espacio objetivo un valor de utilidad. Es importante mencionar que una función de utilidad realmente no refleja el nivel de intensidad (psicológica) de la preferencia del tomador de decisiones. La función de utilidad sólo provee un modelo de este comportamiento.

En la literatura especializada en optimización evolutiva multi-objetivo, un tipo de funciones de utilidad escasamente usadas son las denominadas funciones escalares por logros (ASFs, por sus siglas en inglés). Este tipo de funciones fueron introducidas por Wierzbicki [187, 188, 189]. Su principal ventaja con respecto a funciones de utilidad como la métrica L_p o la métrica ponderada de Tchebycheff (MPT) [143], es que aún cuando no se conozca el vector ideal global y se emplee una aproximación dentro de la región objetivo factible, una ASF será capaz de producir soluciones (débiles) Pareto-óptimas, a diferencia de los dos tipos de funciones antes mencionadas. Además, Hernández y Coello en [101] muestran un experimento donde se hace notar que ASF tiene mejor desempeño en alta dimensionalidad que MPT y la función de intersección de límite basada en penalización (PBI, por sus siglas en inglés). Existen diferentes ASFs, pero para este trabajo se hará uso de la siguiente función:

$$u_{asf}(\vec{v} : \vec{r}, \vec{\lambda}) = \max_{i \in \{1, \dots, m\}} \left\{ \frac{|v_i - r_i|}{\lambda_i} \right\}. \quad (4.3)$$

Finalmente, con la finalidad de evitar problemas con funciones objetivo no conmensurables, es decir, objetivos con escalas de magnitud muy desiguales, se emplea la función de normalización introducida en la sección 2.3 (página 13):

$$v'_i = \frac{v_i - z_i^*}{z_i^{nad} - z_i^*}. \quad (4.4)$$

4.3. Generación de vectores de pesos

Uno de los componentes necesarios del algoritmo clasificación-R2 es la generación de un conjunto de vectores de pesos Λ para la evaluación de la función de utilidad ASF. Cada vector de pesos $\vec{\lambda}$ debe cumplir con ser una combinación convexa de valores, es decir, $\lambda_i \geq 0 \forall i = 1, \dots, m, \wedge \sum_{i=1}^m \lambda_i = 1$. En consecuencia, estos vectores describen una superficie simplex. Una característica deseable de este conjunto de vectores de pesos es que estén uniformemente distribuidos en el espacio objetivo para que así se puedan obtener diferentes soluciones en el frente óptimo de Pareto.

De acuerdo con la construcción original del algoritmo clasificación-R2, se emplea el método de diseño *Simplex-Lattice* (SLD, por sus siglas en inglés) introducido por Scheffé entre 1958 y 1963 [164, 165]. El conjunto de puntos factibles forma lo que ha sido llamado en la literatura especializada un simplex-lattice $\{m, h\}$ [164, 36], o en combinatoria una composición de h en m -partes [157], donde m es el número de objetivos y h es un parámetro de proporción. Esta estructura consiste de todas las combinaciones posibles de proporciones de cada función objetivo. Por tanto, cada coeficiente del vector de pesos toma $h + 1$ valores equitativamente espaciados entre 0 y 1, esto es:

$$\lambda_i = \epsilon, \frac{1}{h}, \frac{2}{h}, \dots, 1 \quad (4.5)$$

donde ϵ es un valor cercano a cero, usualmente 10^{-4} , empleado con el fin de evitar cancelaciones en los cálculos. De acuerdo con Scheffe [164] y Feller [70], el número de puntos factibles dentro del simplex-lattice $\{m, h\}$ está dado por el número combinatorio $C_{m-1}^{h+m-1} = \frac{(h+m-1)!}{h!(m-1)!}$.

Existen diferentes propuestas de algoritmos que implementan el método SLD. En el algoritmo 4.2 se presenta una propuesta eficiente realizada por Chasalow y Brand [167] que es empleada para los fines de este trabajo. Un aspecto importante a notar es que para contar con puntos intermedios dentro del simplex se debe cumplir que $h \geq m$ [45]. De acuerdo con los autores, la complejidad de este algoritmo es $\mathcal{O}(|\Lambda|)$.

Una de las principales desventajas de SLD yace en su incapacidad para escalar en altas dimensiones en el espacio objetivo. Puesto que la cardinalidad de Λ depende de un número combinatorio, y en ocasiones las poblaciones de los algoritmos evolutivos se relacionan con este número, aún cuando $h = m$ (p. ej., $h = 7$ resultará en $C_{7-1}^{7+7-1} = 1716$) se obtendrán números que pueden agravar el desempeño computacional del algoritmo. En la figura 4.1 se aprecia un ejemplo de la distribución de vectores con este método.

4.4. Manejo de puntos de referencia

En la sección 2.3 (página 13) se introdujeron dos puntos de referencia importantes denominados: vector ideal (\vec{z}^*) y vector de nadir (\vec{z}^{nad}). El vector ideal retiene los valores mínimos por objetivo. Por su parte, el vector de nadir está compuesto por los

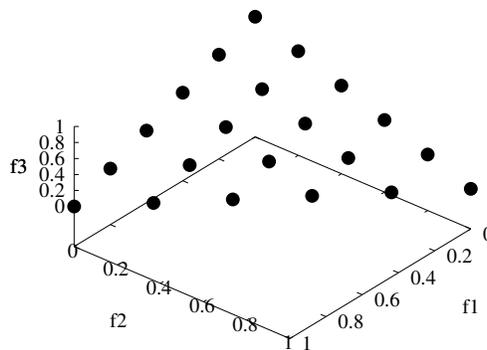
Algoritmo 4.2 Diseño Simplex-Lattice

Entrada: Dimensión m , parámetro de proporción h
Salida: Conjunto de vectores de pesos Λ

 1: Inicializar vector $\vec{x} = (x_1, \dots, x_m)$, tal que:

$$x_i \leftarrow \begin{cases} h & \text{si } i = 1 \\ 0 & \text{en otro caso} \end{cases}$$

 2: $\Lambda \leftarrow \emptyset$
 3: $j \leftarrow 1$
 4: **repeat**
 5: $\Lambda \leftarrow \Lambda \cup \{\frac{\vec{x}}{h}\}$
 6: $x_j \leftarrow x_j - 1$
 7: **if** $j < m - 1$ **then**
 8: $x_{j+1} \leftarrow h - \sum_{i=1}^j x_i$
 9: $\forall i \in \{j + 2, \dots, m\}, x_i \leftarrow 0$
 10: $j \leftarrow j + 1$
 11: **else**
 12: $x_m \leftarrow x_m + 1$
 13: $I \leftarrow \{i | x_i > 0, i = 1, \dots, m - 1\}$
 14: **if** $I \neq \emptyset$ **then**
 15: $j \leftarrow \text{máx}(I)$
 16: **until** $x_m = h$
 17: $\Lambda \leftarrow \Lambda \cup \{\frac{\vec{x}}{h}\}$
 18: **return** Λ


 Figura 4.1: Generación de vectores de pesos usando SLD con $h = 5$ y $m = 3$.

valores máximos de los objetivos obtenidos de las soluciones en el frente óptimo de Pareto. Estos vectores son especialmente importantes en el proceso de normalización de los vectores objetivo, realizada con la finalidad de tener una misma escala de medición.

Sin embargo, el cálculo de \bar{z}^{nad} es particularmente difícil para $m > 2$ [143] funciones objetivo. En consecuencia, Hernández y Coello [101] propusieron un esquema de actualización de estos puntos empleando información estadística de poblaciones de soluciones pasadas. La idea principal es monitorear el vector de nadir de las poblaciones que actúan como padres en cada generación, determinando qué tan cercanos están los individuos del verdadero frente de Pareto. Una alta varianza significa que las soluciones están muy alejadas, lo cual sesga fuertemente la localización del punto. Una varianza pequeña indica que las soluciones están cercanas y, por ende, debe realizarse pequeños movimientos.

En el algoritmo 4.3 se muestra el esquema de actualización de los puntos de referencia. Este algoritmo requiere de una estructura de datos denominada RECORD que guarda los vectores de nadir obtenidos de las poblaciones durante cierto número de generaciones y a partir de estos registros calcula las varianzas por cada objetivo. Además, se necesita un parámetro α que determina un umbral de varianzas para el vector \bar{z}^{nad} , y un parámetro ϵ , el cual es el umbral de tolerancia. La complejidad de este algoritmo es $\mathcal{O}(|P|m)$ de acuerdo con los autores.

Algoritmo 4.3 Actualización de puntos de referencia

Entrada: \bar{z}^{min} , \bar{z}^{max} , población P , número de objetivos m

Salida: \bar{z}^{min} , \bar{z}^{max}

- 1: Actualizar \bar{z}^* y \bar{z}^{nad} usando definiciones 2.3.1 y 2.3.3
 - 2: $z_i^{min} \leftarrow \min z_i^{min}, z_i^* \forall i \in \{1, \dots, m\}$
 - 3: Guardar \bar{z}^{nad} en RECORD
 - 4: Obtener vector de varianzas $\vec{v} \in \mathbb{R}^m$ para \bar{z}^{nad} de RECORD
 - 5: **if** $\max_{j \in \{1, \dots, m\}} v_j > \alpha$ **then**
 - 6: $z_i^{max} \leftarrow \max_{j \in \{1, \dots, m\}} z_j^{nad} \forall i \in \{1, \dots, m\}$
 - 7: **else**
 - 8: **for all** $i \in \{1, \dots, m\}$ **do**
 - 9: **if** $|z_i^{max} - z_i^{min}| < \epsilon$ **then**
 - 10: $z_i^{max} \leftarrow \max_{j \in \{1, \dots, m\}} z_j^{max}$
 - 11: Marcar z_i^{max}
 - 12: **else if** $z_i^{nad} > z_i^{max}$ **then**
 - 13: $z_i^{max} \leftarrow 2z_i^{nad} - z_i^{max}$
 - 14: Marcar z_i^{max}
 - 15: **else if** $v_i = 0 \wedge z_i^{max}$ no ha sido marcado recientemente **then**
 - 16: Obtener el máximo valor a para z_i^{nad} para RECORD
 - 17: $z_i^{max} \leftarrow (z_i^{max} + a)/2$
 - 18: Marcar z_i^{max}
 - 19: **return** \bar{z}^{min} y \bar{z}^{max}
-

4.5. Construcción de iMOACO_ℝ

En las secciones precedentes se describieron los mecanismos básicos que se usan en la construcción de la propuesta presentada en el algoritmo 4.4, llamado iMOACO_ℝ (*indicator-based Multi-Objective Ant Colony Optimizer for Continuous Search Spaces*). Esta propuesta es una extensión de la metaheurística ACO para la resolución de POMs en espacios continuos, por lo que emplea como motor de búsqueda a ACO_ℝ (ver sección 3.3.4 en la página 52).

Anteriormente, se había mencionado que el algoritmo clasificación-R2 requiere una estructura especial para cada solución. En el caso de iMOACO_ℝ, cada hormiga $a \in \mathcal{A}$ y cada feromona $p \in \mathcal{T}$, donde \mathcal{A} denota el conjunto de hormigas y \mathcal{T} el archivo de feromonas, cuentan con la siguiente estructura de campos:

- \vec{x} : vector de variables de decisión,
- \vec{F} : vector de valores objetivo,
- \vec{F}_{norm} : vector de valores objetivo normalizados (este vector es empleado en el algoritmo clasificación-R2),
- α : valor actual de utilidad (empleado en clasificación-R2),
- u^* : mejor valor de utilidad (utilizado en clasificación-R2),
- $rank$: rango otorgado por clasificación-R2,
- $kernelIndex$: índice de la función gaussiana a emplear en el proceso de muestreo.

Este último campo es únicamente empleado por las hormigas; los restantes son comunes para hormigas y feromonas.

El algoritmo requiere seis parámetros para su funcionamiento: G_{max} , q , ξ , α , ϵ y h . G_{max} representa el número máximo de iteraciones que iMOACO_ℝ buscará la solución a un POM. Los parámetros q y ξ son empleados por el motor de búsqueda basado en ACO_ℝ. El algoritmo de actualización de puntos de referencia hace uso de los parámetros α y ϵ como umbral de varianzas y umbral de tolerancia, respectivamente. Finalmente, h es el parámetro de proporción empleado para calcular $N = C_{m-1}^{h+m-1}$, donde N también es usado como el número total de hormigas y el número de feromonas a guardar en \mathcal{T} .

En la línea 1 se calculan los N vectores de pesos convexos mediante el método SLD implementado con el algoritmo 4.2. En las líneas 2 a 4, se inicializa el archivo de feromonas \mathcal{T} generando soluciones a partir de una distribución uniforme y con base en ellas se inicializan los vectores \vec{z}^* y \vec{z}^{nad} para posteriormente normalizar los vectores de valores objetivo de las feromonas. En las líneas 5 y 6 se crea una instancia R de la estructura de datos RECORD y enseguida se introduce en ella el vector de nadir calculado a partir del conjunto de feromonas iniciales. En la línea 7 se establecen los rangos para las feromonas en \mathcal{T} usando el algoritmo clasificación-R2. Entre las líneas

Algoritmo 4.4 Estructura principal de iMOACO_ℝ

Entrada: G_{max} , q , ξ , α , ϵ , h **Salida:** Aproximación al frente óptimo de Pareto

- 1: Generar un conjunto de vectores de pesos Λ usando el algoritmo 4.2
 - 2: Inicializar aleatoriamente las feromonas en \mathcal{T}
 - 3: Inicializar \bar{z}^* y \bar{z}^{nad} usando las soluciones en \mathcal{T}
 - 4: Normalizar(\mathcal{T} , \bar{z}^* , \bar{z}^{nad})
 - 5: Crear instancia R de RECORD
 - 6: Insertar \bar{z}^{nad} en R
 - 7: $\mathcal{T} \leftarrow$ clasificación-R2(\mathcal{T})
 - 8: $t \leftarrow 0$
 - 9: **while** $t < G_{max}$ **do**
 - 10: **for all** $a \in \mathcal{A}$ **do**
 - 11: Generar solución para a empleando las feromonas en \mathcal{T}
 - 12: $\{\bar{z}^*, \bar{z}^{nad}\} \leftarrow$ ActualizarPuntosReferencia(\mathcal{A} , m , R) {Algoritmo 4.3}
 - 13: $\Psi \leftarrow \mathcal{A} \cup \mathcal{T}$
 - 14: Normalizar(Ψ , \bar{z}^* , \bar{z}^{nad})
 - 15: $\Psi \leftarrow$ clasificación-R2(Ψ)
 - 16: Ordenar Ψ , en forma creciente, con respecto a los criterios: (1) $rank$, (2) u^* , y (3) norma L_2
 - 17: $\mathcal{T} \leftarrow \emptyset$
 - 18: Copiar en \mathcal{T} los primeros N elementos de Ψ
 - 19: $\mathcal{T} \leftarrow$ clasificación-R2(\mathcal{T})
 - 20: $t \leftarrow t + 1$
 - 21: **return** \mathcal{T}
-

9 y 20 se ejecuta el ciclo principal de búsqueda de iMOACO_ℝ siempre y cuando no se exceda el límite máximo de generaciones dado por G_{max} . La generación de nuevas soluciones por parte de las hormigas, muestreando las FDPs multimodales gaussianas generadas con base en las feromonas de \mathcal{T} , es llevada a cabo en las líneas 10 y 11. Cabe mencionar que para la construcción de soluciones se emplean los mismos mecanismos que en ACO_ℝ, salvo que en este caso la calidad de las soluciones es establecida por el *rank* calculado por el algoritmo clasificación-R2. La nueva población de soluciones es usada para actualizar los vectores de referencia en la línea 12. En las líneas 13 a 15 se unen en un solo conjunto Ψ las nuevas soluciones en \mathcal{A} y las feromonas de \mathcal{T} para posteriormente normalizar sus vectores objetivo y así poder ser procesadas por el algoritmo clasificación-R2. El proceso de actualización de feromonas está comprendido en las líneas 16 a 19. En primer lugar, el conjunto Ψ es ordenado en forma creciente con base en tres criterios de ordenamiento: (1) el rango dado por clasificación-R2, (2) en caso de empate en el valor del rango, el criterio de desempate es el mejor valor de utilidad u^* , (3) si aún permanece el empate después de aplicar el segundo criterio, se toma en cuenta el valor de la norma L_2 . A continuación, todas las soluciones en \mathcal{T} son sustituidas tomando las primeras N soluciones en Ψ . El nuevo contenido del archivo de feromonas es sujeto al algoritmo clasificación-R2 para así ser usado en la siguiente generación. iMOACO_ℝ entrega la aproximación al frente óptimo de Pareto guardada en el archivo de feromonas.

La complejidad del ciclo principal de iMOACO_ℝ se analiza a continuación. La generación de nuevas soluciones toma $\mathcal{O}(N^2n)$. La actualización de los puntos de referencia requiere $\mathcal{O}(Nm)$. Tanto la unión de \mathcal{A} y \mathcal{T} como la normalización de sus vectores objetivo son ejecutadas en $\mathcal{O}(Nm)$. El algoritmo clasificación-R2 necesita $\mathcal{O}(N^2(\log N + m))$ para procesar Ψ , mientras que su ordenamiento requiere $\mathcal{O}(N \log N)$. Vaciar el archivo de feromonas toma $\mathcal{O}(1)$. El traslado de las N mejores soluciones en Ψ hacia \mathcal{T} toma $\mathcal{O}(N(n + m))$. Finalmente, vuelve a ser necesario $\mathcal{O}(N^2(\log N + m))$ para asignar rangos a \mathcal{T} . Con base en esto, la complejidad computacional para iMOACO_ℝ es $\mathcal{O}(N^2(\log N + m + n))$. El costo en memoria de este algoritmo es $\mathcal{O}(N(m + n))$.

4.6. Resumen

En este capítulo, se ha introducido un nuevo algoritmo de optimización multiobjetivo basado en la metaheurística ACO para resolver POMs en espacios de búsqueda continuos, llamado iMOACO_ℝ. La propuesta emplea un algoritmo de asignación de rangos basado en el indicador de calidad R2 con el fin de discriminar cuáles soluciones guardar en el archivo de feromonas. Se debe hacer notar que iMOACO_ℝ no usa en ninguno de sus mecanismos la dominancia de Pareto. En consecuencia, se espera que tenga un buen desempeño en la solución de POMs con muchas funciones objetivo ($m > 3$).

La complejidad del tiempo de ejecución para iMOACO_ℝ es $\mathcal{O}(N^2(\log N + m + n))$. El costo de almacenamiento es $\mathcal{O}(N(n + m))$, donde N es el tamaño de población y

también la cardinalidad del archivo de feromonas, n es la dimensión del espacio de decisión, y m es la dimensión del espacio objetivo.

Capítulo 5

Estudio experimental

Este capítulo está orientado a mostrar el desempeño de la propuesta $iMOACO_{\mathbb{R}}$ con respecto a tres AEMOs y un AOMCH para dominios continuos. $iMOACO_{\mathbb{R}}$ es analizado empleando POMs desde dos hasta diez dimensiones en el espacio de las funciones objetivo. Además, se examina la sensibilidad de $iMOCO_{\mathbb{R}}$ al variar dos parámetros que afectan el motor de búsqueda subyacente, es decir, $ACO_{\mathbb{R}}$.

El desempeño de los optimizadores se analiza empleando una metodología basada en estadística no paramétrica. Primeramente, para cada POM en particular, todos los optimizadores ejecutan 30 corridas independientes. Los frentes de Pareto generados son evaluados empleando indicadores de desempeño del estado del arte [32], para así contar con un valor escalar que denote su calidad. Posteriormente, para realizar las comparaciones de desempeño entre los optimizadores, se utiliza una de las pruebas no paramétricas de una cola más populares en el área de optimización multi-objetivo: la prueba de la suma de rango de Wilcoxon [185]. Para realizar esta prueba se ha empleado el paquete R-project para calcular estos valores. Todos los experimentos han sido llevados a cabo en una PC que cuenta con un procesador Intel(R) Xeon(R) con una frecuencia de reloj de 3.0 GHz y memoria RAM de 64 Gb, empleado el sistema operativo Fedora 15.

En la sección 5.1, se describen los 21 problemas adoptados para los estudios comparativos, mencionando sus características y parámetros empleados. En la sección 5.2, se especifican los indicadores de evaluación de desempeño que son utilizados para determinar la calidad de los optimizadores. En la sección 5.3, se presenta el experimento principal de este trabajo, es decir, la comparación de desempeño de $iMOACO_{\mathbb{R}}$ con respecto a otras metaheurísticas. En la sección 5.4, se explica y analiza el segundo experimento que está relacionado con la sensibilidad de la propuesta a dos parámetros que controlan su capacidad de búsqueda. Finalmente, en la sección 5.6 se resume el comportamiento observado de $iMOACO_{\mathbb{R}}$ a través de los experimentos realizados.

Tabla 5.1: Características de los conjuntos de problemas de prueba ZDT, DTLZ y WFG.

Problema	Separabilidad	Frontalidad	Geometría	Sesgo
ZDT1	separable	unifrontal	convexo	no
ZDT2	separable	unifrontal	cóncavo	no
ZDT3	separable	unifrontal	desconectado	no
ZDT4	separable	multifrontal	convexo	no
ZDT6	separable	multifrontal	cóncavo	sí
DTLZ1	separable	multifrontal	lineal	no
DTLZ2	separable	unifrontal	cóncavo	no
DTLZ3	separable	multifrontal	cóncavo	no
DTLZ4	separable	unifrontal	cóncavo	polinomial
DTLZ5	desconocido	unifrontal	degenerado	dependiente de los parámetros
DTLZ6	desconocido	unifrontal	degenerado	dependiente de los parámetros
DTLZ7	$f_{1:m-1}$ no aplica, f_m separable	$f_{1:m-1}$ unimodal, f_m multimodal	desconectado, mixto	no
WFG1	separable	unifrontal	$f_{1:m-1}$ convexo, f_m mixto	polinomial, plano
WFG2	no separable	$f_{1:m-1}$ unimodal, f_m multimodal	convexo, desconectado	no
WFG3	no separable	unifrontal	lineal, degenerado	no
WFG4	separable	multifrontal	cóncavo	no
WFG5	separable	deceptivo	cóncavo	no
WFG6	no separable	unifrontal	cóncavo	no
WFG7	separable	unifrontal	cóncavo	dependiente de los parámetros
WFG8	no separable	unifrontal	cóncavo	dependiente de los parámetros
WFG9	no separable	multifrontal, deceptivo	cóncavo	dependiente de los parámetros

Tabla 5.2: Configuración adoptada para el conjunto de problemas de prueba WFG.

Parámetros	Dimensión del espacio objetivo							
	3D	4D	5D	6D	7D	8D	9D	10D
Relativo a la posición	4	6	8	10	12	14	16	18
Variables de decisión	24	36	47	59	70	82	93	105

5.1. Problemas de prueba

Con la finalidad de comparar las metaheurísticas, se han elegido tres conjuntos de problemas de prueba. El primero es el conjunto Zitzler-Deb-Thiele (ZDT) [194], el cual ha sido ampliamente usado en el área de la optimización evolutiva multi-objetivo. En segunda instancia, se usa el banco de problemas de prueba Deb-Thiele-Laumanns-Zitzler (DTLZ) [48]. Por último, se adopta el conjunto de problemas Walking-Fish-Group (WFG) [104]. A diferencia del conjunto ZDT, los conjuntos DTLZ y WFG permiten escalar sus problemas en el número de objetivos. Todos los problemas de prueba adoptados son usados para minimización y cuentan con una gran variedad de geometrías para el frente óptimo de Pareto, tales como linealidad, convexidad, concavidad, degeneración, frentes mixtos y desconectados. Asimismo, presentan características deseables tales como separabilidad y multifrontalidad, las cuales hacen a los problemas más difíciles de resolver. En la tabla 5.1 se resumen las características principales de estos problemas y en el apéndice A se definen formalmente todos ellos.

En el conjunto de problemas DTLZ, el número total de variables de decisión está dado por $n = m + k - 1$, donde m es el número de objetivos, k ha sido puesto en 5 para DTLZ1, 10 para DTLZ2-6 y 20 para DTLZ7. En el caso de WFG, el número de variables de decisión y de parámetros relativos a la posición son mostrados en la tabla 5.2.

5.2. Evaluación de desempeño

Con el fin de evaluar y comparar los resultados generados por las metaheurísticas, se han seleccionado tres indicadores de desempeño del estado del arte. El primero de ellos es el hipervolumen (o métrica \mathcal{S}) [192, 27] que mide el grado de convergencia que ostenta una aproximación a \mathcal{PF}^* . Este es un indicador de desempeño unario y ha adquirido gran popularidad en optimización evolutiva multi-objetivo por ser el único indicador unario que es Pareto compatible. Se define como la suma de todos los hipervolumenes comprendidos entre el frente de Pareto y un punto de referencia. Puesto que el punto de referencia es sumamente importante, en la tabla 5.3 se muestran los valores adoptados para cada problema de prueba. Matemáticamente, el hipervolumen (HV) se describe mediante la ecuación (5.1). Cabe destacar que cuanto más grande su valor, se indica un mejor resultado.

$$HV(A : \bar{z}^{ref}) := \left\{ \bigcup \text{volumen}(v : \bar{z}^{ref}) \mid v \in A \right\} \quad (5.1)$$

Tabla 5.3: Puntos de referencia empleados en el cálculo del hipervolumen.

Problema de prueba	Punto de referencia
ZDT1-6	(11, 11)
DTLZ1	(1, 1, 1, ...)
DTLZ2, DTLZ4	(2, 2, 2, ...)
DTLZ3	(7, 7, 7, ...)
DTLZ5	(4, 4, 4, ...)
DTLZ6	(11, 11, 11, ...)
DTLZ7	(1, 1, 1, ..., 21)
WFG1-9	(3, 5, 7, ..., 2m + 1)

Se ha empleado el algoritmo propuesto por While et al. [186] para el cálculo exacto del hipervolumen desde 2D a 10D¹. La principal ventaja de esta implementación es su gran rapidez en comparación con otros algoritmos como el de Fonseca et al. [72].

El segundo indicador de calidad empleado es denominado *Spacing* (S) [32, 166]. Este indicador describe numéricamente la propagación de los vectores que constituyen la aproximación al frente de Pareto (PF_{known}). En otras palabras, este indicador mide la varianza de la distancia entre vectores vecinos en PF_{known} . Las ecuaciones (5.2) y (5.3) definen este indicador de desempeño.

$$S := \sqrt{\frac{1}{|PF_{known}| - 1} \sum_{i=1}^{|PF_{known}|} (\bar{d} - d_i)^2} \quad (5.2)$$

y

$$d_i = \min_j \left(\sum_{k=1}^m |f_k^i(\vec{x}) - f_k^j(\vec{x})| \right) \quad (5.3)$$

donde $i, j = 1, \dots, N$, \bar{d} es el promedio de todos los d_i , N es el número de vectores en PF_{known} y m es la dimensión del espacio objetivo. Cuando $S = 0$, todos los puntos están espaciados uniformemente. Este indicador de desempeño no requiere que se conozca el frente verdadero de Pareto, aunque normalmente se presupone que el algoritmo ha convergido antes de aplicarlo.

Finalmente, el tercer indicador seleccionado es el denominado *modified Inverted Generational Distance* (IGD+), propuesto por Ishibuchi et al. [108, 110]. Este indicador, recientemente propuesto, tiene por objeto tratar de resolver el problema con los conjuntos de referencia que presenta el indicador IGD [32, 33], particularmente en POMs en alta dimensionalidad [109]. IGD+ surge gracias a una pequeña modificación de IGD al tomar en cuenta la relación de dominancia de Pareto entre una solución y un punto de referencia cuando su distancia es calculada. Debido a la consideración de

¹En este trabajo se ha hecho uso de la versión 1.11 del algoritmo, disponible en <http://www.wfg.csse.uwa.edu.au/hypervolume>.

la dominancia de Pareto, IGD+ es débilmente Pareto compatible. La definición formal de IGD+, considerando un problema de minimización, se muestra en las ecuaciones (5.4) y (5.5).

$$\text{IGD}+(A) := \frac{1}{|Z|} \sum_{j=1}^{|Z|} \min_{a_i \in A} d_{\text{IGD}+}(a_i, z_j) \quad (5.4)$$

y

$$d_{\text{IGD}+} = \sqrt{\sum_{k=1}^m (\max\{a_k - z_k, 0\})^2} \quad (5.5)$$

donde A es la aproximación al frente de Pareto y Z es el conjunto de referencia. Cuando $\text{IGD}+ \rightarrow 0$, entonces A está muy cercana a Z . En consecuencia, IGD+ evalúa simultáneamente convergencia y distribución de las soluciones en un conjunto solución. Para un POM en específico, se calcula Z juntando todas las aproximaciones a los frentes de Pareto generadas por los optimizadores (filtrando las soluciones no dominadas) para después aplicar el algoritmo de clusterización k -medias [113] con el objeto de reducir el número de soluciones.

5.3. Experimento 1: Comparación de desempeño

En este experimento se compara el desempeño de iMOACO_R con respecto a tres AEMOs y un AOMCH para espacios continuos del estado del arte. A continuación se presenta una breve reseña de cada algoritmo utilizado, haciendo énfasis en la configuración de parámetros establecida en la experimentación:

1. **MOEA/D**: el *Multi-Objective Evolutionary Algorithm based on Decomposition* [191] tiene como idea fundamental la transformación de un POM en N subproblemas de optimización escalar. Estos subproblemas son optimizados de forma simultánea usando información de los T subproblemas más cercanos. Para la generación de los subproblemas, se crean N vectores de pesos convexos, de tal forma que haya una relación uno-a-uno entre vectores de pesos y subproblemas. MOEA/D emplea el método SLD [164, 167] para la generación de los vectores de pesos. En este experimento, MOEA/D es usado junto con la función de utilidad PBI [143]. Un parámetro crítico para el funcionamiento de este algoritmo es el tamaño de vecindario T el cual es establecido en 20. El código fuente de MOEA/D está disponible en <http://dces.essex.ac.uk/staff/zhang/webofmoead.htm>.
2. **NSGA-III**: el *Nondominating Sorting Genetic Algorithm III* [45] es una variante del NSGA-II [44] para problemas con muchas funciones objetivo. Los principales cambios aportados por NSGA-III yacen en el operador de selección. Además, para el mantenimiento de la diversidad entre los miembros de

la población, se hace uso de un conjunto adaptativo de puntos de referencia bien distribuidos en el espacio objetivo. Debido a que la implementación original del NSGA-III no está disponible al público, existen diferentes versiones disponibles, siendo la de Tsung-Che Chiang una de las más empleadas. En este trabajo se ha utilizado la versión 1.2 de Chiang disponible en <http://web.ntnu.edu.tw/~tcchiang/publications/nsga3cpp/nsga3cpp.htm>.

3. **SMS-EMOA (HypE)**: el *S Metric Selection-Evolutionary Multi-objective Optimization Algorithm* [18] es un AEMO muy popular basado en el hipervolumen. Su principal objetivo de diseño es proporcionar el máximo valor de hipervolumen con un número finito de puntos. Para lograr lo anterior se basa en dos pilares fundamentales: (1) se usa el ordenamiento no dominado como criterio de clasificación, y (2) el hipervolumen es aplicado como segundo criterio de selección para descartar el individuo que contribuya con el menor valor de hipervolumen en el peor nivel de la clasificación hecha previamente. Uno de los inconvenientes más determinantes de SMS-EMOA es su alto costo computacional cuando se calcula el valor exacto del hipervolumen en problemas de alta dimensionalidad. Para atacar esta desventaja, Bader y Zitzler propusieron HypE [10] que calcula una aproximación al hipervolumen usando un cierto número de muestras. En este experimento se usaron 10,000 muestras.
4. **MOACO_ℝ**: el *Multi-Objective Ant Colony Optimizer for Continuous Spaces* [77] (descrito en la sección 3.4.4 en la página 59) es un AOMCH para espacios de búsqueda continuos basado en ACO_ℝ [171]. En cada iteración, las nuevas soluciones generadas por las hormigas son ordenadas de acuerdo con el esquema de ordenamiento no dominado del NSGA-II. Posteriormente, sólo aquellas soluciones en el primer frente no dominado (el más cercano a \mathcal{PF}^*) son guardadas en la matriz de feromonas. En caso de que estas soluciones sobrepasen la capacidad de la matriz, se emplea la distancia de agrupamiento para descartar soluciones hasta quedar dentro del límite de la matriz. Tal como iMOACO_ℝ, MOACO_ℝ requiere los parámetros ξ , q y el tamaño de la matriz de feromonas. El código fuente de este algoritmo fue proporcionado por su autor el Dr. Abel García Nájera.

MOEA/D, NSGA-III y SMS-EMOA son algoritmos evolutivos multi-objetivo representativos del estado del arte en el área. Los operadores de variación adoptados por estos tres algoritmos son: cruce simulada binaria (SBX, por sus siglas en inglés) y mutación polinomial [43]. En todos los casos, la tasa de cruce (P_c) fue establecida en 1.0 y a la tasa de mutación (P_m) se le asignó el valor $1/n$ (donde n representa el número de variables de decisión). En problemas de baja dimensionalidad, el índice de distribución de cruce (N_c) fue establecido en 20, mientras que en problemas de alta dimensionalidad se usó $N_c = 30$, como es sugerido por Deb y Jain [45]. En todos los problemas, el índice de distribución de mutación (N_m) tuvo un valor de 20.

Al estar basados en ACO_ℝ, iMOACO_ℝ y MOACO_ℝ hacen uso de los parámetros q y ξ que controlan la diversificación de la búsqueda y la velocidad de convergencia,

Tabla 5.4: Parámetros comunes entre las metaheurísticas.

m	h	Tamaño de población	Generaciones	Número de evaluaciones
2	119	120	416	49920
3	14			
4	7			
5	5	126	396	49896
6	4			
7	3	84	595	49980
8		120	416	49920
9		165	303	49995
10		220	227	49940

respectivamente. Para todos los problemas, ya sea en baja o alta dimensionalidad, los valores asignados a los parámetros son: $q = 0.1$ y $\xi = 0.5$. Esto se decidió fundamentado en los resultados de $ACO_{\mathbb{R}}$ y en un estudio empírico de desempeño.

El tamaño de población y máximo número de generaciones adoptados en el experimento se muestran en la tabla 5.4. Los valores varían de acuerdo con m (el número de objetivos). Puesto que $iMOACO_{\mathbb{R}}$, MOEA/D y NSGA-III usan un conjunto de vectores de pesos construidos mediante el método SLD, en la tabla se muestra el valor del parámetro h que controla el número de vectores de pesos generados. El número total de evaluaciones de la función objetivo es determinado de tal forma que no exceda 50,000. Para el caso de $iMOACO_{\mathbb{R}}$ y $MOACO_{\mathbb{R}}$, el tamaño de población mostrado en la tabla corresponde con el tamaño de la matriz de feromonas y el número de hormigas usadas.

Se realizaron 30 ejecuciones independientes por cada uno de los cinco optimizadores comparados, en cada uno de los problemas de prueba adoptados. Para asegurar la superioridad de desempeño de $iMOACO_{\mathbb{R}}$ en los experimentos, se hace uso de la prueba de suma de rango Wilcoxon de una cola, empleando un nivel significativo del 5%. Todos los algoritmos fueron implementados usando codificación real en C/C++. El equipo utilizado en el proceso de ejecución de los algoritmos cuenta con un procesador Intel(R) Xeon(R) de cuatro núcleos con frecuencia de reloj de 3.00 GHz y memoria RAM de 64 Gb, bajo Linux en su distribución Fedora 15.

Discusión de resultados

En el apéndice B se muestra, empleando tablas, el resumen de los resultados numéricos obtenidos de la experimentación con las metaheurísticas en los conjuntos ZDT, DTLZ y WFG. Con base en estas tablas, las figuras 5.1, 5.2 y 5.3 muestran la clasificación de desempeño de los algoritmos por cada indicador de calidad en las diferentes dimensiones en el espacio de las funciones objetivo. Los histogramas muestran el número de ocasiones en que un algoritmo se clasificó en primero, segundo, tercero, cuarto o quinto lugar de desempeño, según sea el caso. Si no se muestra información, esto implica que el algoritmo no obtuvo resultados para clasificarse en

Clasificación de metaheurísticas para el conjunto ZDT

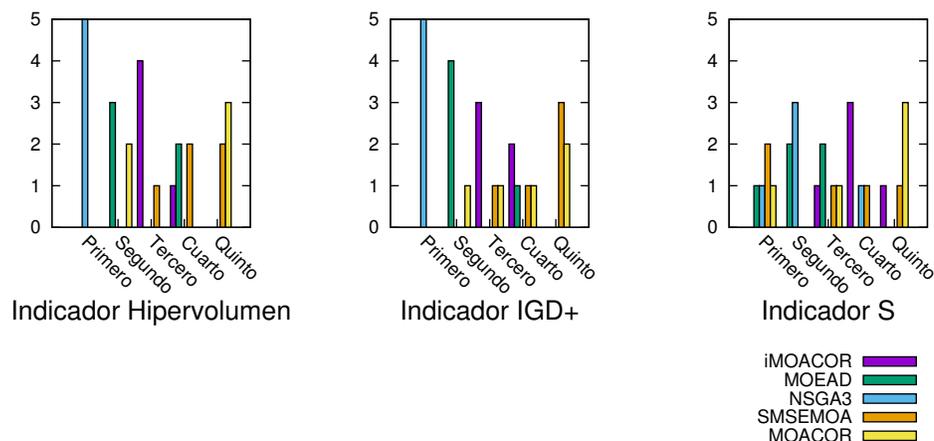


Figura 5.1: Clasificación de desempeño de las metaheurísticas probadas en el conjunto ZDT con base en los indicadores H, IGD+ y S. Por cada algoritmo, se muestra el número de POMs en los que obtiene una clasificación dada.

tal escaño.

En el conjunto de problemas ZDT, MOEA/D sólo pudo obtener el segundo lugar en el 60 % de los problemas (ZDT2, ZDT4 y ZDT6), en cuanto al hipervolumen se refiere. En el restante 40 % se posiciona en cuarto lugar. Los resultados en IGD+ son coherentes con los de hipervolumen, ya que en el 80 % de los problemas de prueba obtiene el segundo lugar, siendo únicamente en ZDT3 donde se obtiene el cuarto lugar. Debido a que MOEA/D usa los vectores de pesos para controlar la diversidad de las soluciones, en el indicador S se obtienen buenos resultados. En cuanto a los problemas de prueba DTLZ, MOEA/D se posiciona como el segundo mejor algoritmo, en términos de convergencia, en baja dimensionalidad. De hecho, MOEA/D presenta en general los mejores resultados en alta dimensionalidad, salvo para siete y ocho dimensiones donde NSGA-III se posiciona en primer lugar. Gracias a sus buenas características de distribución de soluciones y convergencia, en IGD+ ostenta los mejores resultados desde tres hasta diez objetivos. Sin embargo, el rendimiento en convergencia mostrado en los problemas WFG no es tan bueno, siendo superado por NSGA-III e iMOACO_R tanto en alta como en baja dimensionalidad en los indicadores HV e IGD+. No obstante, sigue presentando buena distribución en las soluciones como se aprecia en la figura 5.6.

NSGA-III obtiene los mejores resultados para los problemas ZDT. En el 100 % de los problemas, obtiene el primer lugar en HV e IGD+ siendo muy significativa la diferencia en el valor de la media con respecto al segundo lugar en ambos indicadores. Para el indicador S, sólo en 20 % de los problemas obtiene el primer lugar, pero toma el segundo puesto en 60 % de los problemas y el restante 20 % obtiene el tercer lugar. Para el conjunto DTLZ con tres objetivos, el desempeño en convergencia cae

Clasificación de metaheurísticas para el conjunto DTLZ

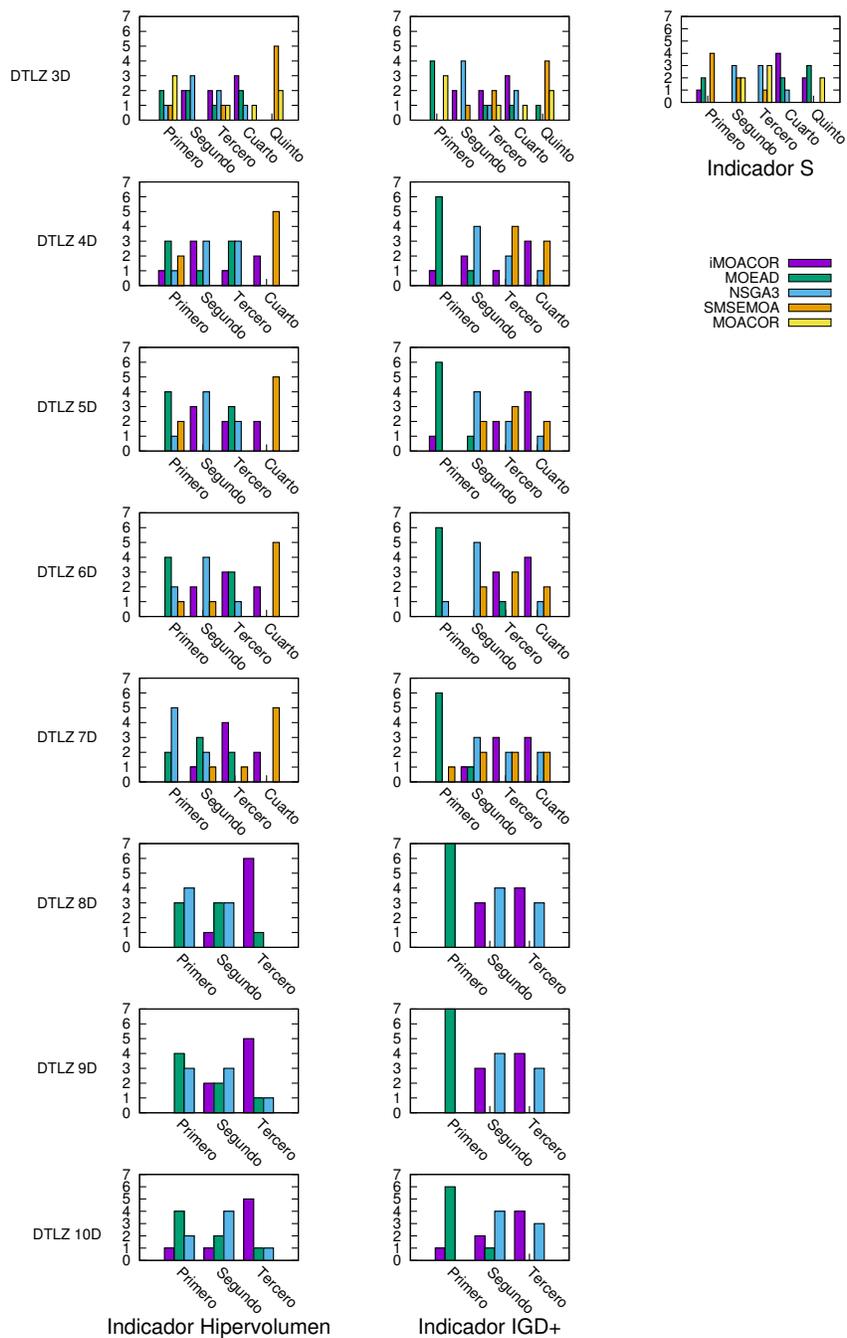


Figura 5.2: Clasificación de desempeño de las metaheurísticas, por cada dimensión del espacio objetivo, probadas en el conjunto DTLZ con base en los indicadores H, IGD+ y S. Por cada algoritmo, se muestra el número de POMs en los que obtiene una clasificación dada.

Clasificación de metaheurísticas para el conjunto WFG

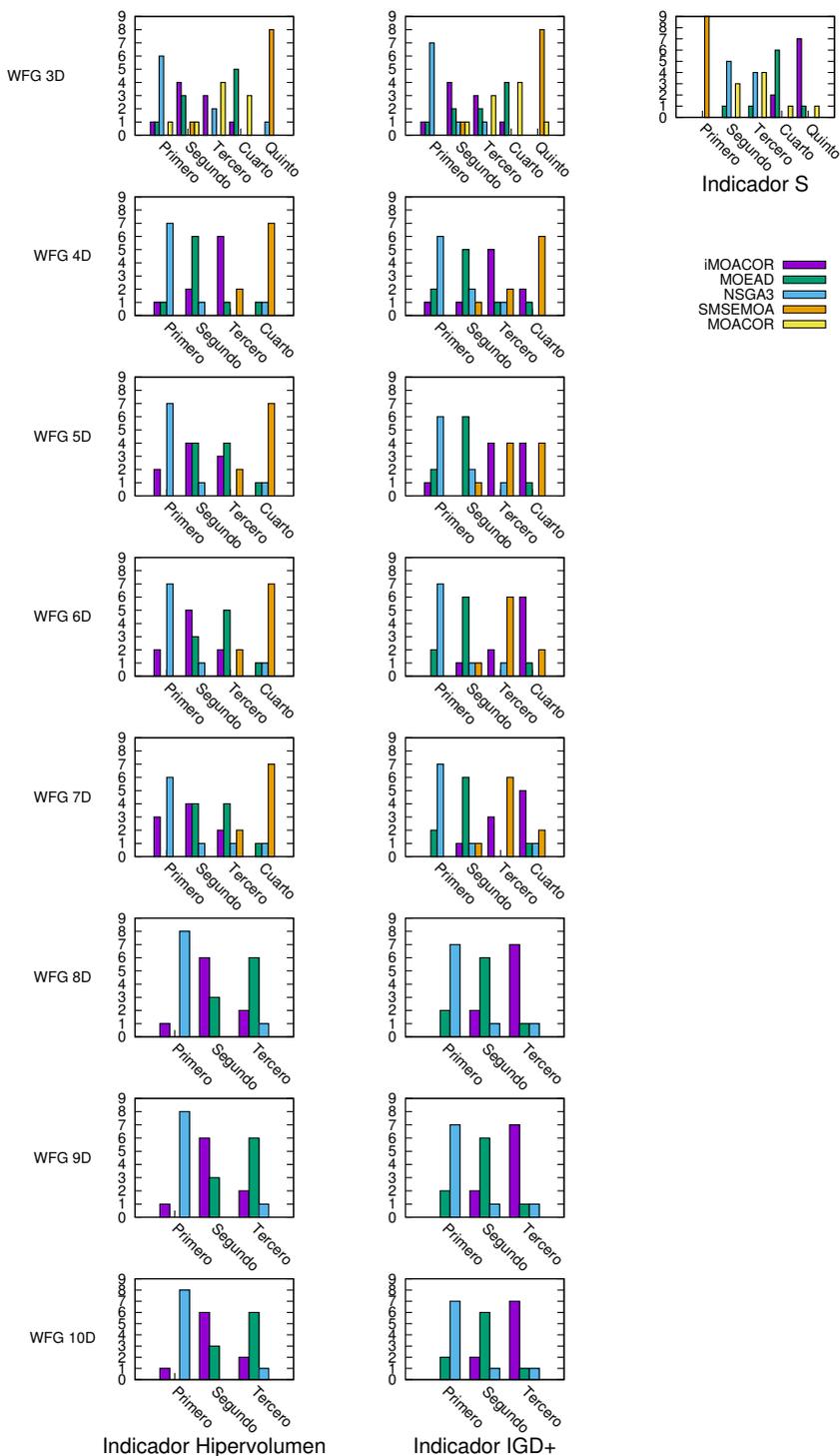


Figura 5.3: Clasificación de desempeño de las metaheurísticas, por cada dimensión del espacio objetivo, probadas en el conjunto WFG con base en los indicadores H, IGD+ y S. Por cada algoritmo, se muestra el número de POMs en los que obtiene una clasificación dada.

hasta el tercer sitio, detrás de MOACO_R y MOEA/D en vista de que únicamente en DTLZ1 se obtiene el primer lugar. En alta dimensionalidad, la convergencia en este conjunto de problemas de prueba mejoró al grado de obtener el segundo lugar general, sólo detrás de MOEA/D. Cabe destacar que esta versión del NSGA-III presenta muy mala convergencia para DTLZ4 en todas las dimensiones consideradas. Los resultados en IGD+ son igualmente coherentes con lo obtenido en hipervolumen. Para los problemas WFG, tanto en baja como en alta dimensionalidad del espacio objetivo, NSGA-III presenta los mejores resultados con respecto al hipervolumen. Para tres dimensiones, obtiene el primer puesto en el 66.66 % de los problemas de prueba (WFG3, WFG4, WFG5, WFG7, WFG8 y WFG9), el tercer puesto en WFG2 y WFG7 y el peor lugar de convergencia en WFG1. En alta dimensionalidad, desde cuatro hasta seis dimensiones obtiene el primer lugar en el 77.77 % de los problemas de prueba, mientras que en el rango de ocho a diez dimensiones tiene la mejor convergencia en el 88.88 % de los problemas. En el 77.77 % de los POMs en tres dimensiones tuvo el primer lugar en IGD+, como era de esperarse, debido a su alto grado de convergencia y distribución (ver figura 5.6). De igual manera que con el hipervolumen, en alta dimensionalidad, se conserva el dominio en relación con el indicador IGD+. Un aspecto a destacar es que presenta problemas de convergencia y distribución para WFG1, WFG2 y WFG3 en todas las dimensiones adoptadas, sin embargo, en WFG1 es más notorio este inconveniente, teniendo en cuenta que nunca puede superar los resultados de iMOACO_R. Finalmente, para el indicador S se posiciona en la cuarta posición general.

El desempeño en convergencia de SMS-EMOA (empleando la aproximación al hipervolumen con HypE) en todos los conjuntos de prueba se ve afectado gravemente por concentrar soluciones en regiones muy pequeñas de los frentes de Pareto. En baja dimensionalidad, en los conjuntos ZDT, DTLZ y WFG, concentra las soluciones únicamente en la rodilla del frente. Un ejemplo de esto se muestra en las figuras 5.4, 5.5 y 5.6, respectivamente. Como consecuencia, en todos los casos figura en la última posición de convergencia. Otro indicador afectado por el problema de convergencia es IGD+, donde SMS-EMOA siempre se posiciona en el último sitio de rendimiento. En contraste, para el indicador S se consiguen muy buenos resultados dado que para los tres conjuntos de problemas de prueba obtiene el primer lugar en la clasificación.

Con respecto a los resultados generados por SMS-EMOA en el indicador S, existe un argumento que puede refutar su buen desempeño. Este indicador representa la desviación estándar de la propagación de los vectores solución en el espacio objetivo, por lo que si su valor es pequeño, mejor será el resultado. Dado que las soluciones de SMS-EMOA se concentran en una región muy reducida del espacio, la variabilidad de la propagación será muy pequeña. Como resultado, se obtienen muy buenos resultados en el indicador S, aunque realmente no estén bien distribuidos en el espacio objetivo. Con base en lo anterior, se puede considerar que SMS-EMOA realmente obtiene los peores resultados en el indicador S debido a su falta de uniformidad y distribución de soluciones. Las figuras 5.4, 5.5 y 5.6 nuevamente sirven de apoyo para observar que los otros algoritmos presentan mejores características de dispersión (y en algunos

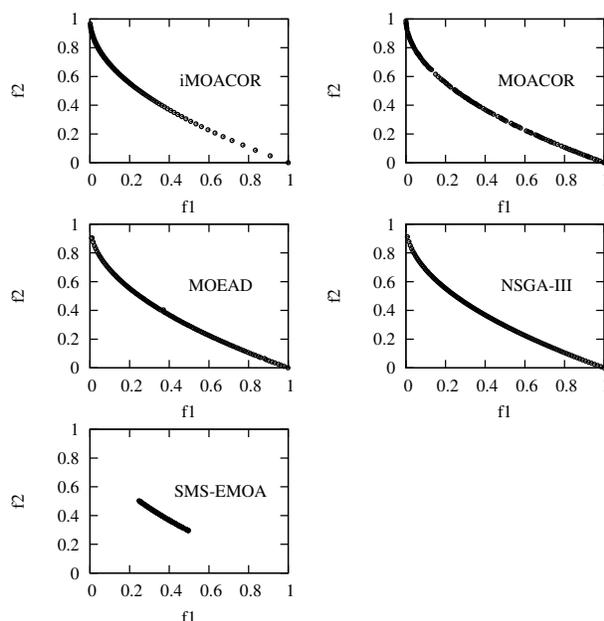


Figura 5.4: Comparación de la cobertura y distribución de soluciones para el problema ZDT1 obtenida por $iMOACO_{\mathbb{R}}$ con respecto a $MOACO_{\mathbb{R}}$, MOEA/D, NSGA-III y SMS-EMOA. La aproximación corresponde con la ejecución en la mediana del valor de hipervolumen.

casos de uniformidad) de las soluciones.

En el 60 % de los problemas ZDT, es decir, en ZDT2, ZDT4 y ZDT6, $MOACO_{\mathbb{R}}$ no puede converger. En ZDT2 se presenta pérdida de diversidad en vista de que se obtienen valores de cero en los objetivos. El motor de búsqueda $ACO_{\mathbb{R}}$ ha demostrado tener serios inconvenientes en problemas multimodales, de donde se infiere que $MOACO_{\mathbb{R}}$ presente problemas con la multifrontalidad y no pueda converger en ZDT4 y ZDT6. En ZDT1 y ZDT3, $MOACO_{\mathbb{R}}$ obtiene en ambos el segundo lugar en el valor de hipervolumen. Tanto en el indicador IGD+ como en S, se obtiene el segundo lugar en ZDT3 y el tercero para ZDT1. En relación con el valor de hipervolumen en el conjunto de problemas DTLZ, $MOACO_{\mathbb{R}}$ obtiene en DTLZ5, DTLZ6 y DTLZ7 el primer lugar, en DTLZ4 el tercer lugar y en DTLZ2 el cuarto puesto, lo cual lo hace el de mejor rendimiento entre todas las metaheurísticas. Sin embargo, no presenta convergencia en DTLZ1 y DTLZ3 al ser POMs multifrontales. Para el indicador IGD+, se consiguen los mismos resultados que en el hipervolumen, pero con ello sólo se clasifica como el segundo mejor algoritmo. En cuanto al indicador S, $MOACO_{\mathbb{R}}$ tiene el quinto lugar general de la clasificación. Este hecho puede ser explicado por la falta de uniformidad en las soluciones como se muestra en la figura 5.5. El rendimiento de esta metaheurística en el conjunto WFG baja considerablemente. En relación con el hipervolumen, sólo en WFG2 se obtiene el primer puesto, en WFG6 el segundo sitio, en cuatro problemas el tercer lugar y en los tres restantes el cuarto puesto. Como se aprecia en la figura 5.6, no hay uniformidad en la distribución de soluciones. En vista

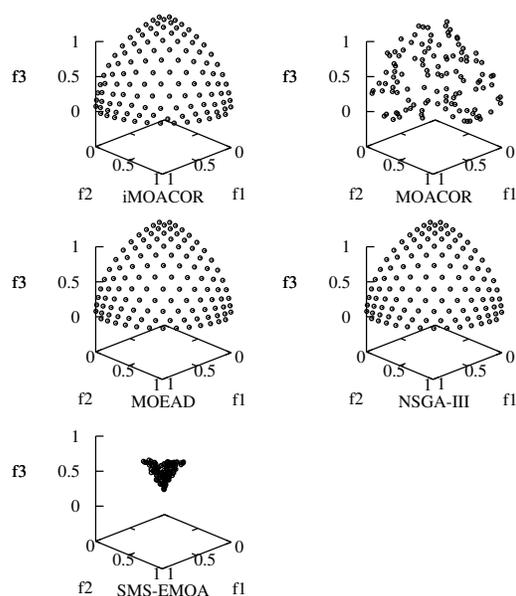


Figura 5.5: Comparación de la cobertura y distribución de soluciones para el problema DTLZ2 en tres dimensiones obtenidas por $iMOACO_{\mathbb{R}}$ con respecto a $MOACO_{\mathbb{R}}$, MOEA/D, NSGA-III y SMS-EMOA. La aproximación corresponde con la ejecución en la mediana del valor de hipervolumen.

de lo anterior, el algoritmo se clasifica en el cuarto lugar general en los indicadores IGD+ y S. En conclusión, en los problemas WFG sólo supera a SMS-EMOA y la diferencia con respecto a las otras metaheurísticas es sumamente significativa.

El desempeño de $iMOACO_{\mathbb{R}}$ en los problemas ZDT es regular. Con respecto a los problemas ZDT1, ZDT2, ZDT3 y ZDT6, en términos de hipervolumen, se obtiene el tercer puesto en cada uno. Al igual que en $MOACO_{\mathbb{R}}$, no se puede converger en ZDT4. En cuando a ZDT1 y ZDT3, la diferencia con respecto al segundo lugar ($MOACO_{\mathbb{R}}$) es muy pequeña (en promedio es del orden de 0.01). En ZDT2 se presenta una mayor variabilidad que se refleja en el valor de la desviación estándar. En la figura B.1 se observa que en ZDT6 hay mayor dificultad para converger totalmente al frente, aunque se ha probado que con un mayor número de evaluaciones de la función objetivo este problema se resuelve. Un aspecto a destacar es que $iMOACO_{\mathbb{R}}$ tiende a concentrar soluciones en mayor medida en la región donde las soluciones tienen valores pequeños en f_1 y altos en f_2 . Este fenómeno se aprecia en la figura B.1 para todos los problemas ZDT. En el indicador IGD+, se mantiene el tercer lugar de rendimiento entre los algoritmos comparados. Como resultado de la distribución de soluciones característica de $iMOACO_{\mathbb{R}}$, se consigue el quinto lugar general para el indicador S.

En los problemas de prueba DTLZ en baja dimensionalidad, los resultados de $iMOACO_{\mathbb{R}}$ son muy pobres. En convergencia logra el cuarto lugar general sólo su-

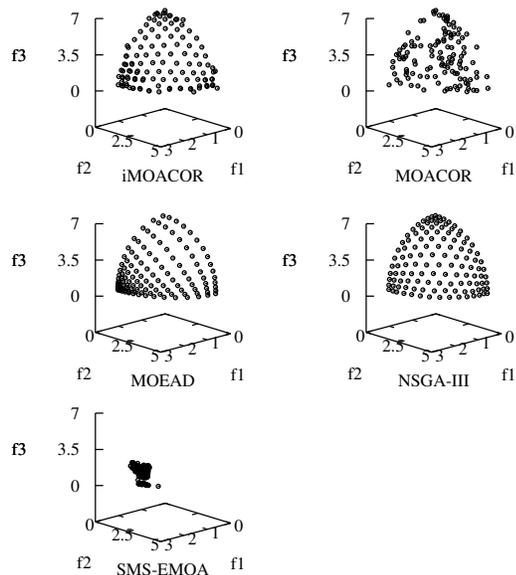


Figura 5.6: Comparación de la cobertura y distribución de soluciones para el problema WFG6 en tres dimensiones obtenidas por $iMOACO_{\mathbb{R}}$ con respecto a $MOACO_{\mathbb{R}}$, MOEA/D, NSGA-III y SMS-EMOA. La aproximación corresponde con la ejecución en la mediana del valor de hipervolumen.

perando a SMS-EMOA. Únicamente, se logran dos segundos lugares para DTLZ4 y DTLZ5 (superando de forma estadísticamente significativa a $MOACO_{\mathbb{R}}$, NSGA-III y SMS-EMOA), dos terceros puestos en DTLZ7 y DTLZ2 y un cuarto lugar para DTLZ6. Con respecto a DTLZ1 y DTLZ3 no se logra converger debido a la multifrontalidad. DTLZ6 presenta la mayor dificultad para $iMOACO_{\mathbb{R}}$ pues hay demasiada variabilidad en los resultados del hipervolumen (ver tabla B.2) y problemas en general de convergencia y distribución como se observa en la figura B.9. A pesar de mostrar un desempeño regular en DTLZ7, como se aprecia en la figura B.11, se vuelve a tener una alta concentración de soluciones para valores altos de f_3 tal como sucede en los problemas ZDT. Sin embargo, en problemas como DTLZ2 se nota una muy buena uniformidad en las soluciones como se muestra en la figura 5.5. Con base en los resultados de convergencia y distribución mostrados, para el indicador IGD+ se logra también el cuarto lugar general, mientras que para el indicador S se obtiene el quinto lugar.

En relación con el conjunto de problemas DTLZ en alta dimensionalidad, $iMOACO_{\mathbb{R}}$ se posiciona, tanto en hipervolumen como en IGD+, en tercer lugar de la clasificación. En convergencia, $iMOACO_{\mathbb{R}}$ se comporta de mejor a peor proporción en los problemas siguientes: DTLZ7, DTLZ6, DTLZ5, DTLZ4 y DTLZ2. Se debe hacer énfasis en que DTLZ7 es sin duda el POM donde hay un mejor comportamiento en todas las dimensiones adoptadas. Desde tres hasta nueve dimensiones, el valor de hipervo-

lumen se mantiene sólo por debajo de aquel de NSGA-III. Por el contrario, en diez dimensiones se logra obtener el primer lugar de desempeño presentando una diferencia considerable con respecto a NSGA-III y, además, los resultados son estadísticamente significativos. Como se aprecia de la figura 5.2, para el hipervolumen los mejores resultados se dan entre cuatro y siete dimensiones, mientras que para IGD+, a partir de ocho dimensiones se comienza a presentar un mejor comportamiento. Esto último puede deberse a una mejor distribución de soluciones en tales dimensiones.

De acuerdo con los resultados de la tabla B.10, resumidos en la figura 5.3, iMOACO_R presenta su mejor desempeño en los problemas WFG. En hipervolumen, gana en WFG6 y obtiene el segundo puesto (con una diferencia muy pequeña con respecto al primer lugar) en WFG2, WFG3, WFG7 y WFG9; se ubica en tercer puesto para WFG4, WFG5 y WFG8, mientras que obtiene el cuarto lugar en WFG1. Con respecto al indicador IGD+, se obtienen resultados similares, salvo que en WFG2 cae al tercer sitio y WFG5 sube al segundo lugar. Todos los resultados muestran diferencias estadísticamente significativas. No obstante, un punto en contra es que suelen presentarse distribuciones no tan uniformes en las orillas de los frentes pero sí en la rodilla como se muestra en la figura 5.6. En consecuencia, el indicador S vuelve a sufrir pérdida de calidad.

Finalmente, es en los problemas WFG en alta dimensionalidad donde se nota un incremento sustancial del desempeño de iMOACO_R. En este caso, los resultados de la propuesta son competitivos con respecto a NSGA-III y superiores a los del MOEA/D. En cuanto a convergencia, los mejores resultados están en el rango de cinco a siete dimensiones donde se gana en el 33.33 % de los problemas, mientras que de cuatro a diez dimensiones se queda en segundo lugar en el 66.66 % de los problemas. iMOACO_R demuestra ser muy bueno para los problemas WFG1, WFG2 y WFG3. En particular, iMOACO_R ostenta un dominio absoluto en WFG1 desde cuatro a diez dimensiones al obtener siempre el primer lugar de desempeño con resultados estadísticamente significativos. Para WFG2 se domina el primer lugar de convergencia desde cuatro hasta siete dimensiones. En lo que respecta a WFG3 siempre se mantiene en segundo lugar salvo en siete dimensiones donde obtiene el primer puesto. El indicador IGD+ se ve disminuido por la inconstante uniformidad de las soluciones. Entre cuatro y siete dimensiones, mantiene en promedio el cuarto lugar de rendimiento; y entre ocho y diez dimensiones, el tercer sitio. A pesar del dominio en WFG1, no se puede expresar esto a través de IGD+. En conclusión, iMOACO_R demuestra su potencial en los problemas WFG tanto en baja como en alta dimensionalidad, pero con mayor fuerza para alta cardinalidad del espacio objetivo.

5.4. Experimento 2: Análisis de varianza

El análisis de varianza de un solo factor [185, 141] (ANOVA², por sus siglas en inglés) ha sido llevado a cabo para determinar qué tan sensible resulta iMOACO_R

²La prueba ANOVA fue realizada mediante el paquete de computación estadística R.

Tabla 5.5: Combinaciones resultantes de los niveles establecidos para los parámetros q y ξ .

Nombre	q	ξ
C1	0.0001	0.5
C2	0.0001	0.85
C3	0.1	0.5
C4	0.1	0.85
C5	0.25	0.5
C6	0.25	0.85
C7	1.0	0.5
C8	1.0	0.85

con respecto a los parámetros q y ξ , y cómo esto afecta en el valor del hipervolumen. El parámetro q controla la diversificación de la búsqueda, mientras que ξ regula la velocidad de convergencia del algoritmo (ver sección 4.5 en la página 69). En el estudio se empleó un subconjunto de los POMs (en baja dimensionalidad) utilizados en el experimento anterior: ZDT2, ZDT6, DTLZ2, DTLZ5, DTLZ7 y WFG1-WFG9. La elección para los conjuntos ZDT y DTLZ se basó en tomar los problemas donde iMOACO_ℝ tuvo un desempeño bueno, regular y malo de acuerdo con el experimento de la sección anterior; por lo que respecta al conjunto WFG, se decidió usar todo el conjunto en vista de que la propuesta demostró tener un buen rendimiento en estos problemas.

Puesto que se analiza la sensibilidad de iMOACO_ℝ con respecto a q y ξ , el ANOVA requiere el establecimiento de diferentes niveles para cada uno de los parámetros. Estos niveles han sido establecidos basados en información de ACO_ℝ, así como también mediante un proceso previo de experimentación. Con base en esto, el diseño del ANOVA es el siguiente:

- Los niveles de cada parámetro son:
 - $q = \{0.0001, 0.1, 0.25, 1.0\}$
 - $\xi = \{0.5, 0.85\}$
- Con base en los niveles, se han formado 8 combinaciones de parámetros como se muestra en la tabla 5.5.
- Se realizaron 30 ejecuciones independientes por cada combinación de parámetros en cada problema de prueba.
- Para las pruebas estadísticas se usó un nivel de significancia $\alpha = 0.05$, es decir, un nivel de confianza del 95%.
- Las hipótesis empleadas en ANOVA se exponen a continuación.

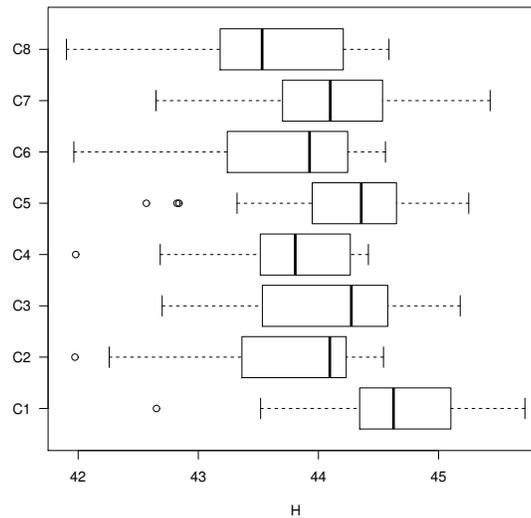


Figura 5.7: Datos de estadística descriptiva para el problema WFG1 empleando iMOACO_R con las diferentes combinaciones.

- Hipótesis nula (H_0): no hay diferencia significativa entre los promedios del hipervolumen de los resultados obtenidos y si hubiera diferencias, son producto de efectos aleatorios.
- Hipótesis alternativa (H_1): hay una o varias combinaciones de los parámetros q y ξ para los cuales los promedios del hipervolumen son diferentes y no son producto de efectos aleatorios.

La prueba ANOVA tiene únicamente la capacidad de indicar cuándo se tiene una diferencia estadísticamente significativa entre las poblaciones adoptadas, pero no proporciona información específica acerca de cuáles de ellas difirieron. Para poder saber con exactitud qué combinaciones de parámetros difieren de forma estadísticamente significativa, se ha optado por emplear la prueba Tukey³ [185] para efectuar comparaciones pareadas y así conocer tal información. Los resultados de la prueba Tukey pueden ser mostrados mediante un diagrama de intervalos de confianza, donde cada intervalo de confianza corresponde con un emparejamiento. En este diagrama, aquellos intervalos que contienen el valor cero indican que no existe una diferencia estadísticamente significativa entre las poblaciones comparadas, mientras que aquellas que no los contienen sí presentan una diferencia estadísticamente significativa.

Los resultados obtenidos rechazan H_0 en todos los casos, salvo para el problema DTLZ5. A pesar de que esto indique en primera instancia que hay una gran sensibilidad a los parámetros, se han encontrado patrones que permiten establecer un

³La prueba Tukey permite la formación de intervalos de confianza del $100(1-\alpha)\%$ simultáneos para todas las comparaciones pareadas. El método se basa en la distribución del rango «estudentizado».

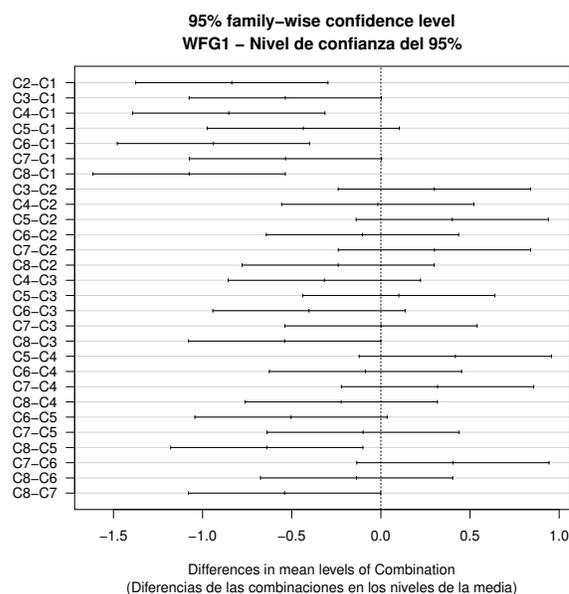


Figura 5.8: Intervalos de confianza generados por la prueba Tukey en el problema WFG1. Aquellos intervalos que intersectan el valor cero, indican que no hay una diferencia estadísticamente significativa entre las combinaciones pareadas.

comportamiento regular entre los conjuntos de problemas de prueba. Con la finalidad de analizar los casos donde H_1 ha sido aceptada, se hace uso de los diagramas de cajas (generados previamente a la aplicación del ANOVA) y de los diagramas de intervalos de confianza producidos por la prueba Tukey. Un ejemplo de estos tipos de diagramas para el problema WFG1 se muestran en las figuras 5.7 y 5.8. El resto de las figuras se muestran en el apéndice B.

Para los casos donde se prueba la hipótesis nula, se puede concluir lo siguiente:

- **Conjunto ZDT.** Los problemas seleccionados de este conjunto no presentan alta sensibilidad a los parámetros. Sin embargo, hay una tendencia a optar por valores de q que promuevan un muestreo más uniforme de las soluciones en la matriz de feromonas. El valor de ξ no interfiere en la calidad de las soluciones. Se puede notar que en combinaciones donde $q, \xi \rightarrow 0$ (como en C1) la calidad de los conjuntos solución se degrada considerablemente. Esto se debe a que cuando el valor de q está cercano a cero, se eligen con mucha mayor probabilidad soluciones de la parte alta de la clasificación de la matriz de feromonas; mientras que cuando $\xi \rightarrow 0$, la capacidad de exploración se reducen a una región muy cercana alrededor de la solución elegida. Evidentemente, esto degrada en términos generales la habilidad exploratoria del algoritmo.
- Para el problema ZDT2, los resultados entre la combinación C1 con respecto a las combinaciones C2 a C8 son estadísticamente significativas, dando

un mejor desempeño para estas últimas. En efecto, no hay una diferencia estadísticamente significativa entre todas las parejas formadas por las combinaciones C2 a C8.

- Para el problema ZDT6, se observa un desempeño mayor para las combinaciones C3 a C8. Entre estas combinaciones las diferencias no son estadísticamente significativas, pero sí lo son con respecto a las combinaciones C1 y C2.
- **Conjunto DTLZ.** De los problemas de prueba elegidos de este conjunto, no fue posible encontrar un patrón de comportamiento de los parámetros, aunque no se encuentra una injerencia específica de q . A continuación se detallan los resultados por cada problema.
- Para el problema DTLZ2, se encontró que aquellas combinaciones que tienen en común el valor $\xi = 0.5$ (C1, C3, C5 y C7) presentan un mejor valor de hipervolumen y, asimismo, este comportamiento es estadísticamente significativo con respecto a las combinaciones con valor $\xi = 0.85$ (C2, C4, C6 y C8). Con base en esto, este problema sólo es sensible al valor de ξ y no al parámetro q .
 - Para el problema DTLZ5, no hubo suficiente evidencia estadística para rechazar H_0 , por lo que se puede decir que este problema no es sensible a los parámetros.
 - Para el problema DTLZ7, las combinaciones donde $\xi = 0.85$ muestran un mejor desempeño y, además, los resultados son estadísticamente significativos con respecto a las combinaciones restantes. Tal como en DTLZ2, este problema es sensible al parámetro ξ pero no a q .
- **Conjunto WFG.** En estos problemas de prueba se encontró en mayor medida un comportamiento regular, y únicamente se detectó sensibilidad ante el valor del parámetro ξ , no importando el valor de q . En lo subsecuente se especifican los detalles para estos problemas de prueba.
- Para los problemas WFG1 a WFG8, se muestra claramente que las combinaciones C1, C3, C5 y C7 ostentan un mejor desempeño. Estas combinaciones tienen en común el valor $\xi = 0.5$. Las diferencias entre ellas no son estadísticamente significativas pero sí lo son con respecto a las combinaciones C2, C4, C6 y C8. Esto indica que para estos problemas no hay diferencia en tener un muestreo elitista o más equitativo. Lo que realmente afecta el desempeño es la amplitud de la búsqueda efectuada alrededor de la solución elegida.
 - Para el problema WFG9, se presenta una mayor sensibilidad a los parámetros. Se encontraron diferencias estadísticamente significativas sólo entre los emparejamientos C4-C2, C8-C2, C4-C3, C8-C3 y C7-C4 (ver figura

Tabla 5.6: Valor medio del hipervolumen para el problema DTLZ2 resuelto por iMOACO_R incrementando el número máximo de generaciones en múltiplos de 416. El mejor valor del hipervolumen para este problema corresponde al NSGA-III con 7.421715 (ver tabla B.2.)

G_{max}	Valor medio H
416	7.420231
832	7.420959
1248	7.421165
1664	7.421338
3328	7.421592
4992	7.421797

B.54). En consecuencia, no es claro determinar un patrón de comportamiento en vista de que las combinaciones C2, C3, C4, C7 y C8 no son del todo homogéneas. No obstante, se puede afirmar que se prefieren combinaciones con $q > 0.0001$ y $\xi = 0.5$ en mayor medida.

5.5. Análisis de debilidades

Para explicar la razón por la que iMOACO_R tiene un pobre desempeño en baja dimensionalidad, se analizará el problema DTLZ2 en tres dimensiones. En la figura 5.5, se puede notar que no hay diferencia visible entre los frentes generados por iMOACO_R, NSGA-III y MOEA/D para el problema seleccionado. Sin embargo, en la tabla B.2 se puede notar que iMOACO_R tiene el peor valor de hipervolumen de los tres. Con base en esta evidencia, la hipótesis formulada es la siguiente: el bajo rendimiento de iMOACO_R en baja dimensionalidad, y en general para todas las dimensiones, resulta de una habilidad muy débil de explotación de las buenas soluciones. Para comprobar esto, se ejecutó iMOACO_R sobre el problema DTLZ2 incrementando el número máximo de generaciones (lo que implícitamente incrementa el número máximo de evaluaciones de la función objetivo) hasta que el valor medio del hipervolumen de las 30 ejecuciones por caso sobrepasara el mejor valor de hipervolumen registrado en la tabla B.2, el cual corresponde al NSGA-III. En la tabla 5.6 se aprecia que sólo cuando $G_{max} = 4992$, el valor medio del hipervolumen es mayor que el del NSGA-III. Esto confirma la hipótesis establecida. Con 416 iteraciones, iMOACO_R genera un frente como el de la figura 5.5. Sin embargo, para poder alcanzar el mejor valor de hipervolumen, el algoritmo debe refinar o explotar las buenas soluciones actuales. Este proceso de explotación es demasiado lento en iMOACO_R.

Una posible razón de la disminuida habilidad de explotación se debe al mecanismo de generación de soluciones, tomado del ACO_R. La parte crítica del mecanismo es el cálculo de la desviación estándar (ver ecuación (3.13)) que caracteriza la distribución normal que es muestreada con el fin de generar una nueva solución. Esta desviación estándar representa el error promedio de todas las feromonas con respecto a la feromona elegida, para la variable de decisión en turno. Adicionalmente, este promedio

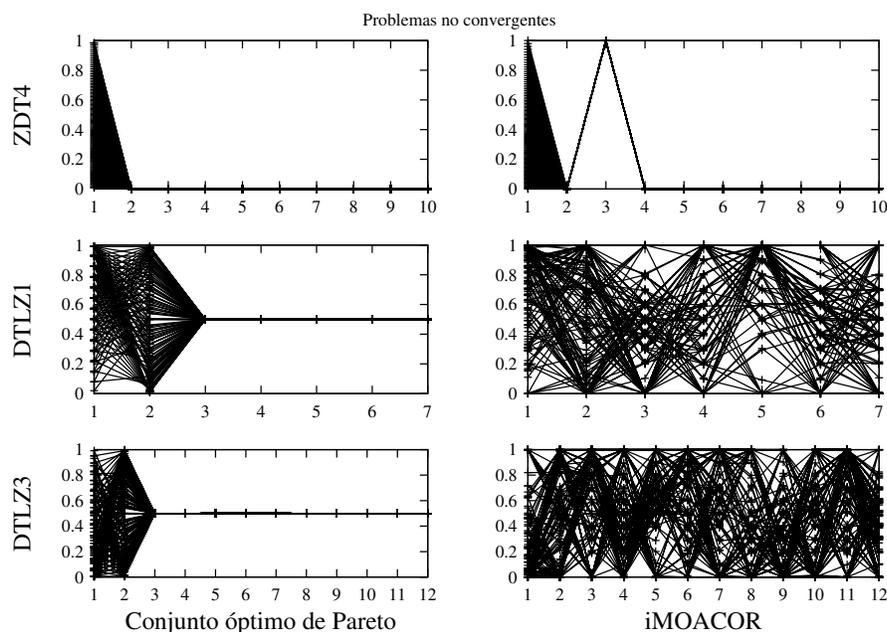


Figura 5.9: Para los problemas ZDT4, DTLZ1 y DTLZ3, se muestra el espacio de las variables de decisión correspondiente al conjunto óptimo de Pareto y para la aproximación al conjunto de Pareto generada por $iMOACO_{\mathbb{R}}$.

es ponderado por el parámetro ξ que controla la velocidad de convergencia o, en otras palabras, el rango de exploración. Con el experimento realizado con la función DTLZ2, se observó que el valor de ξ en la parte final del proceso de búsqueda es muy grande. En consecuencia, el algoritmo ofrece un gran poder exploratorio pero no puede explotar las buenas soluciones, haciendo que el refinamiento de las soluciones sea muy lento, tal como se demostró en el experimento.

Otro aspecto importante que se notó es que $iMOACO_{\mathbb{R}}$ no puede converger en los problemas ZDT4, DTLZ1 y DTLZ3. Estos problemas tienen en común una alta multifrontalidad. Esta deficiencia no ha sido esclarecida en su totalidad, pues en otros problemas con alto grado de multifrontalidad como WFG4, WFG5 y WFG9, la propuesta ha podido converger, aunque sus valores de hipervolumen no son los mejores entre todos los algoritmos. Sin embargo, se piensa nuevamente que la poca capacidad de explotación de $iMOACO_{\mathbb{R}}$ juega un papel importante en este problema. Esta inferencia surge de analizar el espacio de las variables de decisión para cada problema, como se muestra en la figura 5.9. De la figura es posible advertir que $iMOACO_{\mathbb{R}}$ no puede centrar la búsqueda alrededor de un cierto valor como pasa, por ejemplo, en las variables x_3 a x_7 en DTLZ1. Debido a la alta capacidad de exploración de $iMOACO_{\mathbb{R}}$, no se pueden explotar ciertas soluciones.

5.6. Resumen

En este capítulo se presentaron los estudios experimentales hechos con $iMOACO_{\mathbb{R}}$. Por una parte, se proporcionó un estudio comparativo de la propuesta con respecto a cuatro metaheurísticas del estado del arte empleando varios indicadores de desempeño. El otro experimento llevado a cabo consistió en determinar la sensibilidad de $iMOACO_{\mathbb{R}}$ con respecto a dos de sus parámetros.

Los resultados numéricos de la comparación de desempeño indican que $iMOACO_{\mathbb{R}}$ es un optimizador competitivo para la resolución de POMs. En la tabla 5.7, se muestra el lugar que ocupa de forma general la propuesta por cada conjunto de problemas de prueba de acuerdo con los indicadores de calidad considerados. Con base en esta tabla, se observa que $iMOACO_{\mathbb{R}}$ presenta un rendimiento competitivo en el conjunto WFG en todas las dimensiones adoptadas. Sin embargo, se aprecia un bajo desempeño en los problemas DTLZ en todas las dimensiones. En lo que respecta al conjunto ZDT, el comportamiento de la propuesta está por debajo de NSGA-III y MOEA/D. No obstante, un punto importante a mencionar es su falta de convergencia en los problemas ZDT4, DTLZ1 y DTLZ3.

NSGA-III fue, sin lugar a dudas, el mejor optimizador de todos. Su desempeño fue superior para los problemas ZDT y WFG. En los problemas DTLZ únicamente queda por debajo de MOEA/D, donde éste muestra el mejor desempeño de todos.

SMS-EMOA mostró un deterioro en la calidad de las soluciones en todos los conjuntos de prueba. Esto fue consecuencia de una alta concentración de soluciones en regiones muy pequeñas del espacio objetivo. Debido a esto, los valores de HV e IGD+ se vieron altamente afectados. SMS-EMOA presenta el mejor lugar de desempeño para el indicador S en todos los conjuntos de prueba. Sin embargo, esto se debe a la alta concentración de soluciones mencionada, por lo que realmente esto es consecuencia de un problema en el indicador.

El desempeño de $MOACO_{\mathbb{R}}$ en convergencia resultó muy bueno en el conjunto DTLZ en tres dimensiones donde se clasificó como el mejor, a pesar de no converger en DTLZ1 y DTLZ3. En cuanto al conjunto ZDT, sólo pudo resolver los problemas ZDT1 y ZDT3 ya que en ZDT2, ZDT4 y ZDT6 no logró converger. En los problemas WFG, sólo pudo superar a SMS-EMOA en el valor del hipervolumen. Debido a su mecanismo de diversidad basado en distancia de agrupamiento, no muestra una distribución uniforme de soluciones.

Otro aspecto estudiado en $iMOACO_{\mathbb{R}}$ es su sensibilidad ante los parámetros q y ξ que controlan la capacidad exploratoria del algoritmo. En vista de los resultados arrojados por la prueba ANOVA junto con la prueba Tukey, la calidad de las soluciones generadas por $iMOACO_{\mathbb{R}}$ presenta una mayor sensibilidad al parámetro ξ que a q . Sin embargo, para el conjunto WFG (donde $iMOACO_{\mathbb{R}}$ presentó un mejor desempeño en el experimento de la sección anterior en la página 77) se ha observado un patrón de comportamiento donde con $\xi = 0.5$ se aseguran resultados mejores con un nivel de confianza del 95 %, no importando el valor de q . Esto respalda la elección de parámetros en la comparación contra otras metaheurísticas. Para el conjunto ZDT, se pide

Tabla 5.7: Clasificación general de los optimizadores considerados en el experimento de comparación en cada conjunto de problemas de prueba por cada indicador de desempeño.

Conjunto	Indicador	1 ^{er} lugar	2 ^{do} lugar	3 ^{er} lugar	4 ^{to} lugar	5 ^{to} lugar
ZDT	HV	NSGA-III	MOEA/D	iMOACO _R	SMS-EMOA	MOACO _R
	IGD+	NSGA-III	MOEA/D	iMOACO _R	SMS-EMOA	MOACO _R
	S	SMS-EMOA	NSGA-III	MOEA/D	MOACO _R	iMOACO _R
DTLZ 3D	HV	MOACO _R	MOEA/D	NSGA-III	iMOACO _R	SMS-EMOA
	IGD+	MOEA/D	MOACO _R	NSGA-III	iMOACO _R	SMS-EMOA
	S	SMS-EMOA	MOEA/D	iMOACO _R	NSGA-III	MOACO _R
DTLZ 4D-10D	HV	MOEA/D	NSGA-III	iMOACO _R	SMS-EMOA	N/A
	IGD+	MOEA/D	NSGA-III	iMOACO _R	SMS-EMOA	N/A
WFG 3D	HV	NSGA-III	iMOACO _R	MOEA/D	MOACO _R	SMS-EMOA
	IGD+	NSGA-III	iMOACO _R	MOEA/D	MOACO _R	SMS-EMOA
	S	SMS-EMOA	NSGA-III	MOEA/D	MOACO _R	iMOACO _R
WFG 4D-10D	HV	NSGA-III	iMOACO _R	MOEA/D	SMS-EMOA	N/A
	IGD+	NSGA-III	MOEA/D	iMOACO _R	SMS-EMOA	N/A

un valor de q que promueva un muestreo equitativo y no hay evidencia estadística de que ξ afecte la calidad de las soluciones. Finalmente, los problemas DTLZ presentan una alta sensibilidad ante los parámetros del algoritmo propuesto, lo cual explica su mal comportamiento en ellos.

Capítulo 6

Conclusiones y trabajo futuro

En este trabajo se ha introducido un nuevo algoritmo de optimización multi-objetivo basado en el comportamiento colectivo de las hormigas y en el indicador R2, denominado $iMOACO_{\mathbb{R}}$. La propuesta usa el motor de búsqueda $ACO_{\mathbb{R}}$, el cual es un algoritmo de optimización mono-objetivo para espacios de búsqueda continuos cuya idea principal se centra en el comportamiento de las hormigas en su búsqueda de alimento. Gracias a que el mecanismo de selección empleado está completamente fundamentado en el indicador de desempeño R2 (y no utiliza en lo absoluto la dominancia de Pareto), $iMOACO_{\mathbb{R}}$ es capaz de escalar su funcionamiento en altas dimensiones en el espacio de las funciones objetivo. Esto último permite que $iMOACO_{\mathbb{R}}$ sea una alternativa viable en el área denominada *many-objective optimization*.

Los resultados experimentales muestran que $iMOACO_{\mathbb{R}}$ es capaz de resolver problemas complejos y mantener un comportamiento competitivo con respecto a las metaheurísticas consideradas. En primera instancia, NSGA-III presenta un comportamiento superior a $iMOACO_{\mathbb{R}}$ para los conjuntos de problemas ZDT y DTLZ (en todas las dimensiones adoptadas). Por su parte, MOEA/D sólo es superior a la propuesta para los problemas DTLZ y tiene un desempeño competitivo en los problemas ZDT. Sin embargo, de acuerdo con los resultados, $iMOACO_{\mathbb{R}}$ ostenta un rendimiento competitivo en el conjunto WFG, en todas las dimensiones adoptadas, con respecto al NSGA-III y superior al del MOEA/D. $iMOACO_{\mathbb{R}}$ supera ampliamente al SMS-EMOA en todos los conjuntos de problemas. Esto se debe a que SMS-EMOA utiliza el algoritmo HypE que realiza una aproximación al valor del hipervolumen, puesto que el cálculo exacto de éste es computacionalmente costoso. La utilización de la aproximación al hipervolumen produce un detrimento en el comportamiento de SMS-EMOA.

Por otra parte, $MOACO_{\mathbb{R}}$ tiene un desempeño pobre en los problemas ZDT, no convergiendo en tres de ellos (ZDT2, ZDT4 y ZDT6), mientras que $iMOACO_{\mathbb{R}}$ no converge únicamente en uno de ellos (ZDT4). En los problemas ZDT restantes, el comportamiento de ambos algoritmos es similar aunque $iMOACO_{\mathbb{R}}$ exhibe mejor uniformidad en las soluciones. Sin embargo, el comportamiento de $MOACO_{\mathbb{R}}$ es superior al de $iMOACO_{\mathbb{R}}$ en los problemas DTLZ en tres dimensiones. Se debe hacer énfasis en que ambos no convergen en DTLZ1 y DTLZ3. En cuanto a los problemas WFG en ba-

ja dimensionalidad, $iMOACO_{\mathbb{R}}$ es absolutamente superior a $MOACO_{\mathbb{R}}$. Con base en los resultados numéricos expuestos, $iMOACO_{\mathbb{R}}$ supera a $MOACO_{\mathbb{R}}$ en la mayoría de los problemas de prueba. Asimismo, $iMOACO_{\mathbb{R}}$ es, hasta el momento, el único ACO multi-objetivo para espacios de búsqueda continuos capaz de resolver problemas de alta dimensionalidad del espacio de las funciones objetivo.

Como conclusión final del desempeño de la propuesta se tiene lo siguiente. Se recomienda la utilización de $iMOACO_{\mathbb{R}}$ en problemas unifrontales y que presenten un frente de Pareto degenerado y/o desconectado, no importando si es lineal, convexo, cóncavo o mixto. Asimismo, el algoritmo no sufre afectación alguna en caso de haber sesgo en las variables. Esto se concluye basado en que $iMOACO_{\mathbb{R}}$ presenta sus mejores valores de hipervolumen, IGD+ y S en los problemas DTLZ7, DTLZ5, DTLZ6, WFG1, WFG2 y WFG3. También es necesario hacer énfasis en que $iMOACO_{\mathbb{R}}$ presentó un desempeño claramente superior a todos los algoritmos para WFG1 desde 4 hasta 10 dimensiones. WFG1 ha sido históricamente un POM sumamente difícil de resolver para los AEMOs del estado del arte. Entonces, un nicho importante para $iMOACO_{\mathbb{R}}$ yace en la solución de problemas similares a WFG1. Por otra parte, no se recomienda el uso de $iMOACO_{\mathbb{R}}$ en problemas con un alto grado de multifrontalidad.

Con respecto al análisis de sensibilidad a los parámetros se obtiene la siguiente conclusión. $iMOACO_{\mathbb{R}}$ es únicamente sensible al valor del parámetro ξ . En $ACO_{\mathbb{R}}$ se recomienda el uso de $\xi = 0.85$. Sin embargo, para el caso de $iMOACO_{\mathbb{R}}$, con este valor se obtienen valores más bajos de hipervolumen que con $\xi = 0.5$. Esto se debe a que un valor más alto de ξ favorece la exploración y hace más lenta la velocidad de convergencia en $iMOACO_{\mathbb{R}}$.

Evidentemente, falta mucho trabajo por hacer para mejorar el desempeño de $iMOACO_{\mathbb{R}}$. En primera instancia, se debe hacer una revisión profunda al motor de búsqueda $ACO_{\mathbb{R}}$ para resolver sus diferentes problemas, en especial en equilibrar la exploración y explotación del espacio de búsqueda. Con base en esto, se podría redefinir la adaptación del motor de búsqueda para la resolución de POMs. Otro punto importante a tratar es probar diferentes ASFs con el objeto de determinar su influencia en la diversidad y uniformidad de las soluciones. Finalmente, el algoritmo de selección denominado clasificación-R2, basado en el indicador R2, puede ser combinado con otro indicador de desempeño tal como IGD+ o Δ_p con el fin de compensar sus debilidades. Una línea posible de investigación es emplear un indicador para el proceso de exploración, mientras que el segundo indicador podría usarse para el proceso de explotación. Otra posibilidad es aplicar un indicador exclusivamente como criterio de selección y el restante emplearlo en un mecanismo de diversidad que ayude en problemas de alta dimensionalidad.

Apéndice A

Problemas de prueba

Con el fin de obtener un mejor entendimiento acerca del funcionamiento de los optimizadores, existe una gran cantidad de conjuntos de problemas de prueba artificiales propuestos para el área de optimización evolutiva multi-objetivo [32]. Estos problemas prueban las habilidades de los optimizadores para lidiar con diversas dificultades en los frentes óptimos de Pareto.

En este apéndice, se revisan tres bancos de problemas de prueba extensamente empleados en el área de optimización evolutiva multi-objetivo. Dos de estos conjuntos tienen la peculiaridad de poder ser empleados en optimización con muchas funciones objetivo. Finalmente, los problemas que se presentan en este apéndice están definidos para dominios de búsqueda continuos y no tienen restricciones.

A.1. Conjunto de problemas Zitzler-Deb-Thiele

Uno de los conjuntos de problemas de prueba más ampliamente usados en el área de optimización evolutiva multi-objetivo es sin duda el denominado Zitzler-Deb-Thiele¹ (ZDT) [194]. Los problemas de prueba ZDT comprenden seis POMs (ZDT1-ZDT6), sin embargo, ZDT5 usa codificación binaria por lo que es descartado de esta sección, mientras que los restantes son POMs para espacios de diseño continuos. En la tabla A.1 se presenta un resumen de las características de estos POMs.

Los problemas ZDT comparten muchas características con el kit de problemas de Deb [41], incluyendo aspectos tales como la multifrontalidad, que puede causar problemas con transformaciones muchos-a-uno (ZDT6), problemas desconectados (ZDT3), y problemas multifrontales (ZDT4). Es importante notar que todos los problemas ZDT sólo usan un parámetro de posición, es decir, una función objetivo (f_1) de una única variable de decisión, mientras que las variables restantes son denominadas parámetros de distancia.

De acuerdo con Huband et al. [104], estos problemas de prueba ofrecen dos ventajas principales: los frentes óptimos de Pareto están bien definidos y los resultados de una gran cantidad de investigaciones están comúnmente disponibles, lo que facilita

¹Se nombra de esta manera por las primeras letras de los apellidos de sus autores.

Tabla A.1: Resumen de características del conjunto de problemas de prueba ZDT.

Problema	Separabilidad	Frontalidad	Geometría
ZDT1	separable	unifrontal	convexo
ZDT2	separable	unifrontal	cóncavo
ZDT3	separable	unifrontal	desconectado
ZDT4	separable	multifrontal	convexo
ZDT6	separable	multifrontal	cóncavo, sesgado

la comparación de nuevos algoritmos. Sin embargo, a pesar de ser inmensamente populares, presentan las siguientes desventajas:

- No son escalables en el número de funciones objetivo; los problemas son únicamente bi-objetivo.
- Sólo son escalables en el número de parámetros de distancia.
- Ninguno de estos problemas tienen un paisaje de aptitud con regiones planas.
- Su único problema deceptivo tiene codificación binaria.
- Ninguno de los problemas es no separable.

A continuación se ofrece la definición formal de cada uno de los problemas ZDT considerados para este trabajo.

ZDT1

Este problema presenta un frente óptimo de Pareto convexo y se define de la siguiente forma:

$$\begin{aligned}
 \text{Dado} \quad \vec{x} &= (x_1, \dots, x_n)^T \\
 \text{Minimizar} \quad f_1(\vec{x}) &= x_1 \\
 f_2(\vec{x}) &= g(\vec{x}) \cdot (1 - \sqrt{f_1/g(\vec{x})}) \quad (\text{A.1}) \\
 \text{donde} \quad g(\vec{x}) &= 1 + \frac{9}{n-1} \cdot \sum_{i=2}^n x_i \\
 \text{sujeto a} \quad &\forall i \in \{1, \dots, n\} 0 \leq x_i \leq 1.
 \end{aligned}$$

donde $n = 30$. El frente óptimo de Pareto está formado con $g(\vec{x}) = 1$. En la figura [A.1](#) se muestra el frente correspondiente.

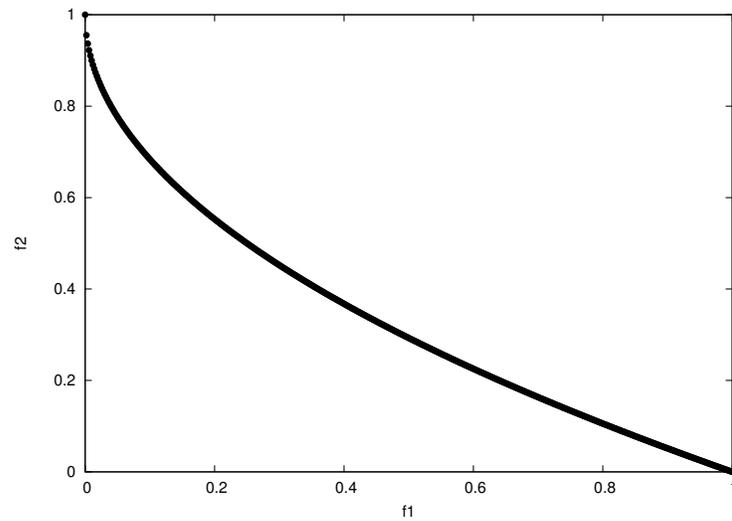


Figura A.1: Frente óptimo de Pareto para ZDT1.

ZDT2

El frente de Pareto de este problema (ver figura A.2) tiene geometría convexa. Se define como sigue:

$$\begin{aligned}
 \text{Dado} \quad \vec{x} &= (x_1, \dots, x_n)^T \\
 \text{Minimizar} \quad f_1(\vec{x}) &= x_1 \\
 f_2(\vec{x}) &= g(\vec{x}) \cdot (1 - (f_1/g(\vec{x}))^2) \quad (\text{A.2}) \\
 \text{donde} \quad g(\vec{x}) &= 1 + \frac{9}{n-1} \cdot \sum_{i=2}^n x_i \\
 \text{sujeto a} \quad &\forall i \in \{1, \dots, n\} 0 \leq x_i \leq 1.
 \end{aligned}$$

donde $n = 30$. El frente óptimo de Pareto está formado con $g(\vec{x}) = 1$.

ZDT3

El frente óptimo de Pareto de este problema consta de diversos fragmentos convexos no contiguos (frente desconectado). A continuación se establece su definición

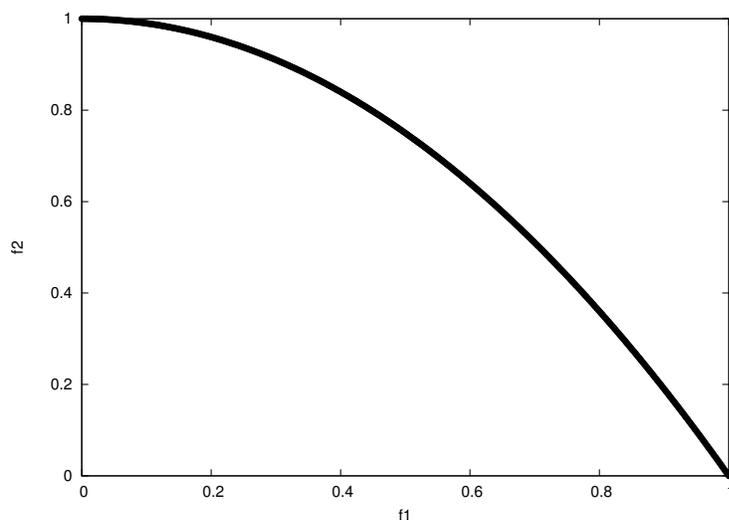


Figura A.2: Frente óptimo de Pareto para ZDT2.

formal:

$$\text{Dado} \quad \vec{x} = (x_1, \dots, x_n)^T$$

$$\text{Minimizar} \quad f_1(\vec{x}) = x_1$$

$$f_2(\vec{x}) = g(\vec{x}) \cdot \left(1 - \sqrt{\frac{f_1}{g(\vec{x})}} - \frac{f_1}{g(\vec{x})} \cdot \text{sen}(10\pi f_1) \right) \quad (\text{A.3})$$

$$\text{donde} \quad g(\vec{x}) = 1 + \frac{9}{n-1} \cdot \sum_{i=2}^n x_i$$

$$\text{sujeto a} \quad \forall i \in \{1, \dots, n\} 0 \leq x_i \leq 1.$$

donde $n = 30$. El frente óptimo de Pareto se forma cuando $g(\vec{x}) = 1$. La introducción de la función senoidal causa discontinuidad en el frente. Sin embargo, no hay discontinuidad en el espacio de las variables de decisión. En la figura A.3 se muestra el frente correspondiente.

ZDT4

Este problema se caracteriza por ser multifrontal (ver figura A.4). En efecto, contiene 21^9 frentes óptimos de Pareto locales y, por ende, prueba la habilidad de un optimizador para resolver problemas con esta propiedad. Su definición matemática es

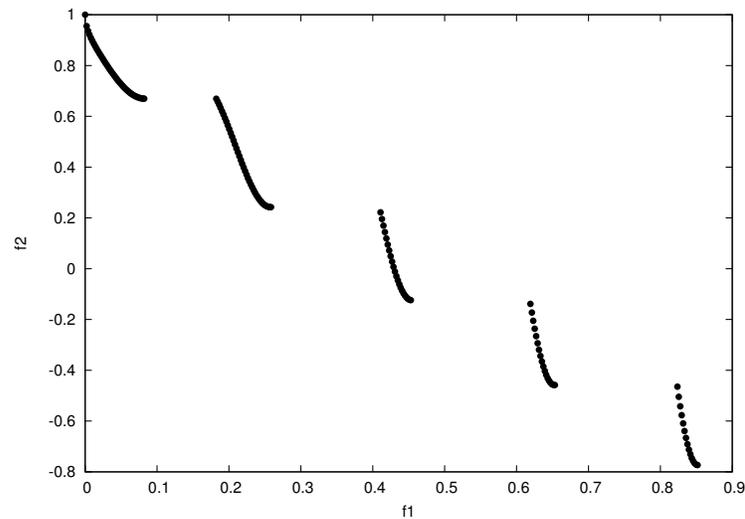


Figura A.3: Frente óptimo de Pareto para ZDT3.

la siguiente:

$$\begin{aligned}
 \text{Dado} \quad \vec{x} &= (x_1, \dots, x_n)^T \\
 \text{Minimizar} \quad f_1(\vec{x}) &= x_1 \\
 f_2(\vec{x}) &= g(\vec{x}) \cdot \left(1 - \sqrt{f_1/g(\vec{x})}\right) \tag{A.4} \\
 \text{donde} \quad g(\vec{x}) &= 1 + 10 \cdot (n - 1) + \sum_{i=2}^n (x_i^2 - 10 \cos(4\pi x_i)) \\
 \text{sujeto a} \quad &x_1 \in [0, 1] \wedge x_{i=2:n} \in [-5, 5].
 \end{aligned}$$

donde $n = 10$. El frente de Pareto se forma con $g(\vec{x}) = 1$ y el mejor frente local se obtiene cuando $g(\vec{x}) = 1.25$.

ZDT6

El último problema de este conjunto incluye dos dificultades causadas por la no uniformidad del espacio de búsqueda: (1) el conjunto óptimo de Pareto no está uniformemente distribuido a lo largo del frente óptimo de Pareto (el frente está sesgado para soluciones en las cuales $f_1(\vec{x})$ es cercano a uno), (2) la densidad de las soluciones es más baja cuanto más próximo esté el frente y más alta al estar lejos de éste. El

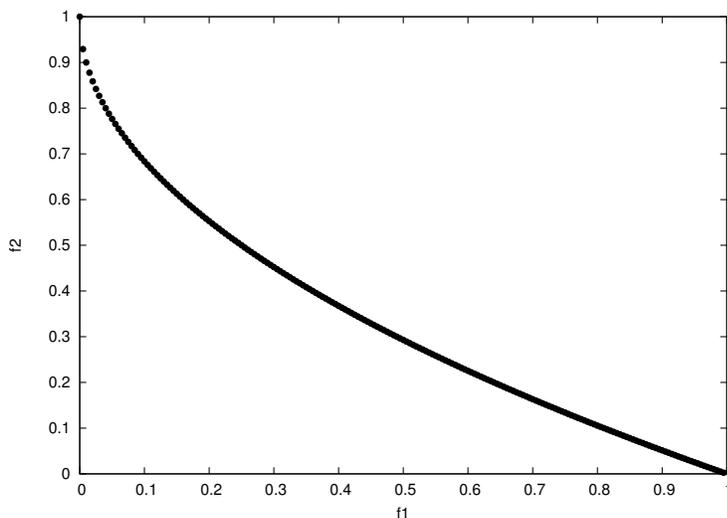


Figura A.4: Frente óptimo de Pareto para ZDT4.

problema se define de la siguiente manera:

$$\begin{aligned}
 \text{Dado} \quad \vec{x} &= (x_1, \dots, x_n)^T \\
 \text{Minimizar} \quad f_1(\vec{x}) &= 1 - e^{-4x_1} \cdot \text{sen}^6(6\pi x_1) \\
 f_2(\vec{x}) &= g(\vec{x}) \cdot (1 - (f_1/g(\vec{x}))^2) \\
 \text{donde} \quad g(\vec{x}) &= 1 + 9 \cdot \left[\frac{\sum_{i=2}^n x_i}{9} \right]^{0.25} \\
 \text{sujeto a} \quad &\forall i \in \{1, \dots, n\} 0 \leq x_i \leq 1.
 \end{aligned} \tag{A.5}$$

donde $n = 10$. El frente óptimo de Pareto se forma con $g(\vec{x}) = 1$ y es no convexo tal como se muestra en la figura A.5.

A.2. Conjunto de problemas Deb-Thiele-Laumanns-Zitzler

El banco de problemas de prueba Deb-Thiele-Laumanns-Zitzler (DTLZ) fue descrito originalmente de forma abreviada por Deb et al. [48] y, posteriormente, los mismos autores lo presentaron en forma completa en [49]. A diferencia de la mayoría de los problemas de prueba que existían previamente en optimización evolutiva multi-objetivo, los problemas DTLZ son genéricos y escalables en el número de objetivos (debido a esto son muy útiles en el área denominada *many-objective optimization*

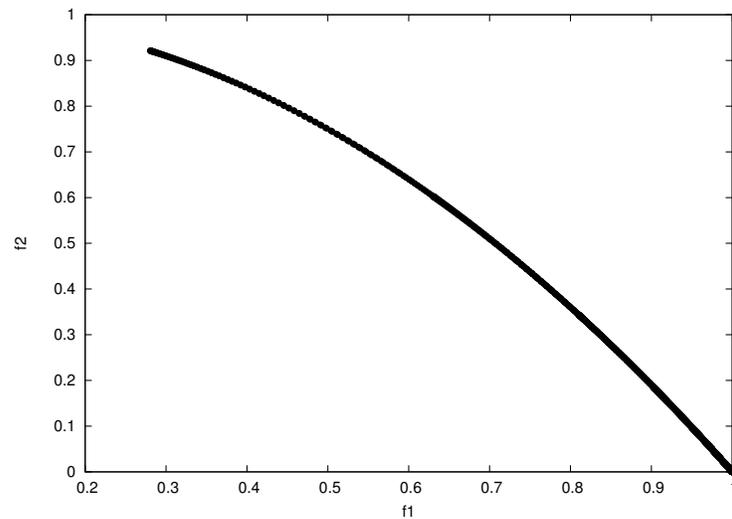


Figura A.5: Frente óptimo de Pareto para ZDT6.

[112]). El primer reporte técnico generado por Deb et al. [49] contenía nueve problemas de prueba (DTLZ1-DTLZ9) donde siete de ellos no tenían restricciones y dos sí, sin embargo, una versión más reciente [48] reduce la cantidad a sólo siete problemas, eliminando así los problemas originales DTLZ5 y DTLZ9. En esta sección se describen los POMs del artículo del conjunto DTLZ y en la tabla A.2 se resumen sus características principales.

Los problemas DTLZ plantean un gran avance dado que permiten investigar las propiedades de los optimizadores en problemas con muchas funciones objetivo en una manera controlada. Sin embargo, al igual que el kit de problemas de Deb y el conjunto ZDT, el banco de pruebas DTLZ tiene varias desventajas:

- Ninguno de sus problemas ostenta un paisaje de aptitud con regiones planas.
- Ningún problema es engañoso.
- Ningún problema es no separable.
- El número de parámetros de posición es siempre fijo y relativo al número de objetivos.
- Para espacios objetivo de alta dimensionalidad, los frentes óptimos de Pareto para DTLZ5 y DTLZ6 no son del todo claros [104].

En lo subsecuente, se presentan los siete problemas sin restricciones del banco de pruebas DTLZ. Se debe enfatizar que el número total de variables de decisión está dado por $n = m + k - 1$, donde m representa el número de objetivos y k es el número de parámetros de distancia.

Tabla A.2: Resumen de características del conjunto de problemas de prueba DTLZ. N/A significa «no aplica»

Problema	Separabilidad	Frontalidad	Geometría
DTLZ1	separable	multifrontal	lineal
DTLZ2	separable	unifrontal	cóncavo
DTLZ3	separable	multifrontal	cóncavo
DTLZ4	separable	unifrontal	cóncavo
DTLZ5	N/A	unifrontal	degenerado
DTLZ6	N/A	unifrontal	degenerado
DTLZ7	separable	unifrontal	desconectado

DTLZ1

Este problema es separable y multifrontal, teniendo un frente óptimo de Pareto lineal. Su descripción matemática es la siguiente:

$$\begin{aligned}
 \text{Dado} \quad \vec{x} &= (x_1, \dots, x_{m-1}, x_m, \dots, x_n)^T \\
 \text{Minimizar} \quad f_1(\vec{x}) &= 0.5(1 + g(\vec{y})) \prod_{i=1}^{m-1} x_i \\
 f_{j=2:m-1}(\vec{x}) &= 0.5(1 + g(\vec{y}))(1 - x_{m-j+1}) \prod_{i=1}^{m-j} x_i \\
 f_m(\vec{x}) &= 0.5(1 + g(\vec{y}))(1 - x_1) \\
 \text{donde} \quad y_{i=1:k} &= \{x_m, x_{m+1}, \dots, x_n\} \\
 g(\vec{y}) &= 100 \left\{ k + \sum_{i=1}^k [(y_i - 0.5)^2 - \cos(20\pi(y_i - 0.5))] \right\} \\
 \text{sujeto a} \quad \forall i \in \{1, \dots, n\} & 0 \leq x_i \leq 1.
 \end{aligned} \tag{A.6}$$

Todos los valores objetivo del frente óptimo de Pareto yacen sobre el hiperplano $\sum_{i=1}^m f_i = 0.5$. Las soluciones Pareto óptimas corresponden con $\vec{y} = (0, 0, \dots)^T$. Se sugiere un valor $k = 5$. La dificultad presentada por este problema es converger al hiperplano, puesto que el espacio de búsqueda contiene $(11^k - 1)$ frentes óptimos de Pareto locales que pueden atraer a un optimizador. En la figura A.6 se muestra el frente óptimo de Pareto correspondiente.

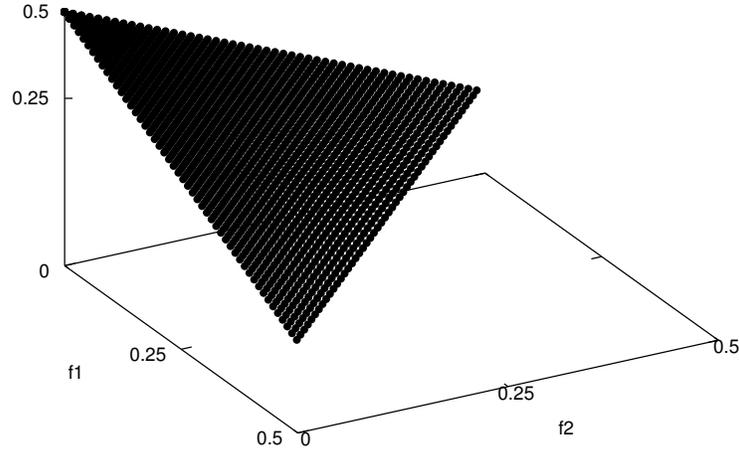


Figura A.6: Frente óptimo de Pareto para DTLZ1 con tres objetivos.

DTLZ2

Este problema se caracteriza por ser unifrontal y separable. Su geometría es cóncava (ver figura A.7). El problema se define a continuación:

$$\begin{aligned}
 \text{Dado} \quad \vec{x} &= (x_1, \dots, x_{m-1}, x_m, \dots, x_n)^T \\
 \text{Minimizar} \quad f_1(\vec{x}) &= (1 + g(\vec{y})) \prod_{i=1}^{m-1} \cos(x_i \pi/2) \\
 f_{j=2:m-1}(\vec{x}) &= (1 + g(\vec{y})) \left(\prod_{i=1}^{m-j} \cos(x_i \pi/2) \right) \sin(x_{m-j+1} \pi/2) \\
 f_m(\vec{x}) &= (1 + g(\vec{y})) \sin(x_1 \pi/2) \\
 \text{donde} \quad y_{i=1:k} &= \{x_m, x_{m+1}, \dots, x_n\} \\
 g(\vec{y}) &= \sum_{i=1}^k (y_i 0.5)^2 \\
 \text{sujeto a} \quad \forall i \in \{1, \dots, n\} & 0 \leq x_i \leq 1.
 \end{aligned} \tag{A.7}$$

Las soluciones Pareto-óptimas corresponden con $\vec{y} = (0.5, 0.5, \dots)^T$ y todos los valores de las funciones objetivo deben satisfacer: $\sum_{i=1}^m f_i^2 = 1$. Se recomienda usar $k = 10$. Esta función es muy útil para investigar la habilidad de un optimizador para escalar su funcionamiento a altas dimensiones en el espacio objetivo.

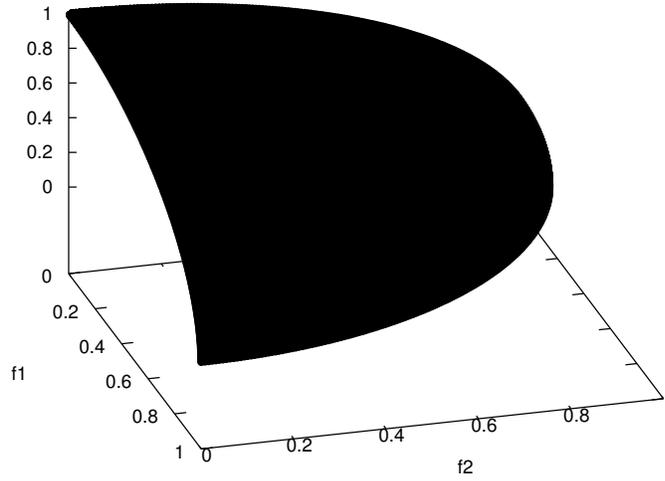


Figura A.7: Frente óptimo de Pareto para DTLZ2 con tres objetivos.

DTLZ3

Este problema simplemente agrega multifractalidad a DTLZ2 mediante la definición de una nueva función g . La formulación es dada como sigue:

$$\text{Dado} \quad \vec{x} = (x_1, \dots, x_{m-1}, x_m, \dots, x_n)^T$$

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

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

$$\text{donde} \quad \begin{aligned} f_m(\vec{x}) &= (1 + g(\vec{y})) \sin(x_1 \pi / 2) \\ y_{i=1:k} &= \{x_m, x_{m+1}, \dots, x_n\} \end{aligned}$$

$$\text{sujeto a} \quad \begin{aligned} g(\vec{y}) &= 100 \left\{ k + \sum_{i=1}^k [(y_i - 0.5)^2 - \cos(20\pi(y_i - 0.5))] \right\} \\ \forall i \in \{1, \dots, n\} & 0 \leq x_i \leq 1. \end{aligned} \tag{A.8}$$

Se sugiere emplear $k = 10$. Al introducir la función g se añaden $(3^k - 1)$ frentes óptimos de Pareto locales y un frente óptimo de Pareto global. Todos los frentes locales son paralelos al frente global. En consecuencia, un optimizador puede estancarse en alguno de ellos antes de lograr convergencia al frente verdadero en $g = 0$. El frente global Pareto-óptimo corresponde con $\vec{y} = (0.5, 0.5, \dots)^T$. El siguiente frente óptimo

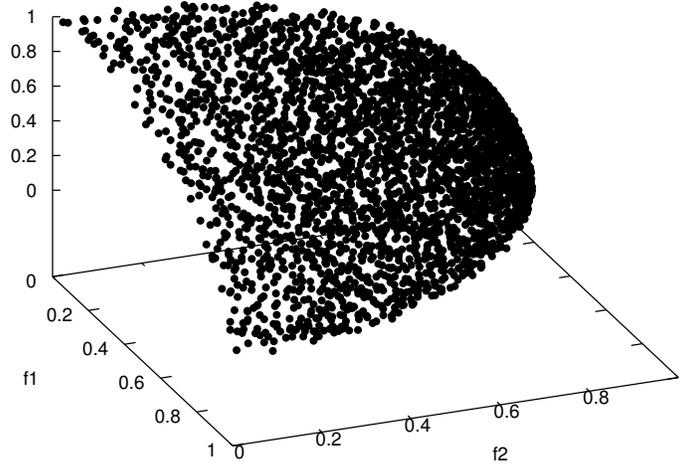


Figura A.8: Frente óptimo de Pareto para DTLZ3 con tres objetivos.

de Pareto global se encuentra en $g = 1$. En la figura A.8 se muestra su frente óptimo de Pareto.

DTLZ4

Este problema es cóncavo, separable y unifrontal (ver figura A.9). Su utilidad principal recae en probar la habilidad de un optimizador para mantener una buena distribución de soluciones, y se define como sigue:

$$\begin{aligned}
 \text{Dado} \quad \vec{x} &= (x_1, \dots, x_{m-1}, x_m, \dots, x_n)^T \\
 \text{Minimizar} \quad f_1(\vec{x}) &= (1 + g(\vec{y})) \prod_{i=1}^{m-1} \cos(x_i^\alpha \pi/2) \\
 f_{j=2:m-1}(\vec{x}) &= (1 + g(\vec{y})) \left(\prod_{i=1}^{m-j} \cos(x_i^\alpha \pi/2) \right) \text{sen}(x_{m-j+1}^\alpha \pi/2) \\
 f_m(\vec{x}) &= (1 + g(\vec{y})) \text{sen}(x_1^\alpha \pi/2) \\
 \text{donde} \quad y_{i=1:k} &= \{x_m, x_{m+1}, \dots, x_n\} \\
 g(\vec{y}) &= \sum_{i=1}^k (y_i - 0.5)^2 \\
 \text{sujeto a} \quad &\forall i \in \{1, \dots, n\} 0 \leq x_i \leq 1.
 \end{aligned} \tag{A.9}$$

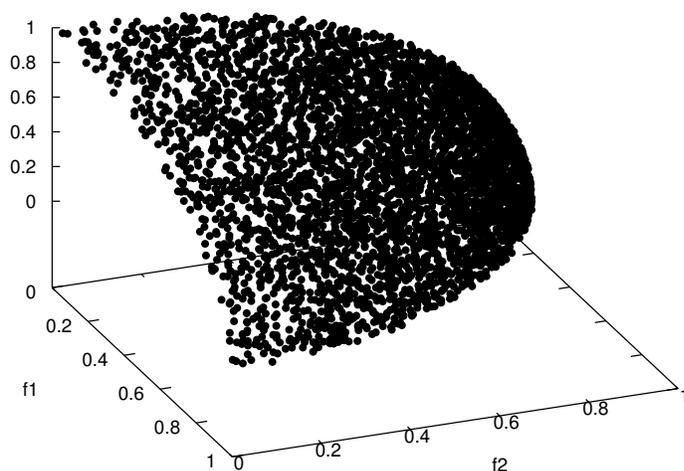


Figura A.9: Frente óptimo de Pareto para DTLZ4 con tres objetivos.

Se sugiere emplear $\alpha = 100$ y $k = 10$. Las modificaciones presentadas por este problema permiten que exista un denso conjunto de soluciones cercanas al plano $f_m - f_1$. Es interesante notar que aunque el espacio de búsqueda tenga una densidad variable de soluciones, los métodos clásicos de sumas ponderadas u otros métodos direccionales no tendrían ninguna dificultad adicional en la resolución de este problema a diferencia de DTLZ2.

DTLZ5

Nuevamente, DTLZ2 es modificado con el objeto de producir un problema unifrontal y degenerado (ver figura A.10). Su definición formal es la siguiente:

$$\begin{aligned}
 \text{Dado} \quad \vec{x} &= (x_1, \dots, x_{m-1}, x_m, \dots, x_n)^T \\
 \text{Minimizar} \quad f_1(\vec{x}) &= (1 + g(\vec{y})) \prod_{i=1}^{m-1} \cos(\theta_i \pi/2) \\
 f_{j=2:m-1}(\vec{x}) &= (1 + g(\vec{y})) \left(\prod_{i=1}^{m-j} \cos(\theta_i \pi/2) \right) \text{sen}(\theta_{m-j+1} \pi/2) \\
 f_m(\vec{x}) &= (1 + g(\vec{y})) \text{sen}(\theta_1 \pi/2) \\
 \text{donde} \quad y_{i=1:k} &= \{x_m, x_{m+1}, \dots, x_n\} \\
 \theta_i &= \begin{cases} x_i & i = 1 \\ \frac{1+2g(\vec{y})}{2(1+g(\vec{y}))} x_i & \forall i \in \{2, 3, \dots, m-1\} \end{cases} \\
 g(\vec{y}) &= \sum_{i=1}^k (y_i - 0.5)^2 \\
 \text{sujeto a} \quad \forall i \in \{1, \dots, n\} & 0 \leq x_i \leq 1.
 \end{aligned} \tag{A.10}$$

Este problema prueba la habilidad de un optimizador para converger hacia una curva y permitirá demostrar visualmente (sólo graficando f_m con otra función objetivo) el desempeño del optimizador. Dado que existe un sesgo natural para las soluciones cercanas a la curva Pareto-óptima, este problema podría ser fácilmente resuelto. El frente óptimo de Pareto corresponde con $\vec{y} = (0.5, 0.5, \dots)^T$, y todos los valores objetivo deben satisfacer $\sum_{i=1}^m f_i^2 = 1$. Se sugiere usar $k = 10$.

DTLZ6

Modificando DTLZ5, un problema más complejo surge como consecuencia de cambiar la función g . El problema resultante es unifrontal, degenerado, presenta sesgo y

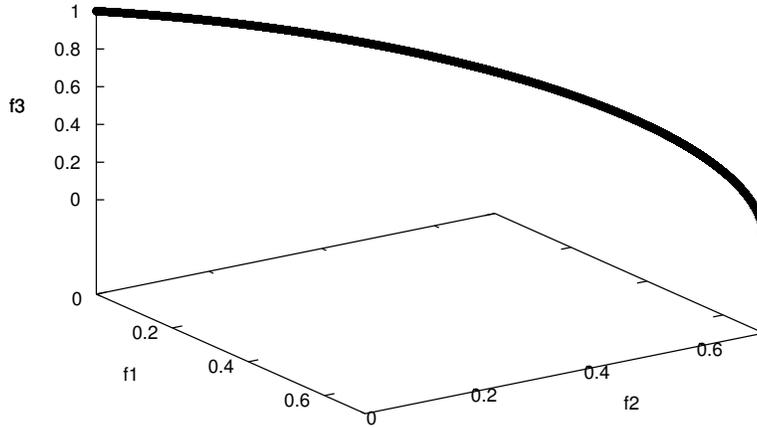


Figura A.10: Frente óptimo de Pareto para DTLZ5 con tres objetivos.

es una transformación muchos-a-uno:

$$\text{Dado} \quad \vec{x} = (x_1, \dots, x_{m-1}, x_m, \dots, x_n)^T$$

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

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

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

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

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

$$g(\vec{y}) = \sum_{i=1}^k y_i^{0.1}$$

$$\text{sujeto a} \quad \forall i \in \{1, \dots, n\} 0 \leq x_i \leq 1.$$

(A.11)

El frente óptimo de Pareto corresponde con $\vec{y} = (0, 0, \dots)^T$ y se muestra en la figura A.11. La modificación impuesta lo vuelve más difícil para un optimizador. La falta de convergencia al frente de Pareto verdadero causa que los optimizadores busquen una superficie no dominada como frente resultante, mientras que el frente óptimo de

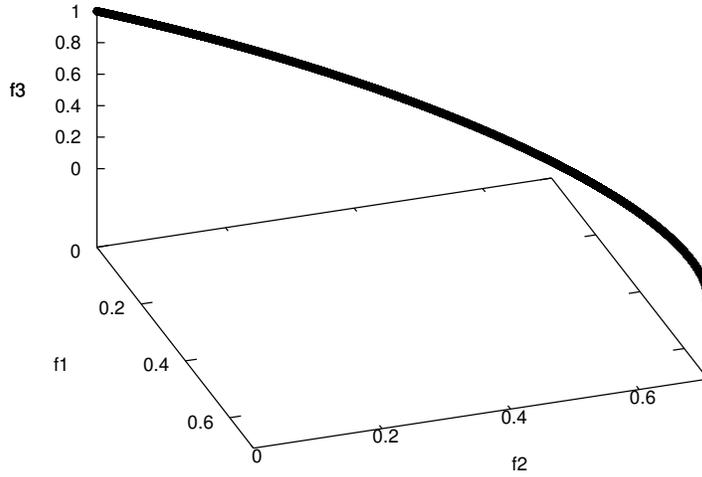


Figura A.11: Frente óptimo de Pareto para DTLZ6 con tres objetivos.

Pareto es una curva. En problemas del mundo real, este comportamiento tiende a proveer información falsa acerca de las propiedades del frente óptimo de Pareto.

DTLZ7

Este problema tiene un conjunto de 2^{m-1} regiones Pareto-óptimas desconectadas en el espacio de búsqueda. La habilidad de mantener subpoblaciones en diferentes regiones es probada por este problema.

$$\text{Dado } \vec{x} = (x_1, \dots, x_{m-1}, x_m, \dots, x_n)^T$$

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

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

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

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

$$\text{sujeto a } \forall i \in \{1, \dots, n\} 0 \leq x_i \leq 1.$$

(A.12)

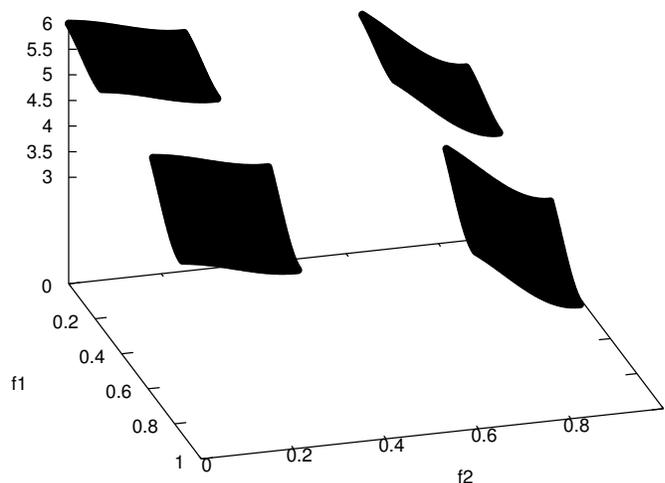


Figura A.12: Frente óptimo de Pareto para DTLZ7 con tres objetivos.

La función g requiere $k = 20$ variables de decisión. Las soluciones Pareto-óptimas corresponden con $\vec{y} = (0, 0, \dots)^T$. El frente óptimo de Pareto se exhibe en la figura [A.12](#).

A.3. Conjunto de problemas Walking-Fish-Group

El kit de problemas de prueba Walking-Fish-Group (WFG) propuesto por Huband et al. [104], sugiere nueve problemas de prueba multi-objetivo (WFG1-WFG9) que pueden ser escalados con respecto al número de objetivos y al número de variables de decisión.

En contraste con otros conjuntos de prueba donde la complejidad de los problemas está fuertemente embebida, el conjunto WFG permite al diseñador de problemas controlar, por medio de una serie de transformaciones compuestas, las características que presentarán los problemas. Con el fin de crear un problema, el diseñador debe seleccionar diferentes geometrías deseables a presentar por la función en el espacio de aptitud, y debe emplear un número de funciones de transformación que faciliten la creación de vectores de transición. Las funciones de transformación deben ser planteadas cuidadosamente de tal forma que el espacio de aptitud asociado (y el frente óptimo de Pareto) se mantengan intactos. El kit WFG provee una gran variedad de formas y transformaciones predefinidas para ayudar a asegurar este aspecto.

A continuación se presenta el banco de problemas de prueba WFG. Aquí, m representa el número de objetivos, y cada problema está definido en términos del

Tabla A.3: Resumen de características del conjunto de problemas de prueba WFG.

Problema	Separabilidad	Frontalidad	Geometría
WFG1	separable	unifrontal	convexo, mixto
WFG2	no separable	multifrontal	convexo, desconectado
WFG3	no separable	unifrontal	lineal, degenerado
WFG4	separable	multifrontal	cóncavo
WFG5	separable	deceptivo	cóncavo
WFG6	no separable	unifrontal	cóncavo
WFG7	separable	unifrontal	cóncavo
WFG8	no separable	unifrontal	cóncavo
WFG9	no separable	multifrontal	cóncavo

vector de parámetros subyacente $\vec{x} \in \mathbb{R}^m$ que define el espacio de aptitud. Todos los $x_i \in \vec{x}$ tiene dominio $[0, 1]$. x_m es conocido como parámetro de distancia, y $x_{1:m-1}$ son los parámetros de posición. El vector \vec{x} es derivado, a través de una serie de vectores de transición, de un vector de parámetros de trabajo $\vec{z} \in \mathbb{R}^n$ (conocido como vector de variables). El dominio de todas las variables $z_i \in \vec{z}$ es $[0, 2i]$. Cabe mencionar que $n \geq m$ y $n = k + l$. Los primeros $k \in \{m - 1, 2(m - 1), 3(m - 1), \dots\}$ parámetros son los relacionados con la posición y los últimos $l \in \{1, 2, \dots\}$ parámetros son los relacionados con la distancia. Cada vector de transición añade mayor complejidad al problema inherente. El optimizador manipula directamente \vec{z} , a través de lo cual \vec{x} es indirectamente operado.

WFG1

Este problema es separable y unifrontal, pero cuenta con una región plana polinomial (ver figura A.13). Existe un fuerte sesgo hacia pequeños valores de las variables, lo que lo convierte en un problema muy difícil para algunos optimizadores. Su defini-

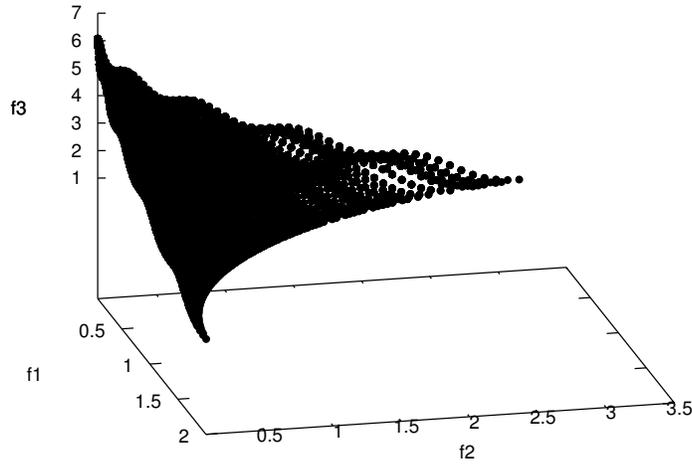


Figura A.13: Frente óptimo de Pareto para WFG1 con tres objetivos.

ción es la siguiente:

$$\text{Dado} \quad \vec{z} = (z_1, \dots, z_k, z_{k+1}, \dots, z_n)^T$$

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

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

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

$$\begin{aligned} \text{donde} \quad x_{i=1:m-1} &= \text{r_sum}(\{y_{(i-1)k/(m-1)+1}, \dots, y_{ik/(m-1)}\}, \\ &\quad \{2(i-1)k/(m-1) + 1, \dots, 2ik/(m-1)\}) \\ x_m &= \text{r_sum}(\{y_{k+1}, \dots, y_n\}, \{2(k+1), \dots, 2n\}) \\ y_{i=1:n} &= \text{b_poly}(y'_i, 0.02) \\ y'_{i=1:k} &= y''_i \\ y'_{i=k+1:n} &= \text{b_flat}(y''_i, 0.8, 0.75, 0.85) \\ y''_{i=1:k} &= z_i / (2i) \\ y''_{i=k+1:n} &= \text{s_linear}(z_i / (2i), 0.35) \end{aligned}$$

(A.13)

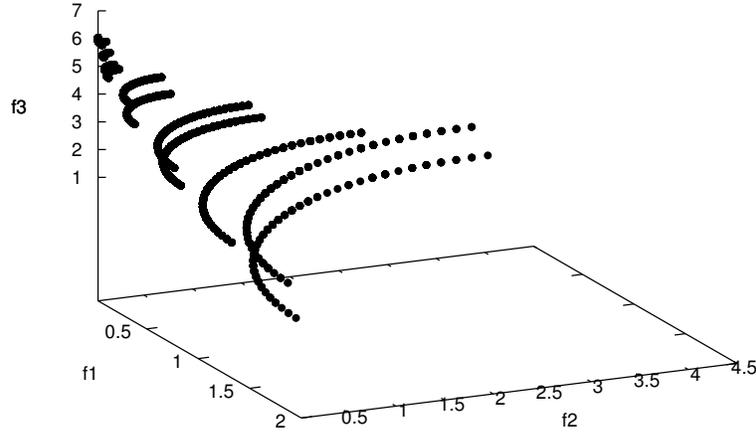


Figura A.14: Frente óptimo de Pareto para WFG2 con tres objetivos.

WFG2

Este problema es no separable y multifrontal. El frente óptimo de Pareto tiene una geometría desconectada (ver figura A.14). Su definición matemática es la siguiente:

$$\begin{aligned}
 \text{Dado} \quad \vec{z} &= (z_1, \dots, z_k, z_{k+1}, \dots, z_n)^T \\
 \text{Minimizar} \quad f_1(\vec{x}) &= x_m + 2 \prod_{i=1}^{m-1} (1 - \cos(x_i \pi / 2)) \\
 f_{j=2:m-1}(\vec{x}) &= x_m + 2j \left[\prod_{i=1}^{m-j} (1 - \cos(x_i \pi / 2)) \right] (1 - \sin(x_{m-j+1} \pi / 2)) \\
 f_m(\vec{x}) &= x_m + 2m(1 - x_1 \cos^2(5x_1 \pi)) \\
 \text{donde} \quad x_{i=1:m-1} &= \text{r_sum} \left(\left\{ y_{(i-1)k/(m-1)+1}, \dots, y_{ik/(m-1)} \right\}, \{1, \dots, 1\} \right) \\
 x_m &= \text{r_sum} \left(\left\{ y_{k+1}, \dots, y_{k+l/2} \right\}, \{1, \dots, 1\} \right) \\
 y_{i=1:k} &= y'_i \\
 y_{i=k+1:k+l/2} &= \text{r_nonsep} \left(\left\{ y'_{k+2(i-k)-1}, y'_{k+2(i-k)} \right\}, 2 \right) \\
 y'_{i=1:k} &= z_i / (2i) \\
 y'_{i=k+1:n} &= \text{s_linear}(z_i / (2i), 0.35)
 \end{aligned} \tag{A.14}$$

WFG3

Este problema es no separable y unifrontal. Su frente óptimo de Pareto está representado por una variedad geométrica degenerada y lineal. El problema se formaliza a continuación:

$$\begin{aligned}
\text{Dado} \quad \vec{z} &= (z_1, \dots, z_k, z_{k+1}, \dots, z_n)^T \\
\text{Minimizar} \quad f_1(\vec{x}) &= x_m + 2 \prod_{i=1}^{m-1} x_i \\
f_{j=2:m-1}(\vec{x}) &= x_m + 2j \left(\prod_{i=1}^{m-j} x_i \right) (1 - x_{m-j+1}) \\
f_m(\vec{x}) &= x_m + 2m(1 - x_1)
\end{aligned}$$

donde

$$\begin{aligned}
x_{i=1} &= u_i \\
x_{i=2:m-1} &= x_m(u_i - 0.5) + 0.5 \\
x_m &= \text{r_sum} \left(\{y_{k+1}, \dots, y_{k+l/2}\}, \{1, \dots, 1\} \right) \\
u_i &= \text{r_sum} \left(\{y_{(i-1)k/(m-1)+1}, \dots, y_{ik/(m-1)}\}, \{1, \dots, 1\} \right) \\
y_{i=1:k} &= y'_i \\
y_{i=k+1:k+l/2} &= \text{r_nonsep} \left(\{y'_{k+2(i-k)-1}, y'_{k+2(i-k)}\}, 2 \right) \\
y'_{i=1:k} &= z_i / (2i) \\
y'_{i=k+1:n} &= \text{s_linear}(z_i / (2i), 0.35)
\end{aligned} \tag{A.15}$$

En la figura [A.15](#) se presenta el frente óptimo de Pareto.

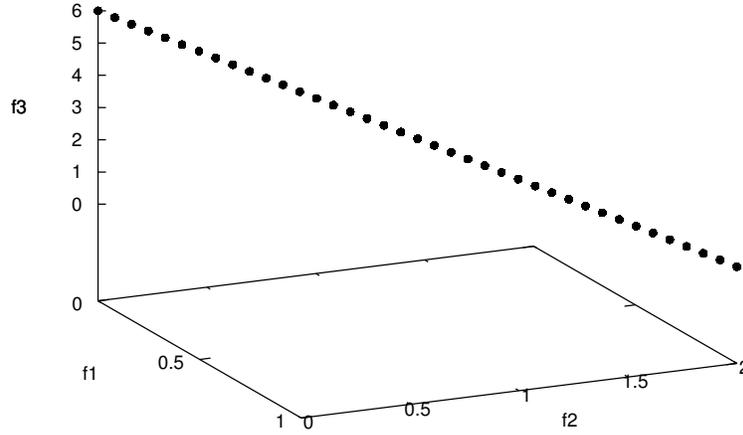


Figura A.15: Frente óptimo de Pareto para WFG3 con tres objetivos.

WFG4

Este problema es separable pero altamente multifrontal. La geometría del frente óptimo de Pareto es cóncava (ver figura A.16) y se define de la siguiente forma:

$$\text{Dado} \quad \vec{z} = (z_1, \dots, z_k, z_{k+1}, \dots, z_n)^T$$

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

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

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

$$\text{donde} \quad x_{i=1:m-1} = \text{r_sum}(\{y_{(i-1)k/(m-1)+1}, \dots, y_{ik/(m-1)}\}, \{1, \dots, 1\})$$

$$x_m = \text{r_sum}(\{y_{k+1}, \dots, y_n\}, \{1, \dots, 1\})$$

$$y_{i=1:n} = \text{s_multi}(z_i / (2i), 30, 10, 0.35)$$

(A.16)

WFG5

Este problema se caracteriza por ser engañoso y separable. La única diferencia con respecto a WFG4 es el uso de la transformación $\text{s_decept}(\cdot)$. El frente óptimo de

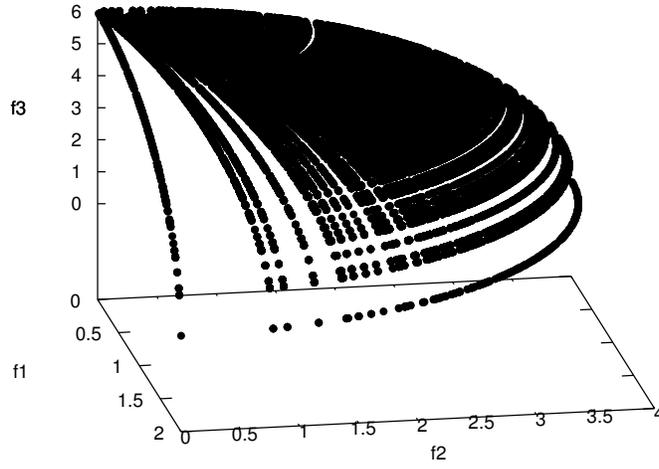


Figura A.16: Frente óptimo de Pareto para WFG4 con tres objetivos.

Pareto es cóncavo (ver figura A.17) y la siguiente expresión lo define:

$$\text{Dado} \quad \vec{z} = (z_1, \dots, z_k, z_{k+1}, \dots, z_n)^T$$

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

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

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

$$\begin{aligned} \text{donde} \quad x_{i=1:m-1} &= \text{r_sum}(\{y_{(i-1)k/(m-1)+1}, \dots, y_{ik/(m-1)}\}, \{1, \dots, 1\}) \\ x_m &= \text{r_sum}(\{y_{k+1}, \dots, y_n\}, \{1, \dots, 1\}) \\ y_{i=1:n} &= \text{s_decept}(z_i / (2i), 0.35, 0.001, 0.05) \end{aligned} \tag{A.17}$$

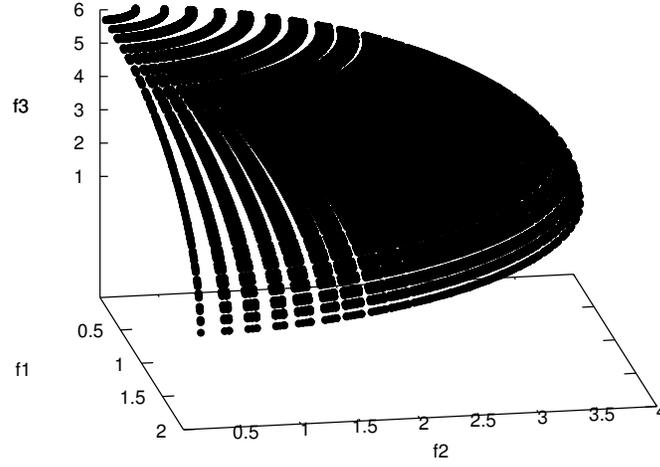


Figura A.17: Frente óptimo de Pareto para WFG5 con tres objetivos.

WFG6

Este problema es no separable y unifrontal. Su frente óptimo de Pareto es cóncavo (ver figura A.18), y se define como sigue:

$$\text{Dado} \quad \vec{z} = (z_1, \dots, z_k, z_{k+1}, \dots, z_n)^T$$

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

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

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

$$\text{donde} \quad x_{i=1:m-1} = \text{r_nonsep}(\{y_{(i-1)k/(m-1)+1}, \dots, y_{ik/(m-1)}\}, k/(m-1))$$

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

$$y_{i=1:k} = z_i / (2i)$$

$$y_{i=k+1:n} = \text{s_linear}(z_i / (2i), 0.35)$$

(A.18)

WFG7

Este problema es separable y unifrontal. Además, presenta un sesgo en la dependencia de parámetros. La geometría del frente óptimo de Pareto es cóncava (ver figura

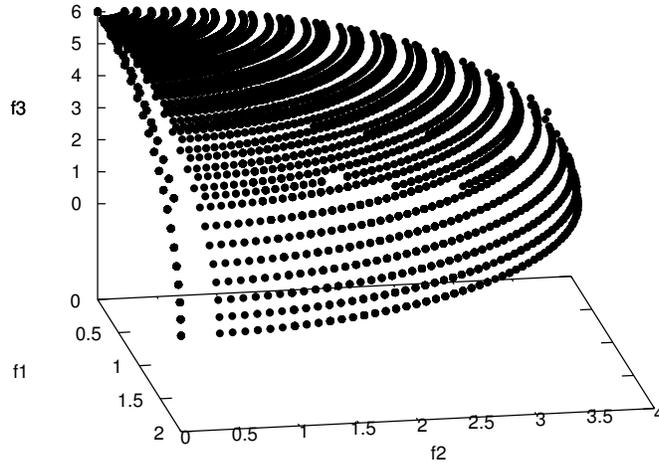


Figura A.18: Frente óptimo de Pareto para WFG6 con tres objetivos.

A.19). El problema se expresa matemáticamente a continuación:

Dado $\vec{z} = (z_1, \dots, z_k, z_{k+1}, \dots, z_n)^T$

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

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

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

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

$$x_m = \text{r_sum}(\{y_{k+1}, \dots, y_n\}, \{1, \dots, 1\})$$

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

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

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

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

(A.19)

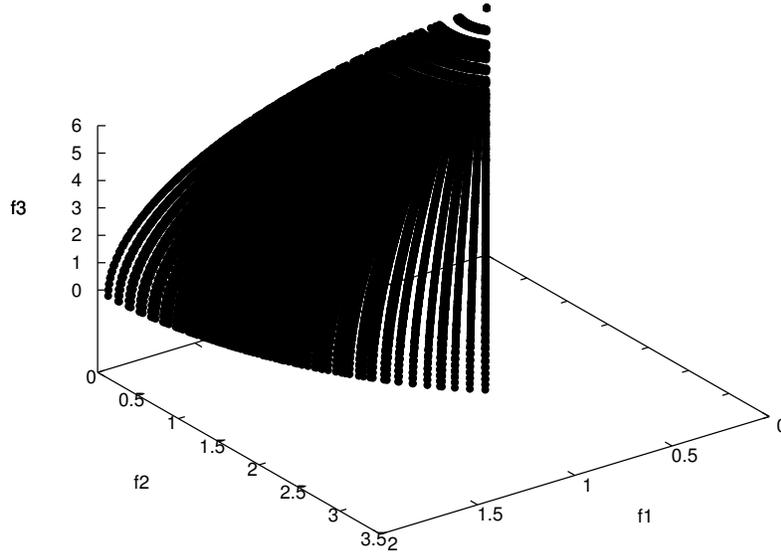


Figura A.19: Frente óptimo de Pareto para WFG7 con tres objetivos.

WFG8

Este problema cuenta también con un sesgo de dependencia en los parámetros, pero también es no separable y unifrontal. El frente óptimo de Pareto tiene geometría cóncava (ver figura A.20). El problema se describe de la forma siguiente:

$$\text{Dado} \quad \vec{z} = (z_1, \dots, z_k, z_{k+1}, \dots, z_n)^T$$

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

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

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

$$\begin{aligned} \text{donde} \quad x_{i=1:m-1} &= \text{r_sum}(\{y_{(i-1)k/(m-1)+1}, \dots, y_{ik/(m-1)}\}, \{1, \dots, 1\}) \\ x_m &= \text{r_sum}(\{y_{k+1}, \dots, y_n\}, \{1, \dots, 1\}) \\ y_{i=1:k} &= y'_i \\ y_{i=k+1:n} &= \text{s_linear}(y'_i, 0.35) \\ y'_{i=k} &= z_i / (2i) \\ y'_{i=k+1:1} &= \text{b_param}(z_i / (2i), \text{r_sum}(\{z_1/2, \dots, z_{i-1}/(2(i-1))\}, \{1, \dots, 1\}), \frac{0.98}{49.98}, 0.02, 50) \end{aligned} \tag{A.20}$$

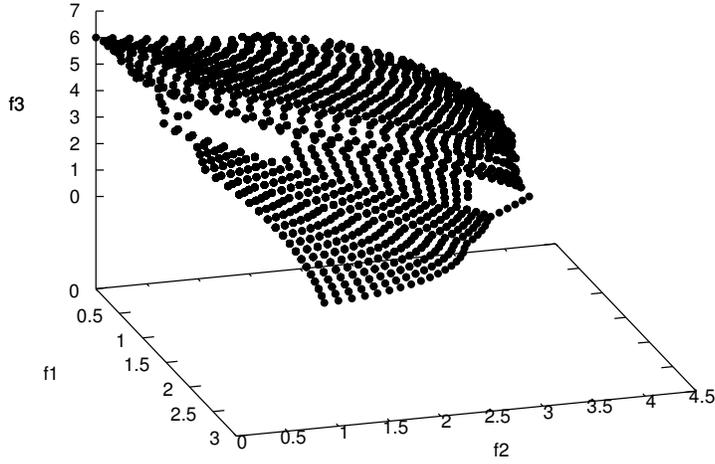


Figura A.20: Frente óptimo de Pareto para WFG8 con tres objetivos.

WFG9

Este problema es no separable, multifrontal, deceptivo y tiene un sesgo de dependencia en los parámetros. Debido a estas características, WFG9 es un problema muy difícil de resolver. Su frente óptimo de Pareto (cóncavo) se muestra en la figura A.21, mientras que su definición matemática se da a continuación:

$$\text{Dado} \quad \vec{z} = (z_1, \dots, z_k, z_{k+1}, \dots, z_n)^T$$

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

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

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

$$\begin{aligned} \text{donde} \quad x_{i=1:m-1} &= \text{r_nonsep}(\{y_{(i-1)k/(m-1)+1}, \dots, y_{ik/(m-1)}\}, k/(m-1)) \\ x_m &= \text{r_nonsep}(\{y_{k+1}, \dots, y_n\}, l) \\ y_{i=1:k} &= \text{s_decep}(y'_i, 0.35, 0.001, 0.05) \\ y_{i=k+1:n} &= \text{s_multi}(y'_i, 30, 95, 0.35) \\ y'_{i=1:n-1} &= \text{b_param}(z_i/(2i), \text{r_sum}(\{z_{i+1}/(2(i+1)), \dots, z_n/(2n)\}, \\ &\quad \{1, \dots, 1\}), \frac{0.98}{49.98}, 0.02, 50) \\ y'_n &= z_n/(2n) \end{aligned}$$

(A.21)

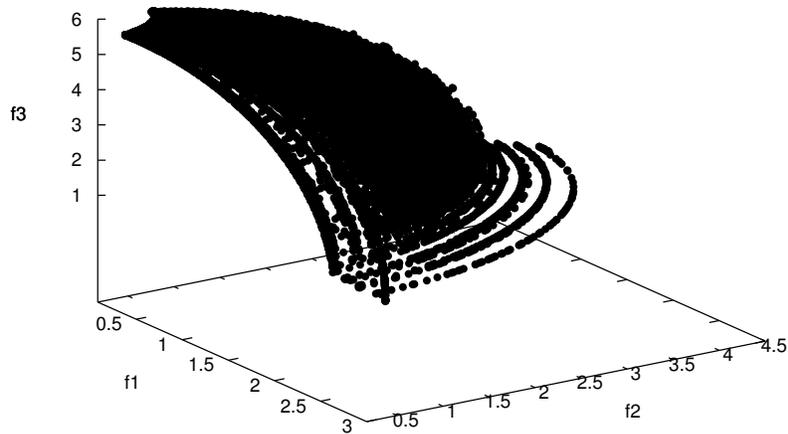


Figura A.21: Frente óptimo de Pareto para WFG9 con tres objetivos.

Funciones de transformación

Los problemas descritos previamente están definidos en términos de algunas funciones de transformación, las cuales mapean los parámetros con dominio $[0, 1]$ al rango $[0, 1]$. Existen tres tipos de funciones de transformación: **sesgo**, **desplazamiento** y **reducción**. Las funciones de sesgo y desplazamiento sólo emplean un parámetro, mientras que la función de reducción puede emplear muchos. Las transformaciones de sesgo tienen un impacto natural en el proceso de búsqueda al sesgar el paisaje de aptitud. Por su parte, las transformaciones de desplazamiento mueven la localización de los valores óptimos, y son usadas para aplicar un desplazamiento lineal, o para producir problemas deceptivos y multifrontales. Finalmente, las transformaciones de reducción son utilizadas para producir el efecto de no separabilidad de los problemas (dependencia entre las variables). En lo subsecuente se definen dichas funciones de transformación.

Sesgo polinomial

Cuando $\alpha > 1$ o cuando $\alpha < 1$, y está sesgada hacia cero o hacia uno, respectivamente.

$$\text{b_poly}(y, \alpha) = y^\alpha \quad (\text{A.22})$$

donde $\alpha > 0$ y $\alpha \neq 1$.

Sesgo: regiones planas

Los valores de y entre B y C , el área de la región plana, son transformados al valor A .

$$\text{b_flat}(y, A, B, C) = A + \min(0, \lfloor y - B \rfloor) \cdot \frac{A(B-y)}{B} - \min(0, \lfloor C - y \rfloor) \cdot \frac{(1-A)(y-C)}{1-C} \quad (\text{A.23})$$

donde $A, B, C \in [0, 1]$, $B < C$, $B = 0 \Rightarrow A = 0 \wedge C \neq 1$, y $C = 1 \Rightarrow A = 1 \wedge B \neq 0$.

Sesgo: dependencia de parámetros

Los valores A, B, C , el vector de parámetros $\vec{w} \in [0, 1]^{|\vec{w}|}$, y la función de reducción u , determinan, en conjunto, el grado en el que y es sesgada mediante su elevación a la potencia asociada: valores de $u(\vec{w}) \in [0, 0.5]$ son transformados linealmente sobre $[B, B + (C - B)A]$, y valores de $u(\vec{w}) \in [0.5, 1]$ son mapeados linealmente sobre $[B + (C - B)A, C]$.

$$\text{b_param}(y, u(\vec{w}), A, B, C) = y^{B+(C-B)(A-(1-2u(\vec{w}))|0.5-u(\vec{w})+A|)} \quad (\text{A.24})$$

donde $A \in (0, 1)$, y $0 < B < C$.

Desplazamiento lineal

$A \in (0, 1)$ es el valor para el cual y es transformada en cero.

$$\text{s_linear}(y, A) = \frac{|y - A|}{|\lfloor A - y \rfloor + A|} \quad (\text{A.25})$$

Desplazamiento deceptivo

A es el valor en el cual y es transformada a cero, y el mínimo global de la transformación. B es el tamaño de «apertura» de la cuenca que conduce al mínimo global en A , y C es el valor del mínimo deceptivo (hay siempre dos mínimos deceptivos).

$$\text{s_decept}(y, A, B, C) = 1 + (|y - A| - B) \left(\frac{\lfloor y - A + B \rfloor (1 - C + \frac{A-B}{B})}{A-B} + \frac{\lfloor A + B - y \rfloor (1 - C + \frac{1-A-B}{B})}{1-A-B} + \frac{1}{B} \right) \quad (\text{A.26})$$

donde $A \in (0, 1)$, $0 < B \ll 1$, $0 < C \ll 1$, $A - B > 0$, y $A + B < 1$.

Desplazamiento multimodal

A controla el número de mínimos, B controla la magnitud de los «tamaños de las cimas» de la multimodalidad, y C es el valor para el cual y se transforma en cero, y

cuando $B \neq 0$, hay $2A$ mínimos locales, y un mínimo global en C . Valores grandes de A y valores pequeños de B generan más dificultad en los problemas.

$$s_multi(y, A, B, C) = \frac{1 + \cos \left[(4A + 2)\pi \left(0.5 - \frac{|y-C|}{2(\lfloor C-y \rfloor + C)} \right) \right] + 4B \left(\frac{|y-C|}{2(\lfloor C-y \rfloor + C)} \right)^2}{B + 2} \quad (\text{A.27})$$

donde $A \in \{1, 2, \dots\}$, $B \geq 0$, $(4A + 2)\pi > 4B$, y $C \in (0, 1)$.

Reducción: sumas ponderadas

Mediante la variación de las constantes del vector de pesos \vec{w} , los optimizadores pueden ser forzados a tratar los parámetros de forma diferente.

$$r_sum(\vec{y}, \vec{w}) = \frac{\sum_{i=1}^{|\vec{y}|} w_i y_i}{\sum_{i=1}^{|\vec{y}|} w_i} \quad (\text{A.28})$$

donde $|\vec{w}| = |\vec{y}|$, y $w_1, \dots, w_{|\vec{y}|} > 0$.

Reducción: no separabilidad

A controla el grado de no separabilidad.

$$r_nonsep(\vec{y}, A) = \frac{\sum_{j=1}^{|\vec{y}|} \left(y_j + \sum_{k=0}^{A-2} |y_j - y_{1+(j+k) \bmod |\vec{y}|}| \right)}{\frac{|\vec{y}|}{A} \lceil A/2 \rceil (1 + 2A - 2 \lceil A/2 \rceil)} \quad (\text{A.29})$$

donde $A \in \{1, \dots, |\vec{y}|\}$, y $|\vec{y}| \bmod A = 0$

A.4. Resumen

Con el propósito de conocer las fortalezas y debilidades de una nueva metaheurística multi-objetivo, es importante contar con un buen conocimiento acerca del problema que se está resolviendo. En consecuencia, en la literatura especializada se han propuesto una gran cantidad de problemas artificiales de prueba que presentan diferentes propiedades.

En este apéndice, se han descrito tres importantes conjuntos de problemas de prueba: el conjunto ZDT que ofrece seis problemas que han tenido una relevancia significativa en el campo de la optimización evolutiva multi-objetivo; el conjunto DTLZ, el cual incluye siete problemas sin restricciones con varias características en los frentes óptimos de Pareto asociados; y el banco de problemas de prueba WFG que incluye nueve problemas cubriendo una gama interesante de geometrías y paisajes de aptitud, tales como sesgo, multifrontalidad y no separabilidad. Es importante resaltar que tanto DTLZ como WFG son conjuntos de prueba extensamente empleados en el

campo de la optimización evolutiva multi-objetivo debido a sus capacidades para escalar en el número de funciones objetivo y número de variables de decisión.

A pesar de que los problemas ZDT han sido ampliamente usados en el área de optimización evolutiva multi-objetivo, una importante desventaja es que tales problemas no pueden ser escalados en el espacio de las funciones objetivo. Adicionalmente, los POMs continuos del conjunto ZDT no exhiben características deseables como: no separabilidad, deceptividad o frentes degenerados.

Por otra parte, un aspecto relevante del conjunto de problemas DTLZ es que se conocen las formas y localización de los frentes óptimos de Pareto. Además, todos los problemas son escalables en el número de objetivos. No obstante, para DTLZ5 y DTLZ6 en altas dimensiones en el espacio de las funciones objetivo no se conocen con certeza los frentes asociados. Otro aspecto negativo acerca de los problemas DTLZ es que características tales como regiones planas, deceptibilidad y no separabilidad no están presentes.

Para solucionar las deficiencias existentes en los conjuntos ZDT y DTLZ, Huband et al. [104] crearon el banco de problemas de prueba WFG. Estos problemas resuelven las inconvenientes anteriormente descritos mediante la implementación de funciones de transformación que cubren un amplio rango de formas y geometrías deseables para un POM. En consecuencia, este conjunto es mucho más difícil para un optimizador, especialmente para aquellos basados en estrategias «escalando la colina».

Apéndice B

Resultados numéricos

En este apéndice se presenta un resumen de los resultados numéricos obtenidos de los experimentos hechos en el capítulo 5. Los datos están organizados en tablas y figuras de la siguiente manera:

- **Tablas de comparación de optimizadores:** muestran los valores de los indicadores de desempeño de cada optimizador, haciendo énfasis en los resultados estadísticamente significativos con respecto a iMOACO_R a través de la prueba de suma de rangos de Wilcoxon de una cola con un nivel del 5 %.
- **Gráficas de aproximación para iMOACO_R:** se presentan las aproximaciones a los frentes de Pareto obtenidos por iMOACO_R para dos y tres dimensiones. Para alta dimensionalidad, se emplean las gráficas de coordenadas paralelas para presentar los conjuntos de aproximación.
- **Diagramas de cajas:** la información estadística proveniente del análisis de sensibilidad de iMOACO_R a sus parámetros se representa mediante este tipo de diagramas.
- **Diagramas de intervalos de confianza:** son empleados para ilustrar los intervalos de confianza generados por la prueba Tukey a las parejas de combinaciones de parámetros.

B.1. Resultados del experimento 1

Los datos del experimento 1 son organizados por tablas donde cada una de éstas corresponde con un conjunto de problemas de prueba para una dimensión del espacio objetivo en particular. Cada tabla contiene la información de la media y desviación estándar (tal valor se muestra entre paréntesis) por cada indicador de desempeño (HV, IGD+ y S). Cuando un optimizador no presenta convergencia al frente óptimo de Pareto, sus valores estadísticos de hipervolumen se establecen en cero, mientras que los valores de los indicadores restantes se marcan como N/A (No Aplica). El símbolo

se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de $iMOACO_{\mathbb{R}}$ en la prueba Wilcoxon con un nivel del 5 %. Los dos mejores valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor.

Los frentes de Pareto generados por $iMOACO_{\mathbb{R}}$ para los conjuntos DTLZ y WFG son mostrados mediante gráficas de coordenadas paralelas en todas las dimensiones del espacio objetivo consideradas. Asimismo, se hace uso de gráficas en tres dimensiones de estos problemas para presentar la distribución obtenida. Las aproximaciones graficadas corresponden con la ejecución que representa la mediana del valor del hipervolumen. En lo que respecta al conjunto de problemas ZDT, se muestran los frentes producidos por la propuesta en gráficas convencionales.

Tabla B.1: Comparación de $iMOACO_{\mathbb{R}}$ con respecto a SMS-EMOA, MOEA/D, NSGA-III y $MOACO_{\mathbb{R}}$ en los problemas ZDT con 2 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de $iMOACO_{\mathbb{R}}$ en la prueba Wilcoxon con un nivel del 5 %. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor.

Problema	Algoritmo	HV	IGD+	S
ZDT1	$iMOACO_{\mathbb{R}}$	1.206517e+02(2.464999e-03)	6.469204e-03(3.167582e-04)	2.111221e-02(8.120666e-04)
	$MOACO_{\mathbb{R}}$	1.206577e+02(8.404663e-04)	3.405466e-03(1.994861e-04)	5.164295e-03(4.822033e-04)
	MOEA/D	1.205004e+02(1.897502e-01)#	3.308778e-03(1.727701e-03)	4.282478e-03(3.285489e-04)
	NSGA-III	1.206616e+02(5.024979e-04)	2.003075e-03(1.528828e-05)	8.593820e-03(1.557212e-04)
	SMS-EMOA	1.150137e+02(1.678361e-01)#	1.115959e-01(5.714445e-03)#	1.846902e-03(3.117669e-04)
ZDT2	$iMOACO_{\mathbb{R}}$	1.199755e+02(1.852404e+00)	1.530710e-02(5.911894e-02)	1.997437e-02(4.041831e-03)
	$MOACO_{\mathbb{R}}$	0.000000e+00(0.000000e+00)	N/A	N/A
	MOEA/D	1.202755e+02(2.758723e-02)#	3.053087e-03(5.827056e-04)	3.766202e-03(1.753340e-04)
	NSGA-III	1.203270e+02(1.155044e-03)	1.950337e-03(9.986811e-06)	3.484849e-03(2.206813e-05)
	SMS-EMOA	1.115536e+02(3.107275e-01)#	1.554463e-01(9.711778e-03)#	2.897028e-03(5.322089e-04)
ZDT3	$iMOACO_{\mathbb{R}}$	1.287334e+02(8.586669e-03)	2.310313e-03(1.026454e-04)	1.681344e-02(7.110032e-04)
	$MOACO_{\mathbb{R}}$	1.287587e+02(3.871777e-03)	1.884608e-03(1.154487e-04)	5.801513e-03(5.356396e-04)
	MOEA/D	1.279497e+02(1.166202e+00)#	7.346613e-03(8.369429e-03)#	1.464451e-02(1.145940e-03)
	NSGA-III	1.287742e+02(3.906680e-04)	1.437305e-03(6.267913e-05)	1.077938e-02(9.742736e-04)
	SMS-EMOA	1.255758e+02(1.813206e+00)#	3.201085e-02(2.515934e-02)#	2.619180e-02(1.523686e-02)#
ZDT4	$iMOACO_{\mathbb{R}}$	0.000000e+00(0.000000e+00)	N/A	N/A
	$MOACO_{\mathbb{R}}$	0.000000e+00(0.000000e+00)	N/A	N/A
	MOEA/D	1.167092e+02(1.937481e+00)	1.982227e-01(1.206844e-01)	7.139591e-03(2.076064e-03)
	NSGA-III	1.206437e+02(1.497950e-02)	3.046360e-03(9.898448e-04)	8.740503e-03(1.739471e-04)
	SMS-EMOA	7.424215e+01(1.003982e+01)	3.883311e+00(8.920886e-01)	1.384897e-02(1.531130e-03)
ZDT6	$iMOACO_{\mathbb{R}}$	1.158962e+02(5.536475e-01)	1.174671e-01(4.397162e-02)	7.457689e-03(6.422928e-03)
	$MOACO_{\mathbb{R}}$	0.000000e+00(0.000000e+00)	N/A	N/A
	MOEA/D	1.159309e+02(1.333049e-01)	1.113708e-01(9.958626e-03)	3.920564e-03(1.079377e-03)
	NSGA-III	1.163957e+02(3.204507e-03)	4.328543e-03(1.278529e-04)	1.289052e-03(4.765128e-05)
	SMS-EMOA	1.149142e+02(6.692734e-01)#	1.393283e-01(3.631004e-02)#	2.277363e-02(6.534523e-02)

Tabla B.2: Comparación de iMOACO_R con respecto a SMS-EMOA, MOEA/D, NSGA-III y MOACO_R en los problemas DTLZ con 3 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de iMOACO_R en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor.

Problema	Algoritmo	HV	IGD+	S
DTLZ1	iMOACO _R	0.000000e+00(0.000000e+00)	N/A	N/A
	MOACO _R	0.000000e+00(0.000000e+00)	N/A	N/A
	MOEA/D	9.740945e-01(2.619792e-04)	1.352432e-02(7.357446e-04)	5.665252e-04(2.658619e-04)
	NSGA-III	9.741172e-01(3.045766e-04)	1.354202e-02(9.167630e-04)	9.639678e-03(1.597502e-02)
	SMS-EMOA	6.766570e-01(1.240803e-02)	1.044670e-01(2.295023e-03)	4.668093e-03(3.088185e-03)
DTLZ2	iMOACO _R	7.420231e+00(2.561773e-04)	2.073739e-02(1.372111e-04)	5.256836e-02(1.370626e-03)
	MOACO _R	7.393920e+00(5.114470e-03)#	2.857186e-02(1.446421e-03)#	4.739753e-02(4.342241e-03)
	MOEA/D	7.421715e+00(1.371960e-04)	1.995184e-02(1.108210e-05)	4.890725e-02(3.032328e-05)
	NSGA-III	7.421572e+00(6.064713e-04)	2.026117e-02(3.115966e-04)	4.842788e-02(8.740667e-04)
	SMS-EMOA	4.111960e+00(9.221669e-02)#	3.271528e-01(7.180339e-03)#	1.478099e-02(4.202561e-03)
DTLZ3	iMOACO _R	0.000000e+00(0.000000e+00)	N/A	N/A
	MOACO _R	0.000000e+00(0.000000e+00)	N/A	N/A
	MOEA/D	3.420659e+02(1.134118e+00)	8.482280e-02(1.843136e-01)	5.521024e-02(3.130568e-02)
	NSGA-III	3.417475e+02(1.59568e+00)	1.769805e-01(3.064822e-01)	1.316527e-01(8.429412e-02)
	SMS-EMOA	1.420606e+04(4.967281e+04)	4.360171e-01(5.285901e-01)	7.033163e+00(1.058432e+01)
DTLZ4	iMOACO _R	7.419261e+00(9.233958e-04)	3.193750e-02(5.791632e-04)	6.107899e-02(3.996887e-03)
	MOACO _R	7.397218e+00(4.404488e-03)#	3.724734e-02(1.161094e-03)#	4.797831e-02(3.209891e-03)
	MOEA/D	7.421635e+00(1.148142e-04)	2.996432e-02(7.542395e-06)	4.891094e-02(2.596731e-05)
	NSGA-III	7.218780e+00(4.062936e-01)	6.666353e-02(7.313550e-02)	4.074251e-02(1.582941e-02)
	SMS-EMOA	4.478782e+00(3.855345e-01)#	2.445605e-01(6.890561e-02)#	2.099415e-02(1.977312e-02)
DTLZ5	iMOACO _R	5.983806e+01(7.194011e-03)	2.129263e-03(3.711193e-04)	4.268713e-03(3.265504e-03)
	MOACO _R	5.986894e+01(1.342001e-03)	1.225832e-03(2.675554e-04)	7.027390e-03(5.236215e-04)#
	MOEA/D	5.973486e+01(1.291159e-03)#	4.703606e-03(1.009971e-05)#	2.164089e-01(5.072710e-03)#
	NSGA-III	5.982962e+01(8.365012e-03)#	1.694491e-03(4.522151e-04)	1.239829e-02(1.881406e-03)#
	SMS-EMOA	5.027973e+01(5.019487e-01)#	1.629077e-03(4.585896e-04)	6.198692e-03(1.705874e-03)#
DTLZ6	iMOACO _R	1.315942e+03(1.175215e+00)	1.111736e-01(4.320450e-02)	7.331618e-02(7.146915e-03)
	MOACO _R	1.318611e+03(1.107433e+00)	2.407273e-02(5.620857e-02)	1.623979e-02(2.086520e-02)
	MOEA/D	1.316428e+03(8.424796e-01)	1.256803e-01(3.448519e-02)	2.380922e-01(2.572184e-03)#
	NSGA-III	1.316830e+03(4.802867e-01)	1.257182e-01(2.720034e-02)	6.772126e-02(4.153498e-02)
	SMS-EMOA	1.186353e+03(1.891027e+01)#	2.437784e-01(4.500974e-02)#	1.313191e-02(3.313168e-03)
DTLZ7	iMOACO _R	1.624747e+01(3.431217e-02)	8.585564e-02(3.574582e-02)	1.234456e-01(3.631822e-02)
	MOACO _R	1.633222e+01(1.027459e-02)	3.434431e-02(3.130326e-03)	6.621065e-02(6.538047e-03)
	MOEA/D	1.620770e+01(1.240921e-01)#	9.662943e-02(1.501319e-01)	1.668798e-01(3.452703e-02)#
	NSGA-III	1.631926e+01(1.253536e-02)	3.709021e-02(2.104592e-03)	6.030668e-02(4.979084e-03)
	SMS-EMOA	8.253488e+00(3.300261e+00)#	1.937586e-01(5.030458e-02)#	4.503081e-02(1.319550e-02)

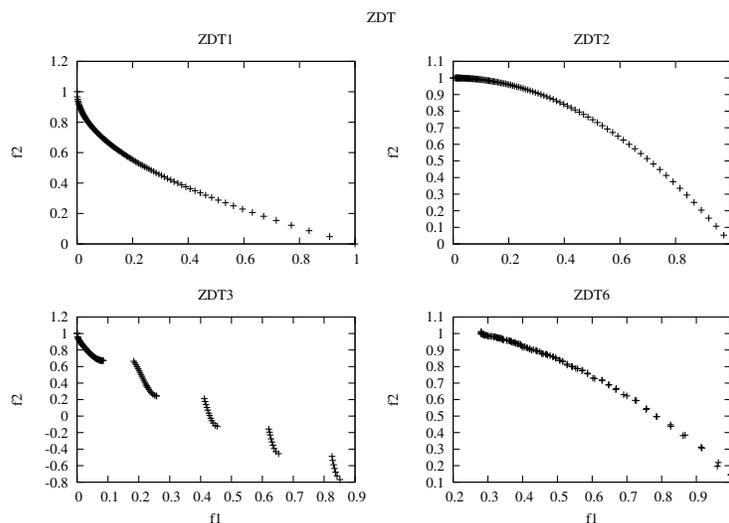


Figura B.1: Gráficas de las aproximaciones obtenidas por iMOACO_R para el conjunto de problemas ZDT, excepto ZDT4 donde no hay convergencia. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes.

Tabla B.3: Comparación de $iMOACO_{\mathbb{R}}$ con respecto a SMS-EMOA, MOEA/D, NSGA-III y $MOACO_{\mathbb{R}}$ en los problemas DTLZ con 4 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de $iMOACO_{\mathbb{R}}$ en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor.

Problema	Algoritmo	HV	IGD+
DTLZ1	$iMOACO_{\mathbb{R}}$	0.000000e+00(0.000000e+00)	N/A
	MOEA/D	9.943311e-01(9.545615e-05)	1.380371e-02(5.481799e-04)
	NSGA-III	9.943283e-01(1.272212e-04)	1.569669e-02(2.486756e-03)
	SMS-EMOA	3.569502e+00(1.531514e+01)	7.048639e-02(1.823935e-03)
DTLZ2	$iMOACO_{\mathbb{R}}$	1.555953e+01(9.926324e-04)	5.826377e-02(6.428884e-03)
	MOEA/D	1.556711e+01(1.801041e-04)	3.808984e-02(4.477105e-05)
	NSGA-III	1.556630e+01(4.883320e-04)	3.897468e-02(3.779742e-04)
	SMS-EMOA	1.018314e+01(6.136675e-01)#	2.624465e-01(8.374329e-03)#
DTLZ3	$iMOACO_{\mathbb{R}}$	0.000000e+00(0.000000e+00)	N/A
	MOEA/D	2.374591e+03(7.402603e+01)	1.614484e-01(3.284945e-01)
	NSGA-III	2.316027e+04(1.106149e+05)	4.688913e-01(5.593228e-01)
	SMS-EMOA	6.498654e+04(5.081899e+04)	1.685212e+00(8.766320e-01)
DTLZ4	$iMOACO_{\mathbb{R}}$	1.555148e+01(5.210018e-03)	5.664812e-02(4.705799e-03)
	MOEA/D	1.556735e+01(1.695190e-04)	3.203692e-02(1.993771e-05)
	NSGA-III	1.540943e+01(3.164946e-01)	6.155514e-02(5.810556e-02)
	SMS-EMOA	9.934107e+00(1.546094e+00)#	3.682866e-01(6.182626e-02)#
DTLZ5	$iMOACO_{\mathbb{R}}$	2.386289e+02(9.687689e-02)	3.667113e-02(3.406346e-03)
	MOEA/D	2.381580e+02(1.733960e-02)#	2.495048e-02(5.240391e-04)
	NSGA-III	2.385753e+02(1.274662e-01)	3.775222e-02(6.784102e-03)
	SMS-EMOA	1.881114e+02(1.194004e+00)#	6.077324e-02(1.756933e-03)#
DTLZ6	$iMOACO_{\mathbb{R}}$	1.445815e+04(3.699529e+01)	6.439012e-02(1.187921e-02)
	MOEA/D	1.447098e+04(1.149119e+01)	1.249933e-01(2.036896e-02)#
	NSGA-III	1.434327e+04(2.304387e+01)#	7.057010e-01(7.500204e-02)#
	SMS-EMOA	1.306067e+04(7.157015e+01)#	1.857343e-01(2.143434e-02)#
DTLZ7	$iMOACO_{\mathbb{R}}$	1.452694e+01(6.004490e-02)	5.518453e-01(1.912137e-01)
	MOEA/D	1.397065e+01(1.517147e-01)#	1.184711e-01(1.503761e-03)
	NSGA-III	1.462163e+01(3.547998e-02)	1.511980e-01(7.751993e-02)
	SMS-EMOA	3.520370e+00(3.613620e+00)#	4.205065e-01(1.795485e-01)

Tabla B.4: Comparación de $iMOACO_{\mathbb{R}}$ con respecto a SMS-EMOA, MOEA/D, NSGA-III y $MOACO_{\mathbb{R}}$ en los problemas DTLZ con 5 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de $iMOACO_{\mathbb{R}}$ en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor.

Problema	Algoritmo	HV	IGD+
DTLZ1	$iMOACO_{\mathbb{R}}$	0.000000e+00(0.000000e+00)	N/A
	MOEA/D	9.985205e-01(6.682053e-05)	1.770441e-02(1.492691e-03)
	NSGA-III	1.017165e+00(9.991802e-02)	2.092960e-02(8.441895e-03)
	SMS-EMOA	4.138286e+00(1.554831e+01)	8.781075e-02(2.831468e-02)
DTLZ2	$iMOACO_{\mathbb{R}}$	3.165018e+01(2.529270e-03)	9.001354e-02(5.425085e-03)
	MOEA/D	3.166768e+01(3.208840e-04)	6.668343e-02(8.984633e-05)
	NSGA-III	3.166519e+01(8.761134e-04)	6.850192e-02(7.848907e-04)
	SMS-EMOA	2.143628e+01(7.673809e-01)#	3.404108e-01(6.097470e-03)#
DTLZ3	$iMOACO_{\mathbb{R}}$	0.000000e+00(0.000000e+00)	N/A
	MOEA/D	1.718503e+04(4.197476e+03)	3.623186e-01(9.500668e-01)
	NSGA-III	2.990271e+04(5.922167e+04)	7.170602e-01(8.917696e-01)
	SMS-EMOA	5.327954e+05(3.461013e+05)	3.887638e+00(2.129632e+00)
DTLZ4	$iMOACO_{\mathbb{R}}$	3.163519e+01(3.624912e-03)	9.464995e-02(3.952929e-03)
	MOEA/D	3.166851e+01(1.701456e-04)	5.104591e-02(4.149652e-05)
	NSGA-III	3.162929e+01(1.453395e-01)	5.894254e-02(2.509120e-02)
	SMS-EMOA	2.323799e+01(2.179757e+00)#	3.921806e-01(3.884855e-02)#
DTLZ5	$iMOACO_{\mathbb{R}}$	9.372627e+02(9.321737e-01)	1.056244e-01(6.993555e-03)
	MOEA/D	9.465698e+02(4.026183e-01)	6.643330e-02(1.089015e-03)
	NSGA-III	9.447159e+02(6.332643e+00)	9.998197e-02(2.243920e-02)
	SMS-EMOA	7.232389e+02(5.931135e+00)#	9.516339e-02(2.970734e-03)
DTLZ6	$iMOACO_{\mathbb{R}}$	1.563606e+05(1.388763e+03)	2.026485e-01(1.348367e-01)
	MOEA/D	1.584583e+05(8.812537e+02)	2.446304e-01(7.294382e-03)
	NSGA-III	1.542282e+05(4.384451e+02)#	1.284266e+00(9.024491e-02)#
	SMS-EMOA	1.417357e+05(7.095308e+02)#	3.054495e-01(1.515756e-02)#
DTLZ7	$iMOACO_{\mathbb{R}}$	1.256712e+01(8.935918e-02)	1.152563e+00(3.334219e-01)
	MOEA/D	5.594105e+00(1.696376e+00)#	2.219557e-01(3.948292e-01)
	NSGA-III	1.283906e+01(3.913289e-02)	8.262998e-01(3.633242e-01)
	SMS-EMOA	2.089669e+00(3.188932e+00)#	7.051361e-01(3.533074e-01)

Tabla B.5: Comparación de $iMOACO_{\mathbb{R}}$ con respecto a SMS-EMOA, MOEA/D, NSGA-III y $MOACO_{\mathbb{R}}$ en los problemas DTLZ con 6 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de $iMOACO_{\mathbb{R}}$ en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor.

Problema	Algoritmo	HV	IGD+
DTLZ1	$iMOACO_{\mathbb{R}}$	0.000000e+00(0.000000e+00)	N/A
	MOEA/D	9.994822e-01(7.770987e-05)	1.580473e-02(1.704967e-03)
	NSGA-III	2.305540e+01(1.180937e+02)	2.795668e-02(2.999228e-02)
	SMS-EMOA	1.835879e+00(2.538413e+00)	1.179120e-01(3.818459e-02)
DTLZ2	$iMOACO_{\mathbb{R}}$	6.370521e+01(4.740828e-03)	9.853207e-02(9.182240e-03)
	MOEA/D	6.373919e+01(5.490093e-04)	4.494186e-02(1.014746e-04)
	NSGA-III	6.373291e+01(1.399292e-03)	4.972848e-02(7.784601e-04)
	SMS-EMOA	4.681115e+01(2.223731e+00)#	3.991549e-01(1.265877e-02)#
DTLZ3	$iMOACO_{\mathbb{R}}$	0.000000e+00(0.000000e+00)	N/A
	MOEA/D	1.106016e+05(7.967112e+03)	3.498793e-01(3.748689e-01)
	NSGA-III	5.500321e+05(9.953110e+05)	7.562300e-01(7.651005e-01)
	SMS-EMOA	2.999532e+06(2.353627e+06)	4.604515e+00(2.073649e+00)
DTLZ4	$iMOACO_{\mathbb{R}}$	6.369322e+01(5.775225e-03)	1.116681e-01(7.186360e-03)
	MOEA/D	6.374054e+01(3.873189e-04)	4.642420e-02(5.326031e-05)
	NSGA-III	6.373943e+01(1.149169e-03)	4.885719e-02(6.071485e-04)
	SMS-EMOA	4.901318e+01(4.450829e+00)#	4.675151e-01(3.851739e-02)#
DTLZ5	$iMOACO_{\mathbb{R}}$	3.598797e+03(2.549662e+01)	1.541606e-01(8.193899e-03)
	MOEA/D	3.761755e+03(5.525697e+00)	1.128131e-01(4.805875e-03)
	NSGA-III	3.747252e+03(4.625366e+01)	1.318704e-01(2.367743e-02)
	SMS-EMOA	2.808032e+03(2.031353e+01)#	1.503918e-01(1.952851e-03)
DTLZ6	$iMOACO_{\mathbb{R}}$	1.673873e+06(1.983622e+04)	1.041255e+00(3.212196e-01)
	MOEA/D	1.745800e+06(9.662609e+03)	1.649451e-01(7.454875e-03)
	NSGA-III	1.667950e+06(8.938931e+03)	2.305685e+00(1.765287e-01)#
	SMS-EMOA	1.548717e+06(1.216713e+04)#	2.492540e-01(1.911449e-02)
DTLZ7	$iMOACO_{\mathbb{R}}$	1.043309e+01(1.123806e-01)	1.501304e+00(3.607197e-01)
	MOEA/D	8.985705e-01(1.287970e+00)#	8.637355e-01(1.333606e-01)
	NSGA-III	1.077918e+01(7.096627e-02)	4.293837e-01(7.465534e-02)
	SMS-EMOA	2.994995e-01(5.840981e-01)#	7.743380e-01(1.564918e-01)

Tabla B.6: Comparación de $iMOACO_{\mathbb{R}}$ con respecto a SMS-EMOA, MOEA/D, NSGA-III y $MOACO_{\mathbb{R}}$ en los problemas DTLZ con 7 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de $iMOACO_{\mathbb{R}}$ en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor.

Problema	Algoritmo	HV	IGD+
DTLZ1	$iMOACO_{\mathbb{R}}$	0.000000e+00(0.000000e+00)	N/A
	MOEA/D	9.996951e-01(7.351713e-05)	4.057104e-02(1.569823e-03)
	NSGA-III	1.967483e+00(3.623084e+00)	5.422202e-02(4.656541e-02)
	SMS-EMOA	8.228500e-01(7.216038e-02)	1.100855e-01(4.155476e-03)
DTLZ2	$iMOACO_{\mathbb{R}}$	1.277012e+02(7.756509e-03)	1.386139e-01(9.467462e-03)
	MOEA/D	1.277477e+02(1.907858e-03)	8.300738e-02(2.521421e-05)
	NSGA-III	1.277489e+02(1.372213e-03)	8.636540e-02(1.142450e-03)
	SMS-EMOA	8.263293e+01(4.223524e+00)#	5.246910e-01(7.837699e-03)#
DTLZ3	$iMOACO_{\mathbb{R}}$	0.000000e+00(0.000000e+00)	N/A
	MOEA/D	7.426452e+05(5.279990e+04)	5.469882e-01(2.872446e-01)
	NSGA-III	7.172101e+06(3.423205e+07)	1.518148e+00(1.529985e+00)
	SMS-EMOA	1.907858e+06(3.530941e+06)	1.213168e+00(9.903056e-01)
DTLZ4	$iMOACO_{\mathbb{R}}$	1.277266e+02(7.245904e-03)	1.480956e-01(1.241785e-02)
	MOEA/D	1.277444e+02(2.351117e-03)	7.252125e-02(1.831991e-05)
	NSGA-III	1.277541e+02(4.924888e-04)	7.330909e-02(4.118729e-04)
	SMS-EMOA	9.249773e+01(3.292193e+00)#	5.534455e-01(1.991639e-02)#
DTLZ5	$iMOACO_{\mathbb{R}}$	1.444221e+04(1.115502e+02)	1.487053e-01(1.088418e-02)
	MOEA/D	1.492758e+04(1.705434e+01)	1.120602e-01(2.013699e-03)
	NSGA-III	1.486426e+04(9.418559e+01)	2.126830e-01(3.582548e-02)#
	SMS-EMOA	1.080888e+04(1.253629e+02)#	1.581263e-01(7.660657e-03)#
DTLZ6	$iMOACO_{\mathbb{R}}$	1.801024e+07(2.869872e+05)	1.102604e+00(3.119227e-01)
	MOEA/D	1.919747e+07(3.721837e+04)	2.096216e-01(2.663509e-02)
	NSGA-III	1.849564e+07(7.311372e+04)	1.819214e+00(1.212679e-01)#
	SMS-EMOA	1.672813e+07(1.029317e+05)#	2.890078e-01(2.385936e-02)
DTLZ7	$iMOACO_{\mathbb{R}}$	8.283139e+00(2.093956e-01)	1.899385e+00(2.490340e-01)
	MOEA/D	1.367398e-01(2.729438e-01)#	1.280057e+00(1.990017e-01)
	NSGA-III	8.446224e+00(3.724552e-01)	1.893459e+00(1.170848e+00)
	SMS-EMOA	9.710104e-03(2.028432e-02)#	1.180358e+00(5.177398e-02)

Tabla B.7: Comparación de $iMOACO_{\mathbb{R}}$ con respecto a SMS-EMOA, MOEA/D, NSGA-III y $MOACO_{\mathbb{R}}$ en los problemas DTLZ con 8 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de $iMOACO_{\mathbb{R}}$ en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor.

Problema	Algoritmo	HV	IGD+
DTLZ1	$iMOACO_{\mathbb{R}}$	0.000000e+00(0.000000e+00)	N/A
	MOEA/D	9.991297e-01(5.270251e-04)	1.189198e-02(1.516013e-03)
	NSGA-III	1.004654e+00(2.542664e-02)	1.892755e-02(7.183429e-03)
DTLZ2	$iMOACO_{\mathbb{R}}$	2.557234e+02(1.717644e-02)	1.220142e-01(5.635271e-03)
	MOEA/D	2.558177e+02(2.311926e-03)	3.013799e-02(3.382909e-04)
	NSGA-III	2.558107e+02(1.843094e-03)	3.788692e-02(1.239000e-03)
DTLZ3	$iMOACO_{\mathbb{R}}$	0.000000e+00(0.000000e+00)	N/A
	MOEA/D	5.253194e+06(3.896634e+05)	4.196636e-01(2.408509e-01)
	NSGA-III	2.480882e+07(1.015554e+08)	1.297698e+00(1.038208e+00)
DTLZ4	$iMOACO_{\mathbb{R}}$	2.557851e+02(5.167972e-03)	1.002000e-01(3.500030e-03)
	MOEA/D	2.558228e+02(1.092105e-03)	2.922782e-02(7.724539e-05)
	NSGA-III	2.558237e+02(5.445499e-04)	3.136773e-02(5.792336e-04)
DTLZ5	$iMOACO_{\mathbb{R}}$	5.798676e+04(4.121899e+02)	1.582678e-01(1.837894e-02)
	MOEA/D	5.950227e+04(6.073573e+01)	1.386604e-01(1.331292e-03)
	NSGA-III	5.921760e+04(4.930818e+02)	1.762621e-01(2.299422e-02)#
DTLZ6	$iMOACO_{\mathbb{R}}$	1.977356e+08(3.549822e+06)	9.415356e-01(2.630001e-01)
	MOEA/D	2.102613e+08(1.440565e+06)	2.489078e-01(2.931289e-02)
	NSGA-III	2.001208e+08(1.328996e+06)	1.614081e+00(1.349933e-01)#
DTLZ7	$iMOACO_{\mathbb{R}}$	6.280363e+00(1.705464e-01)	2.269312e+00(1.195733e-01)
	MOEA/D	6.330314e-02(1.216413e-01)#	1.450037e+00(2.185992e-01)
	NSGA-III	6.473320e+00(4.934296e-01)	2.475903e+00(1.054596e+00)

Tabla B.8: Comparación de $iMOACO_{\mathbb{R}}$ con respecto a SMS-EMOA, MOEA/D, NSGA-III y $MOACO_{\mathbb{R}}$ en los problemas DTLZ con 9 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de $iMOACO_{\mathbb{R}}$ en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor.

Problema	Algoritmo	HV	IGD+
DTLZ1	$iMOACO_{\mathbb{R}}$	0.000000e+00(0.000000e+00)	N/A
	MOEA/D	9.950370e-01(1.771090e-03)	1.988738e-02(1.824069e-03)
	NSGA-III	1.878841e+00(3.381959e+00)	3.148904e-02(1.294439e-02)
DTLZ2	$iMOACO_{\mathbb{R}}$	5.117154e+02(2.490649e-02)	1.392978e-01(5.921408e-03)
	MOEA/D	5.118614e+02(5.674322e-03)	2.659832e-02(5.637311e-04)
	NSGA-III	5.118467e+02(4.824518e-03)	4.058526e-02(2.509496e-03)
DTLZ3	$iMOACO_{\mathbb{R}}$	0.000000e+00(0.000000e+00)	N/A
	MOEA/D	3.638068e+07(2.603432e+06)	4.398037e-01(2.392109e-01)
	NSGA-III	8.573323e+07(1.886268e+08)	1.403462e+00(9.823564e-01)
DTLZ4	$iMOACO_{\mathbb{R}}$	5.118328e+02(4.129335e-03)	1.073289e-01(4.114008e-03)
	MOEA/D	5.118771e+02(5.411850e-04)	3.403791e-02(1.561452e-04)
	NSGA-III	5.118729e+02(1.379050e-03)	3.983688e-02(1.369303e-03)
DTLZ5	$iMOACO_{\mathbb{R}}$	2.326281e+05(1.528926e+03)	1.401497e-01(1.231367e-02)
	MOEA/D	2.373375e+05(2.213578e+02)	1.374697e-01(1.215129e-03)
	NSGA-III	2.360549e+05(2.377259e+03)	1.799687e-01(2.636119e-02)#
DTLZ6	$iMOACO_{\mathbb{R}}$	2.187571e+09(3.612619e+07)	8.693879e-01(2.090121e-01)
	MOEA/D	2.295236e+09(2.545105e+07)	3.301752e-01(9.714217e-02)
	NSGA-III	2.151751e+09(1.953411e+07)#	1.733311e+00(1.413267e-01)#
DTLZ7	$iMOACO_{\mathbb{R}}$	4.078025e+00(2.846239e-01)	2.231164e+00(1.238232e-01)
	MOEA/D	4.497601e-02(1.186930e-01)#	1.486490e+00(2.099893e-01)
	NSGA-III	4.134974e+00(6.218845e-01)	3.135058e+00(1.095893e+00)#

Tabla B.9: Comparación de $iMOACO_{\mathbb{R}}$ con respecto a SMS-EMOA, MOEA/D, NSGA-III y $MOACO_{\mathbb{R}}$ en los problemas DTLZ con 10 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de $iMOACO_{\mathbb{R}}$ en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor.

Problema	Algoritmo	HV	IGD+
DTLZ1	$iMOACO_{\mathbb{R}}$	0.000000e+00(0.000000e+00)	N/A
	MOEA/D	9.850897e-01(4.702988e-03)	2.711520e-02(2.466935e-03)
	NSGA-III	9.997797e-01(4.081014e-04)	5.412175e-02(4.205948e-02)
DTLZ2	$iMOACO_{\mathbb{R}}$	1.023614e+03(9.715725e-02)	1.601365e-01(7.690923e-03)
	MOEA/D	1.023857e+03(1.823335e-02)	3.045827e-02(9.151839e-04)
	NSGA-III	1.023849e+03(1.252941e-02)	4.832999e-02(3.458676e-03)
DTLZ3	$iMOACO_{\mathbb{R}}$	0.000000e+00(0.000000e+00)	N/A
	MOEA/D	2.563130e+08(1.884802e+07)	4.583286e-01(2.665935e-01)
	NSGA-III	2.599402e+09(6.664204e+09)	1.334163e+00(8.890106e-01)
DTLZ4	$iMOACO_{\mathbb{R}}$	1.023870e+03(4.029731e-03)	1.024742e-01(3.222647e-03)
	MOEA/D	1.023912e+03(2.478379e-03)	3.666230e-02(1.242001e-03)
	NSGA-III	1.023902e+03(3.014122e-03)	5.058520e-02(3.238759e-03)
DTLZ5	$iMOACO_{\mathbb{R}}$	9.357044e+05(6.615910e+03)	1.357425e-01(1.717228e-02)
	MOEA/D	9.477366e+05(6.757880e+02)	1.496687e-01(1.917818e-03)#
	NSGA-III	9.411728e+05(8.763266e+03)	1.814031e-01(2.168620e-02)#
DTLZ6	$iMOACO_{\mathbb{R}}$	2.416417e+10(3.315326e+08)	9.391129e-01(2.690567e-01)
	MOEA/D	2.529085e+10(2.087425e+08)	4.568281e-01(2.082470e-01)
	NSGA-III	2.322341e+10(2.063018e+08)#	2.374775e+00(1.287810e-01)#
DTLZ7	$iMOACO_{\mathbb{R}}$	2.152074e+00(6.544952e-01)	3.060151e+00(1.858266e-01)
	MOEA/D	2.947743e-03(7.802227e-03)#	1.587637e+00(1.843043e-01)
	NSGA-III	1.534016e+00(4.499430e-01)#	4.572206e+00(6.019360e-01)#

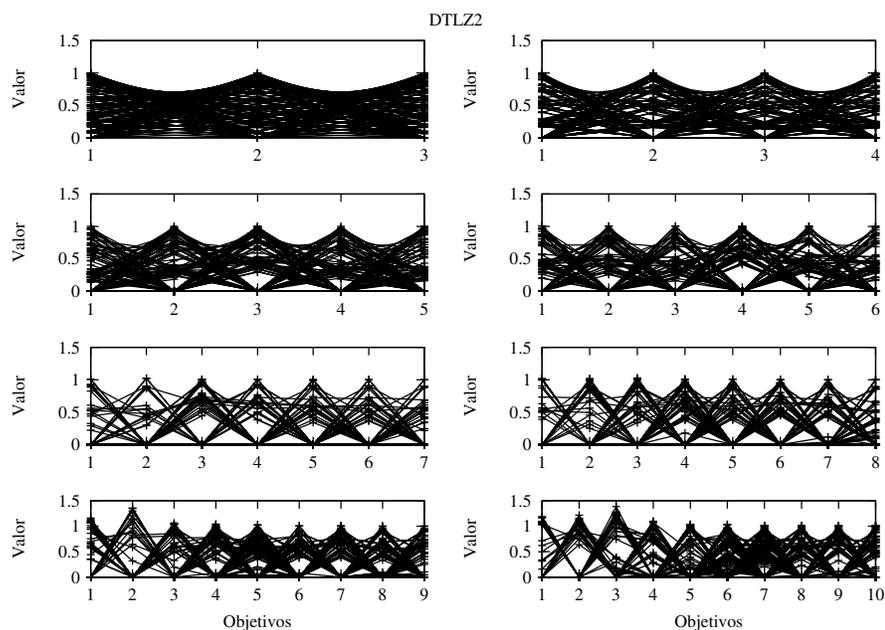


Figura B.2: Gráficas de las aproximaciones obtenidas por $iMOACO_{\mathbb{R}}$ para el problema DTLZ2 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes.

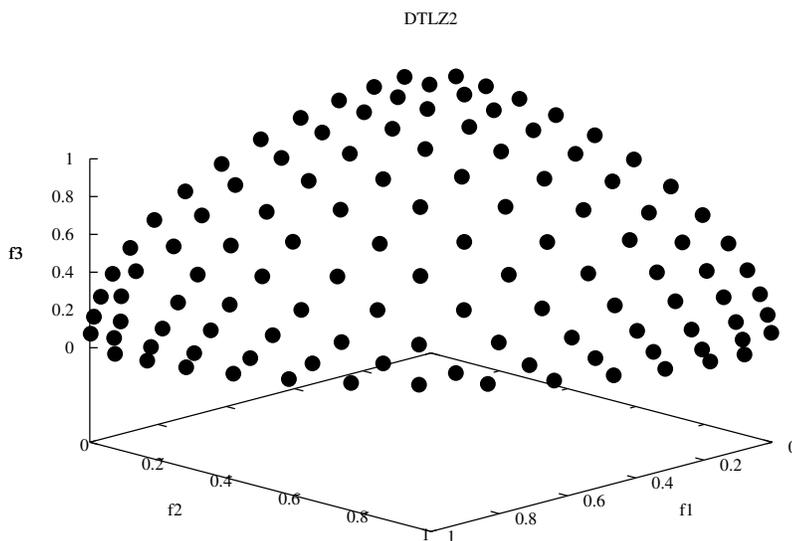


Figura B.3: Gráfica de la aproximación obtenida por $iMOACO_{\mathbb{R}}$ para DTLZ2 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen.

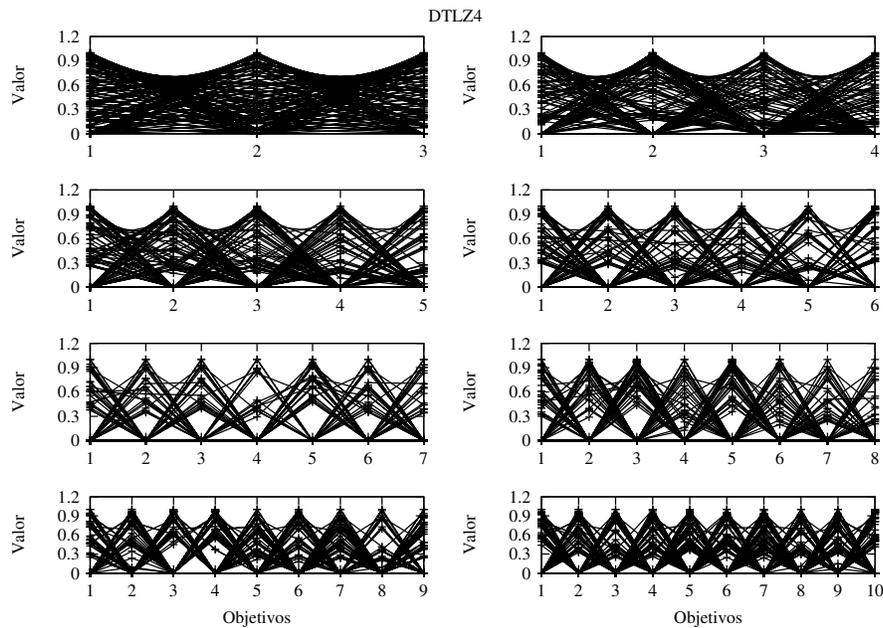


Figura B.4: Gráficas de las aproximaciones obtenidas por $iMOACO_{\mathbb{R}}$ para el problema DTLZ4 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes.

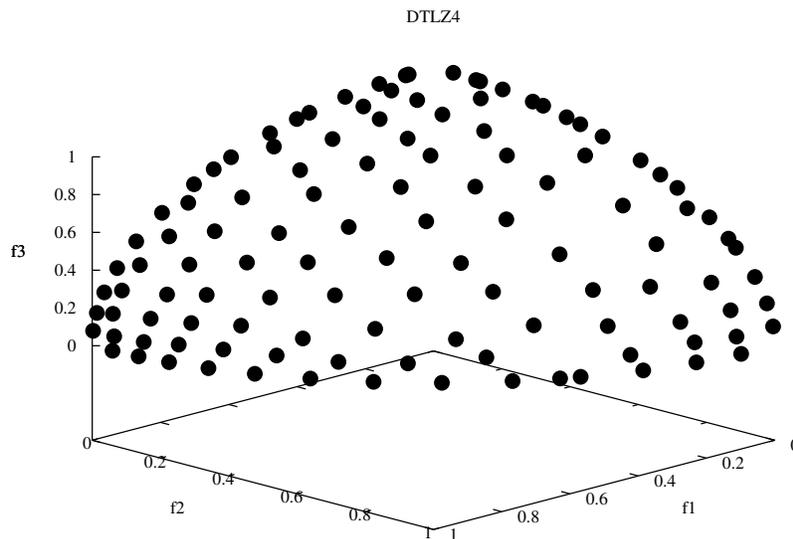


Figura B.5: Gráfica de la aproximación obtenida por $iMOACO_{\mathbb{R}}$ para DTLZ4 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen.

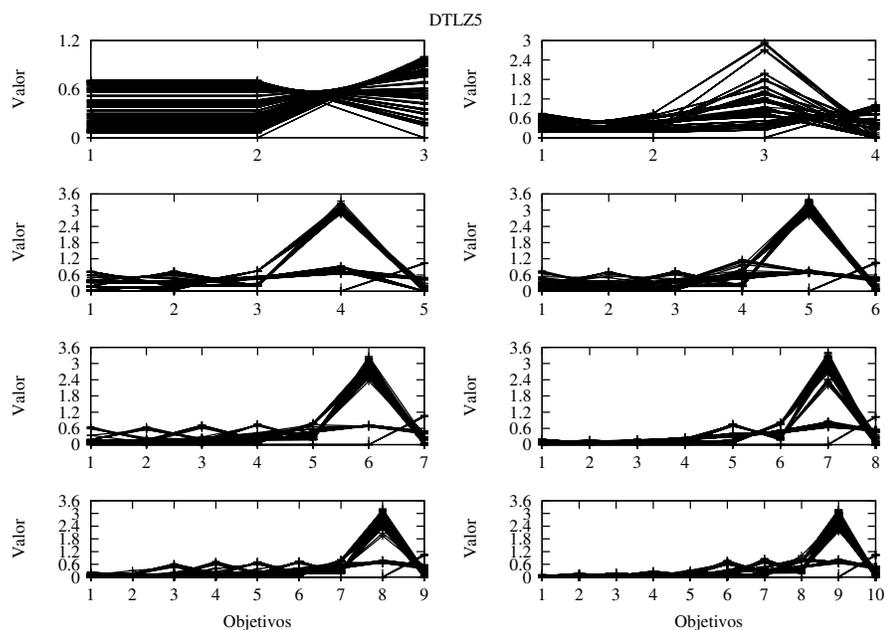


Figura B.6: Gráficas de las aproximaciones obtenidas por $iMOACO_{\mathbb{R}}$ para el problema DTLZ5 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes.

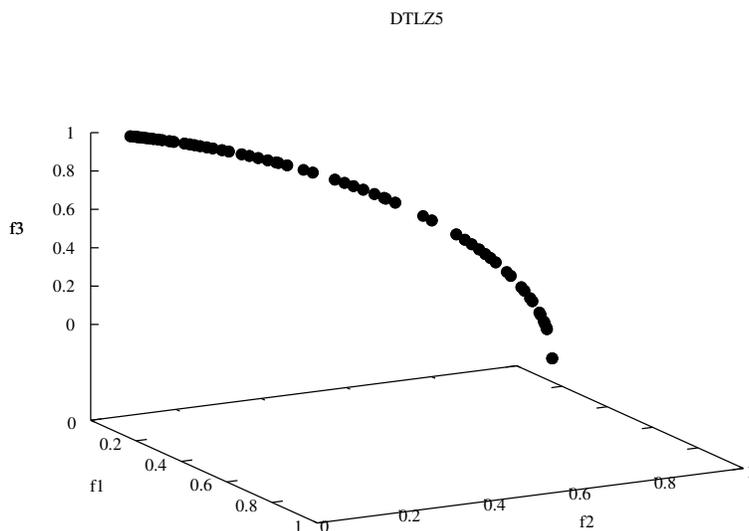


Figura B.7: Gráfica de la aproximación obtenida por $iMOACO_{\mathbb{R}}$ para DTLZ5 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen.

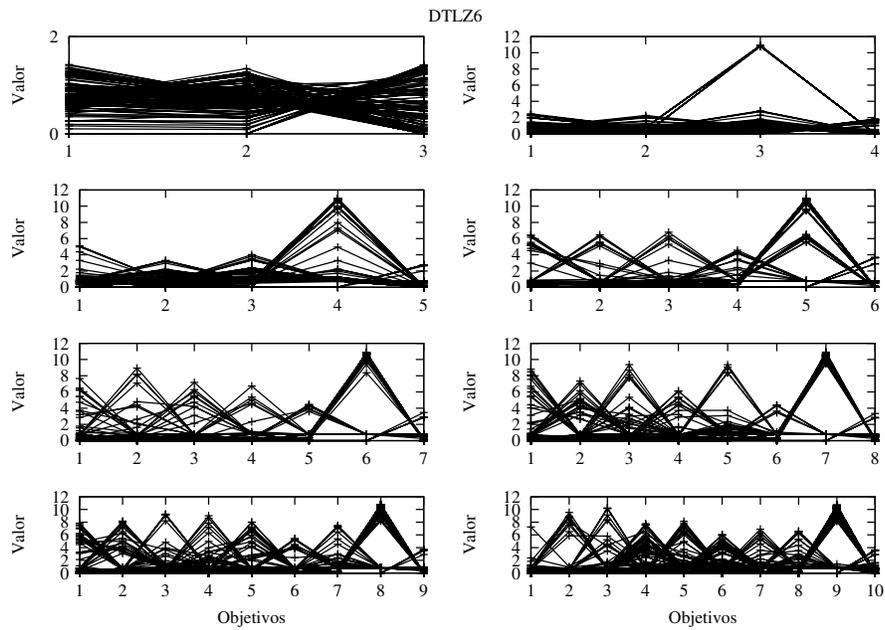


Figura B.8: Gráficas de las aproximaciones obtenidas por $iMOACO_{\mathbb{R}}$ para el problema DTLZ6 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes.

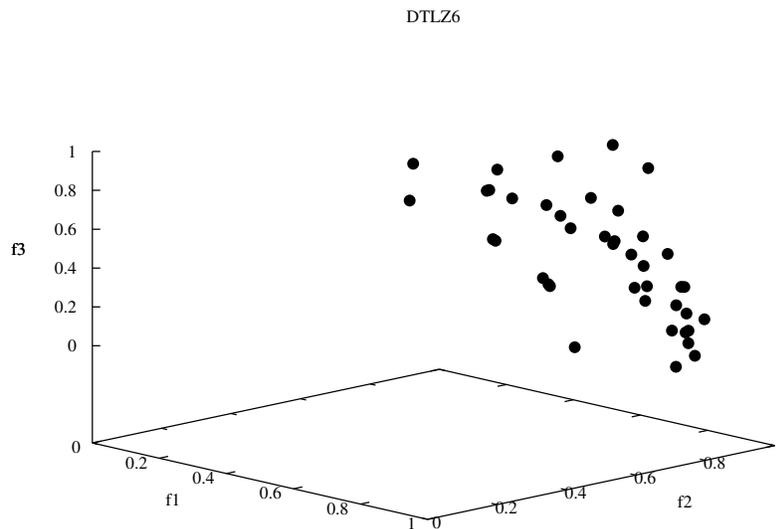


Figura B.9: Gráfica de la aproximación obtenida por $iMOACO_{\mathbb{R}}$ para DTLZ6 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen.

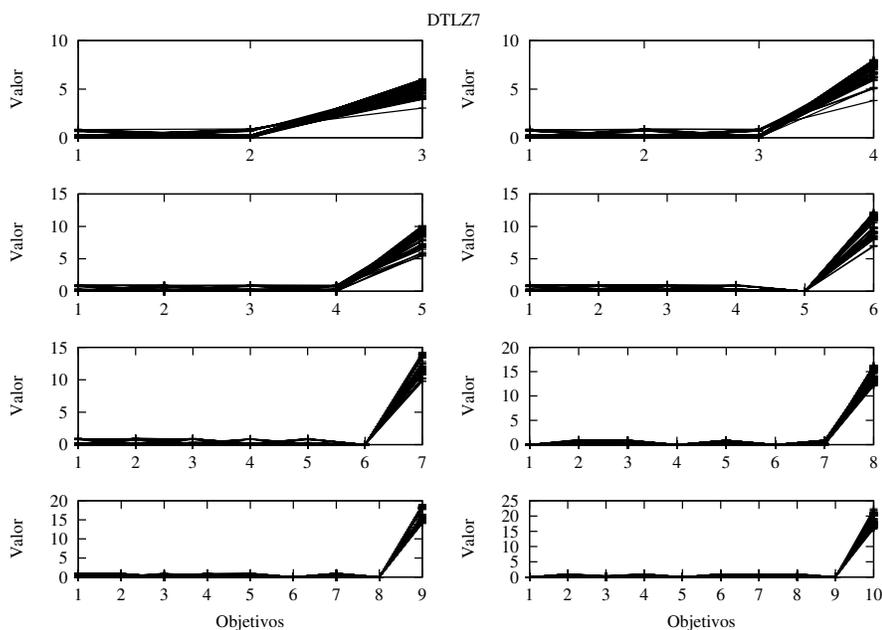


Figura B.10: Gráficas de las aproximaciones obtenidas por iMOACO_R para el problema DTLZ7 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes.

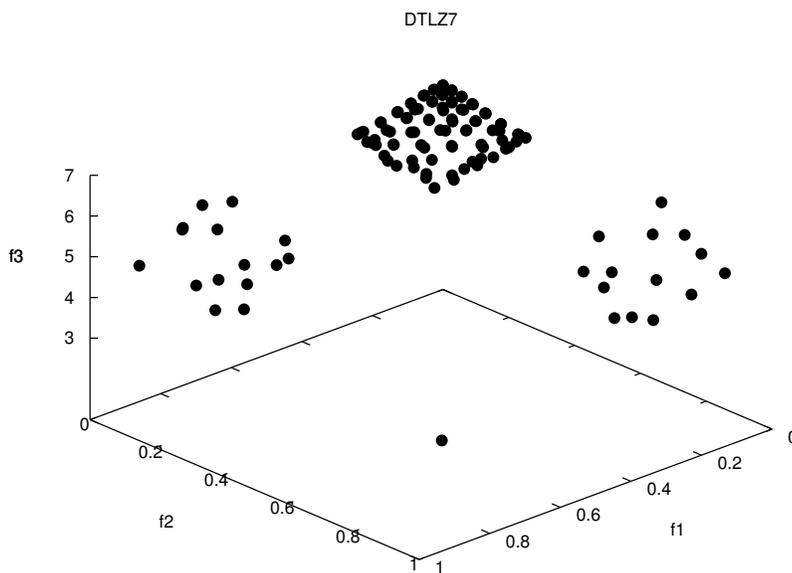


Figura B.11: Gráfica de la aproximación obtenida por iMOACO_R para DTLZ7 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen.

Tabla B.10: Comparación de $iMOACO_{\mathbb{R}}$ con respecto a SMS-EMOA, MOEA/D, NSGA-III y $MOACO_{\mathbb{R}}$ en los problemas WFG con 3 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de $iMOACO_{\mathbb{R}}$ en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor.

Problema	Algoritmo	HV	IGD+	S
WFG1	$iMOACO_{\mathbb{R}}$	4.425029e+01(5.984019e-01)	1.430117e+00(9.456159e-03)	1.055275e-01(2.969877e-02)
	$MOACO_{\mathbb{R}}$	4.482325e+01(2.679841e-01)	1.447081e+00(1.012481e-02)#	1.615551e-01(9.895479e-03)#
	MOEA/D	4.862674e+01(3.106147e+00)	1.294346e+00(4.620829e-02)	8.217338e-02(1.171159e-02)
	NSGA-III	4.353844e+01(2.491824e+00)#	1.363994e+00(2.860287e-02)	6.741428e-02(1.741846e-02)
	SMS-EMOA	4.561803e+01(1.162226e+00)	1.326351e+00(3.493015e-02)	4.239153e-02(9.209237e-03)
WFG2	$iMOACO_{\mathbb{R}}$	9.706166e+01(5.756624e-01)	1.114796e-01(1.584145e-02)	3.382511e-01(7.371948e-02)
	$MOACO_{\mathbb{R}}$	9.980343e+01(2.557216e-01)	7.510881e-02(1.379893e-02)	2.390982e-01(4.452840e-02)
	MOEA/D	8.220310e+01(6.485703e+00)#	2.507221e-01(6.124067e-02)#	8.975647e-02(1.658181e-02)
	NSGA-III	9.556901e+01(6.631542e+00)	4.831741e-02(7.710206e-03)	1.317904e-01(1.410287e-02)
	SMS-EMOA	7.472122e+01(4.189641e+00)#	6.071864e-01(1.551634e-01)#	4.702230e-02(5.693710e-02)
WFG3	$iMOACO_{\mathbb{R}}$	7.243938e+01(2.404101e-01)	1.485811e-01(9.103840e-03)	1.323829e-01(1.711702e-02)
	$MOACO_{\mathbb{R}}$	7.084552e+01(4.362761e-01)#	2.411323e-01(2.320650e-02)#	1.172149e-01(1.024720e-02)
	MOEA/D	6.879431e+01(2.038767e+00)#	1.888430e-01(4.826013e-02)#	1.940602e-01(2.327351e-02)#
	NSGA-III	7.405609e+01(4.437453e-01)	7.228996e-02(1.314983e-02)	9.726909e-02(9.199210e-03)
	SMS-EMOA	4.710888e+01(8.093235e-01)#	6.850529e-01(1.817772e-02)#	3.303524e-02(1.201534e-02)
WFG4	$iMOACO_{\mathbb{R}}$	7.068614e+01(3.010786e-01)	2.048992e-01(5.533108e-03)	3.039401e-01(1.250940e-02)
	$MOACO_{\mathbb{R}}$	6.887500e+01(4.965245e-01)#	2.450116e-01(1.176604e-02)#	1.971638e-01(1.690239e-02)
	MOEA/D	7.267887e+01(5.718139e-01)	1.977936e-01(1.199239e-02)	2.330489e-01(2.605824e-03)
	NSGA-III	7.501456e+01(2.272692e-01)	1.315811e-01(4.272907e-03)	1.884123e-01(4.541478e-02)
	SMS-EMOA	3.822859e+01(1.458471e+00)#	1.184868e+00(6.830780e-03)#	5.557515e-02(6.816786e-03)
WFG5	$iMOACO_{\mathbb{R}}$	6.834469e+01(8.124120e-01)	1.963753e-01(1.223894e-02)	3.104586e-01(1.350036e-02)
	$MOACO_{\mathbb{R}}$	6.346205e+01(1.672373e+00)#	2.926745e-01(3.711006e-02)#	1.784433e-01(1.516558e-02)
	MOEA/D	7.065721e+01(2.755646e-01)	2.057112e-01(6.869348e-03)#	2.295599e-01(1.726209e-03)
	NSGA-III	7.197491e+01(4.053423e-01)	1.725189e-01(2.703449e-03)	1.907428e-01(8.703439e-03)
	SMS-EMOA	3.537145e+01(1.463755e+00)#	8.767424e-01(3.600558e-02)#	6.326510e-02(3.617567e-02)
WFG6	$iMOACO_{\mathbb{R}}$	7.424075e+01(4.159130e-01)	1.104239e-01(5.556324e-03)	3.103782e-01(9.858763e-03)
	$MOACO_{\mathbb{R}}$	7.257026e+01(7.573651e-01)#	1.428572e-01(1.477144e-02)#	1.892799e-01(1.339569e-02)
	MOEA/D	7.065243e+01(7.719031e-01)#	2.002454e-01(2.172124e-02)#	2.355838e-01(5.291902e-03)
	NSGA-III	7.242177e+01(5.130163e-01)#	1.399133e-01(9.798090e-03)#	1.875025e-01(4.058775e-03)
	SMS-EMOA	3.732450e+01(2.274871e+00)#	1.080563e+00(1.993921e-02)#	6.161465e-02(1.452210e-02)
WFG7	$iMOACO_{\mathbb{R}}$	7.531982e+01(2.689471e-01)	1.062657e-01(3.269346e-03)	3.146193e-01(9.699109e-03)
	$MOACO_{\mathbb{R}}$	7.394943e+01(4.090000e-01)#	1.452924e-01(1.385566e-02)#	1.939828e-01(1.636724e-02)
	MOEA/D	7.254121e+01(1.567312e+00)#	1.596908e-01(1.044279e-02)#	2.360618e-01(3.696518e-03)
	NSGA-III	7.615301e+01(1.096385e-01)	9.364718e-02(2.070593e-03)	1.867679e-01(2.335161e-03)
	SMS-EMOA	3.670642e+01(1.643287e+00)#	5.564280e-01(2.119958e-02)#	5.220852e-02(1.183460e-02)
WFG8	$iMOACO_{\mathbb{R}}$	6.535407e+01(5.311808e-01)	2.363381e-01(9.816219e-03)	3.055681e-01(7.712514e-03)
	$MOACO_{\mathbb{R}}$	6.298557e+01(6.187356e-01)#	2.904208e-01(1.485750e-02)#	1.843525e-01(1.700297e-02)
	MOEA/D	6.717480e+01(6.432232e-01)	2.122931e-01(1.056637e-02)	2.654930e-01(5.887172e-03)
	NSGA-III	6.862097e+01(3.072845e-01)	1.729058e-01(6.910004e-03)	2.031002e-01(1.030285e-02)
	SMS-EMOA	3.428887e+01(1.102622e+00)#	9.814335e-01(1.599332e-02)#	6.640425e-02(1.162414e-02)
WFG9	$iMOACO_{\mathbb{R}}$	6.597174e+01(1.815331e-01)	1.841420e-01(1.479890e-03)	3.240877e-01(7.407864e-03)
	$MOACO_{\mathbb{R}}$	6.554430e+01(4.531830e-01)#	2.001286e-01(8.352817e-03)#	1.746757e-01(1.131601e-02)
	MOEA/D	6.533818e+01(1.699713e+00)	2.358542e-01(4.167114e-02)#	2.401901e-01(4.054049e-03)
	NSGA-III	7.002323e+01(1.791011e+00)	1.309404e-01(3.243199e-02)	1.849227e-01(7.993458e-03)
	SMS-EMOA	3.708994e+01(2.585178e+00)#	1.021225e+00(1.309794e-01)#	8.047164e-02(9.723917e-02)

Tabla B.11: Comparación de iMOACO_R con respecto a SMS-EMOA, MOEA/D, NSGA-III y MOACO_R en los problemas WFG con 4 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de iMOACO_R en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor.

Problema	Algoritmo	HV	IGD+
WFG1	iMOACO _R	3.779070e+02(5.435223e+00)	5.204317e-01(1.230109e-02)
	MOEA/D	3.974731e+02(2.264778e+01)	8.264190e-02(1.850842e-02)
	NSGA-III	3.396199e+02(1.457040e+01)#	2.466315e-01(2.203761e-02)
	SMS-EMOA	3.676477e+02(5.266490e+00)#	1.205113e-01(1.423763e-02)
WFG2	iMOACO _R	8.756858e+02(1.142973e+01)	1.896025e-01(9.807244e-03)
	MOEA/D	6.773334e+02(4.315408e+01)#	9.117077e-01(2.066586e-01)#
	NSGA-III	8.420102e+02(7.346656e+01)	4.175083e-01(3.439188e-01)
	SMS-EMOA	7.303554e+02(2.006765e+01)#	4.893210e-01(2.695535e-01)#
WFG3	iMOACO _R	5.073717e+02(1.249089e+01)	2.502167e-01(5.420219e-02)
	MOEA/D	5.477705e+02(1.874659e+01)	1.332975e-01(9.045409e-03)
	NSGA-III	6.035085e+02(6.061683e+00)	2.269870e-01(3.509637e-02)
	SMS-EMOA	3.359900e+02(8.920084e+00)#	3.801391e-01(9.405413e-03)#
WFG4	iMOACO _R	6.541449e+02(9.400810e+00)	2.769169e-01(1.689832e-02)
	MOEA/D	7.082134e+02(8.383225e+00)	1.696974e-01(1.410942e-02)
	NSGA-III	7.253274e+02(5.348179e+00)	1.365446e-01(9.651462e-03)
	SMS-EMOA	3.293540e+02(1.241157e+01)#	7.156384e-01(2.357809e-02)#
WFG5	iMOACO _R	5.281621e+02(1.349228e+01)	5.592144e-01(6.239704e-02)
	MOEA/D	6.926121e+02(9.294632e+00)	1.472366e-01(1.559483e-02)
	NSGA-III	7.081399e+02(2.736129e+00)	1.151855e-01(5.947146e-03)
	SMS-EMOA	3.156322e+02(1.004480e+01)#	6.329423e-01(1.773096e-02)#
WFG6	iMOACO _R	6.330659e+02(1.560068e+01)	3.236800e-01(3.044162e-02)
	MOEA/D	6.865464e+02(3.336583e+01)	2.164255e-01(4.510262e-02)
	NSGA-III	7.195842e+02(5.532542e+00)	1.224001e-01(1.042523e-02)
	SMS-EMOA	3.311214e+02(2.292382e+01)#	7.263399e-01(3.322639e-02)#
WFG7	iMOACO _R	6.537059e+02(9.081987e+00)	2.860414e-01(2.963342e-02)
	MOEA/D	7.047327e+02(1.709279e+01)	1.967921e-01(2.669043e-02)
	NSGA-III	7.409832e+02(6.258282e+00)	1.257572e-01(1.243187e-02)
	SMS-EMOA	3.219050e+02(8.069990e+00)#	6.549651e-01(1.954596e-02)#
WFG8	iMOACO _R	5.291345e+02(1.061961e+01)	4.496425e-01(3.336451e-02)
	MOEA/D	6.523351e+02(8.947127e+00)	1.511012e-01(2.287905e-02)
	NSGA-III	6.749823e+02(4.097776e+00)	1.461218e-01(1.203099e-02)
	SMS-EMOA	2.808962e+02(1.108147e+01)#	4.308027e-01(2.619311e-02)
WFG9	iMOACO _R	6.075473e+02(6.730029e+00)	3.028169e-01(2.168952e-02)
	MOEA/D	5.957882e+02(3.716614e+01)#	3.271483e-01(8.698328e-02)
	NSGA-III	6.710690e+02(7.609535e+00)	1.472087e-01(1.884419e-02)
	SMS-EMOA	3.251687e+02(1.992653e+01)#	6.843476e-01(2.279864e-02)#

Tabla B.12: Comparación de $iMOACO_{\mathbb{R}}$ con respecto a SMS-EMOA, MOEA/D, NSGA-III y $MOACO_{\mathbb{R}}$ en los problemas WFG con 5 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de $iMOACO_{\mathbb{R}}$ en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor.

Problema	Algoritmo	HV	IGD+
WFG1	$iMOACO_{\mathbb{R}}$	3.937020e+03(1.049030e+02)	6.179398e-01(2.081401e-02)
	MOEA/D	3.884123e+03(1.727569e+02)#	1.155149e-01(2.703645e-02)
	NSGA-III	3.246474e+03(7.198990e+01)#	3.577268e-01(2.199150e-02)
	SMS-EMOA	3.688324e+03(5.756271e+01)#	2.341036e-01(2.241015e-02)
WFG2	$iMOACO_{\mathbb{R}}$	9.736942e+03(9.487373e+01)	2.753754e-01(2.809498e-02)
	MOEA/D	7.225543e+03(5.552185e+02)#	1.217884e+00(4.161337e-01)#
	NSGA-III	9.245978e+03(8.013082e+02)	5.133961e-01(4.987213e-01)
	SMS-EMOA	8.281549e+03(1.282037e+02)#	7.239465e-01(4.892194e-01)#
WFG3	$iMOACO_{\mathbb{R}}$	5.400104e+03(2.486218e+02)	6.671848e-01(8.818594e-02)
	MOEA/D	5.236481e+03(1.518667e+02)#	2.547736e-01(1.713460e-02)
	NSGA-III	6.410663e+03(6.333082e+01)	6.035394e-01(5.807259e-02)
	SMS-EMOA	3.196907e+03(1.283129e+02)#	6.095891e-01(1.412658e-02)
WFG4	$iMOACO_{\mathbb{R}}$	7.617384e+03(1.162352e+02)	3.923978e-01(2.091630e-02)
	MOEA/D	7.466392e+03(2.892799e+02)	3.511958e-01(3.000826e-02)
	NSGA-III	8.063859e+03(8.255692e+01)	2.703007e-01(1.533752e-02)
	SMS-EMOA	3.544988e+03(7.739895e+01)#	9.282835e-01(2.034768e-02)#
WFG5	$iMOACO_{\mathbb{R}}$	4.835114e+03(1.938147e+02)	1.197580e+00(9.484894e-02)
	MOEA/D	7.629797e+03(1.728179e+02)	2.585594e-01(8.855850e-03)
	NSGA-III	8.011001e+03(6.378816e+01)	2.193627e-01(1.334362e-02)
	SMS-EMOA	3.485080e+03(1.106861e+02)#	7.810527e-01(2.091194e-02)
WFG6	$iMOACO_{\mathbb{R}}$	6.768850e+03(3.912915e+02)	5.469506e-01(9.212335e-02)
	MOEA/D	6.431094e+03(9.169130e+02)	3.980014e-01(8.110685e-02)
	NSGA-III	8.230034e+03(6.976933e+01)	2.223761e-01(1.321000e-02)
	SMS-EMOA	3.568933e+03(2.393517e+02)#	8.117315e-01(4.400236e-02)#
WFG7	$iMOACO_{\mathbb{R}}$	7.261774e+03(2.611695e+02)	4.764515e-01(5.880451e-02)
	MOEA/D	6.982161e+03(4.230249e+02)#	3.607015e-01(3.271600e-02)
	NSGA-III	8.313109e+03(1.061556e+02)	2.436847e-01(2.408693e-02)
	SMS-EMOA	3.463331e+03(7.242823e+01)#	8.468314e-01(1.661190e-02)#
WFG8	$iMOACO_{\mathbb{R}}$	5.109791e+03(2.478179e+02)	7.648970e-01(5.604834e-02)
	MOEA/D	6.203692e+03(5.497402e+02)	4.274071e-01(1.142042e-01)
	NSGA-III	7.603183e+03(7.756182e+01)	2.406510e-01(1.567165e-02)
	SMS-EMOA	2.934883e+03(1.141020e+02)#	7.894058e-01(2.615700e-02)#
WFG9	$iMOACO_{\mathbb{R}}$	5.768426e+03(3.253641e+02)	8.974369e-01(1.70491e-01)
	MOEA/D	6.362462e+03(9.840560e+02)	4.452596e-01(2.242429e-01)
	NSGA-III	7.476119e+03(2.474276e+02)	2.607509e-01(5.326271e-02)
	SMS-EMOA	3.629624e+03(2.498838e+02)#	7.080046e-01(3.199983e-02)

Tabla B.13: Comparación de iMOACO_R con respecto a SMS-EMOA, MOEA/D, NSGA-III y MOACO_R en los problemas WFG con 6 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de iMOACO_R en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor.

Problema	Algoritmo	HV	IGD+
WFG1	iMOACO _R	4.988048e+04(1.093227e+03)	9.039183e-01(1.044152e-01)
	MOEA/D	4.685337e+04(1.567905e+03)#	9.533830e-02(1.834721e-02)
	NSGA-III	3.941927e+04(8.366054e+02)#	3.324339e-01(1.565156e-02)
	SMS-EMOA	4.414778e+04(8.368155e+02)#	2.699755e-01(2.836786e-02)
WFG2	iMOACO _R	1.202131e+05(2.022269e+03)	5.274913e-01(8.028000e-02)
	MOEA/D	8.661256e+04(6.245738e+03)#	1.200969e+00(2.645566e-01)#
	NSGA-III	1.160896e+05(9.844374e+03)	5.193222e-01(3.930801e-01)
	SMS-EMOA	1.083304e+05(3.888621e+03)#	5.924379e-01(3.682609e-01)
WFG3	iMOACO _R	6.275917e+04(2.558492e+03)	7.519100e-01(4.884092e-02)
	MOEA/D	6.046429e+04(2.557205e+03)#	3.859694e-01(2.280231e-02)
	NSGA-III	7.422489e+04(1.775354e+03)	7.045587e-01(1.546328e-01)
	SMS-EMOA	3.829615e+04(1.115623e+03)#	7.143502e-01(1.533763e-02)
WFG4	iMOACO _R	9.876248e+04(1.426464e+03)	5.158727e-01(4.978884e-02)
	MOEA/D	8.159895e+04(7.073899e+03)#	4.924053e-01(2.031927e-02)
	NSGA-III	1.038887e+05(9.897055e+02)	3.582118e-01(2.451896e-02)
	SMS-EMOA	4.541043e+04(1.342627e+03)#	9.257976e-01(4.059377e-02)#
WFG5	iMOACO _R	5.712345e+04(2.623736e+03)	1.902692e+00(2.066372e-01)
	MOEA/D	9.219766e+04(3.501250e+03)	3.883180e-01(1.315390e-02)
	NSGA-III	1.041978e+05(7.994552e+02)	2.621256e-01(2.335030e-02)
	SMS-EMOA	4.409512e+04(1.878372e+03)#	8.215202e-01(3.947944e-02)
WFG6	iMOACO _R	7.528402e+04(5.028049e+03)	8.660371e-01(1.262445e-01)
	MOEA/D	6.375531e+04(7.368495e+03)#	5.466698e-01(3.634911e-02)
	NSGA-III	1.091265e+05(8.997488e+02)	2.382446e-01(1.756182e-02)
	SMS-EMOA	4.529093e+04(2.226668e+03)#	8.213507e-01(3.529056e-02)
WFG7	iMOACO _R	8.617185e+04(3.412758e+03)	6.707940e-01(7.984867e-02)
	MOEA/D	6.831210e+04(6.788793e+03)#	4.366357e-01(2.542875e-02)
	NSGA-III	1.086707e+05(1.312944e+03)	3.011161e-01(2.243233e-02)
	SMS-EMOA	4.493463e+04(1.364937e+03)#	7.251148e-01(1.971580e-02)#
WFG8	iMOACO _R	5.993587e+04(4.009690e+03)	1.053207e+00(1.383710e-01)
	MOEA/D	5.198895e+04(1.018872e+04)#	8.278785e-01(1.597002e-01)
	NSGA-III	9.877402e+04(1.090021e+03)	2.567906e-01(2.003035e-02)
	SMS-EMOA	3.669157e+04(1.514762e+03)#	9.568788e-01(4.513450e-02)
WFG9	iMOACO _R	6.405953e+04(6.562732e+03)	1.636457e+00(1.923407e-01)
	MOEA/D	6.940689e+04(2.025355e+04)	7.979695e-01(3.670257e-01)
	NSGA-III	9.213486e+04(4.181895e+03)	3.941825e-01(8.708418e-02)
	SMS-EMOA	4.912636e+04(6.259211e+03)#	8.205162e-01(6.087909e-02)

Tabla B.14: Comparación de $iMOACO_{\mathbb{R}}$ con respecto a SMS-EMOA, MOEA/D, NSGA-III y $MOACO_{\mathbb{R}}$ en los problemas WFG con 7 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de $iMOACO_{\mathbb{R}}$ en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor.

Problema	Algoritmo	HV	IGD+
WFG1	$iMOACO_{\mathbb{R}}$	6.762082e+05(3.201365e+04)	1.028239e+00(1.473456e-01)
	MOEA/D	6.618675e+05(1.448923e+04)#	1.115985e-01(2.944947e-02)
	NSGA-III	5.478707e+05(1.352350e+04)#	3.822505e-01(2.072487e-02)
	SMS-EMOA	5.913484e+05(1.588312e+04)#	4.146121e-01(4.105536e-02)
WFG2	$iMOACO_{\mathbb{R}}$	1.701444e+06(2.296490e+04)	8.755361e-01(1.499445e-01)
	MOEA/D	1.227180e+06(6.399181e+04)#	1.680729e+00(4.983218e-01)#
	NSGA-III	1.672969e+06(1.575208e+05)	7.917200e-01(5.896024e-01)
	SMS-EMOA	1.527222e+06(6.119291e+04)#	9.216179e-01(5.588253e-01)
WFG3	$iMOACO_{\mathbb{R}}$	7.904484e+05(2.199719e+04)	9.882048e-01(8.973024e-02)
	MOEA/D	7.264922e+05(5.165113e+04)#	6.327384e-01(2.945734e-02)
	NSGA-III	5.312858e+05(2.120121e+05)#	7.781914e+00(2.409585e+00)#
	SMS-EMOA	4.979576e+05(2.701955e+04)#	9.318176e-01(5.209673e-02)
WFG4	$iMOACO_{\mathbb{R}}$	1.244395e+06(4.907534e+04)	1.065930e+00(2.533077e-01)
	MOEA/D	1.019393e+06(1.131978e+05)#	7.705341e-01(3.280681e-02)
	NSGA-III	1.583968e+06(2.412821e+04)	3.730912e-01(6.329186e-02)
	SMS-EMOA	6.203148e+05(1.688428e+04)#	1.224622e+00(2.313288e-02)#
WFG5	$iMOACO_{\mathbb{R}}$	6.887508e+05(4.105392e+04)	2.450597e+00(2.191749e-01)
	MOEA/D	1.159399e+06(1.543842e+05)	5.237272e-01(6.163893e-02)
	NSGA-III	1.588530e+06(2.191853e+04)	2.674830e-01(2.846235e-02)
	SMS-EMOA	6.047207e+05(2.297134e+04)#	1.090338e+00(3.707254e-02)
WFG6	$iMOACO_{\mathbb{R}}$	8.324711e+05(5.551179e+04)	1.343287e+00(1.042600e-01)
	MOEA/D	8.176655e+05(9.898115e+03)	6.406209e-01(5.949187e-03)
	NSGA-III	1.693310e+06(1.419587e+04)	2.173142e-01(1.622453e-02)
	SMS-EMOA	6.188628e+05(1.439637e+04)#	1.013691e+00(1.940391e-02)
WFG7	$iMOACO_{\mathbb{R}}$	1.094708e+06(5.602899e+04)	1.029619e+00(4.873109e-02)
	MOEA/D	8.603106e+05(1.645899e+04)#	7.712396e-01(7.232041e-03)
	NSGA-III	1.664359e+06(2.987976e+04)	2.739815e-01(3.385329e-02)
	SMS-EMOA	6.225210e+05(6.574758e+03)#	1.223627e+00(1.849134e-02)#
WFG8	$iMOACO_{\mathbb{R}}$	7.712374e+05(6.224059e+04)	1.392503e+00(1.029115e-01)
	MOEA/D	4.876062e+05(1.312924e+05)#	1.243237e+00(1.979969e-01)
	NSGA-III	1.476134e+06(2.868439e+04)	3.524058e-01(2.377230e-01)
	SMS-EMOA	4.547833e+05(4.406183e+04)#	1.294171e+00(8.067600e-02)
WFG9	$iMOACO_{\mathbb{R}}$	7.309045e+05(1.035562e+05)	2.301408e+00(3.109058e-01)
	MOEA/D	8.943812e+05(2.363024e+05)	9.136098e-01(3.396456e-01)
	NSGA-III	1.352921e+06(5.646612e+04)	4.106712e-01(1.066126e-01)
	SMS-EMOA	6.987414e+05(1.173542e+05)#	9.305680e-01(8.084450e-02)

Tabla B.15: Comparación de iMOACO_R con respecto a SMS-EMOA, MOEA/D, NSGA-III y MOACO_R en los problemas WFG con 8 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de iMOACO_R en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor.

Problema	Algoritmo	HV	IGD+
WFG1	iMOACO _R	1.099418e+07(1.868922e+05)	1.034265e+00(1.415052e-01)
	MOEA/D	1.072172e+07(2.752426e+05)#	9.639874e-02(3.207961e-02)
	NSGA-III	9.073276e+06(2.431665e+05)#	3.726214e-01(1.929939e-02)
WFG2	iMOACO _R	2.674582e+07(5.200775e+05)	8.881944e-01(1.022555e-01)
	MOEA/D	2.069881e+07(1.597209e+06)#	1.517084e+00(5.313491e-01)#
	NSGA-III	2.791803e+07(2.430618e+06)	7.205517e-01(4.910127e-01)
WFG3	iMOACO _R	1.275653e+07(3.090688e+05)	1.534220e+00(1.073513e-01)
	MOEA/D	1.010919e+07(1.078870e+06)#	1.010205e+00(2.474098e-02)
	NSGA-III	1.342893e+07(1.956928e+06)	4.164676e+00(1.298595e+00)#
WFG4	iMOACO _R	1.947120e+07(9.466695e+05)	1.628241e+00(2.616811e-01)
	MOEA/D	1.642259e+07(1.698212e+06)#	1.138751e+00(3.005174e-02)
	NSGA-III	2.548122e+07(4.843177e+05)	4.581312e-01(7.354374e-02)
WFG5	iMOACO _R	1.111816e+07(6.557157e+05)	2.865232e+00(1.680154e-01)
	MOEA/D	1.809912e+07(1.719420e+06)	8.423352e-01(4.215075e-02)
	NSGA-III	2.595680e+07(3.368089e+05)	3.507118e-01(3.792481e-02)
WFG6	iMOACO _R	1.319034e+07(9.146966e+05)	1.862088e+00(7.045514e-02)
	MOEA/D	1.384867e+07(2.237035e+05)	1.287216e+00(1.068971e-02)
	NSGA-III	2.760398e+07(2.959042e+05)	2.750048e-01(1.906647e-02)
WFG7	iMOACO _R	1.771717e+07(9.100887e+05)	1.327876e+00(4.129580e-02)
	MOEA/D	1.456138e+07(8.497697e+05)#	1.135239e+00(2.847322e-02)
	NSGA-III	2.739862e+07(5.450471e+05)	3.061215e-01(4.607725e-02)
WFG8	iMOACO _R	1.281206e+07(1.196707e+06)	1.857608e+00(1.992956e-01)
	MOEA/D	7.853218e+06(2.284755e+06)#	1.617155e+00(1.878636e-01)
	NSGA-III	2.437662e+07(3.901969e+05)	2.999606e-01(9.118173e-02)
WFG9	iMOACO _R	1.161878e+07(1.467969e+06)	2.559735e+00(1.959386e-01)
	MOEA/D	1.113221e+07(4.449867e+06)	1.456518e+00(3.692858e-01)
	NSGA-III	2.156169e+07(9.082809e+05)	4.262006e-01(1.105906e-01)

Tabla B.16: Comparación de iMOACO_R con respecto a SMS-EMOA, MOEA/D, NSGA-III y MOACO_R en los problemas WFG con 9 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de iMOACO_R en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor.

Problema	Algoritmo	HV	IGD+
WFG1	iMOACO _R	1.980764e+08(2.976305e+06)	9.563075e-01(6.633308e-02)
	MOEA/D	1.915988e+08(5.744281e+06)#	8.914959e-02(2.229973e-02)
	NSGA-III	1.641070e+08(4.130947e+06)#	3.737487e-01(2.545191e-02)
WFG2	iMOACO _R	4.718763e+08(6.411157e+06)	8.687166e-01(8.312564e-02)
	MOEA/D	3.972435e+08(3.071284e+07)#	2.109405e+00(6.847355e-01)#
	NSGA-III	5.502786e+08(3.409011e+07)	5.460481e-01(7.413055e-01)
WFG3	iMOACO _R	2.380295e+08(1.241648e+07)	1.855672e+00(2.180650e-01)
	MOEA/D	1.501079e+08(2.147825e+07)#	1.198755e+00(4.481365e-02)
	NSGA-III	2.791555e+08(2.842832e+07)	3.167620e+00(1.076804e+00)#
WFG4	iMOACO _R	3.525084e+08(1.312829e+07)	2.236210e+00(2.678635e-01)
	MOEA/D	3.025931e+08(2.723044e+07)#	1.156438e+00(3.037090e-02)
	NSGA-III	4.582219e+08(8.133565e+06)	6.770243e-01(6.839852e-02)
WFG5	iMOACO _R	2.091048e+08(1.052864e+07)	3.486482e+00(2.258387e-01)
	MOEA/D	3.077389e+08(3.618177e+07)	8.515227e-01(3.675762e-02)
	NSGA-III	4.653430e+08(8.007185e+06)	5.099968e-01(4.268390e-02)
WFG6	iMOACO _R	2.324528e+08(1.415061e+07)	2.056769e+00(1.396641e-01)
	MOEA/D	2.535107e+08(1.092558e+07)	1.201651e+00(2.542936e-02)
	NSGA-III	4.839935e+08(1.088999e+07)	4.112335e-01(4.461819e-02)
WFG7	iMOACO _R	3.364178e+08(1.199674e+07)	1.460149e+00(6.861584e-02)
	MOEA/D	2.625455e+08(1.095419e+07)#	1.191356e+00(1.753072e-02)
	NSGA-III	4.929687e+08(8.555610e+06)	4.010810e-01(4.311496e-02)
WFG8	iMOACO _R	2.492487e+08(1.813810e+07)	1.908676e+00(8.479922e-02)
	MOEA/D	1.333578e+08(3.706148e+07)#	1.645003e+00(1.658188e-01)
	NSGA-III	4.376592e+08(8.744481e+06)	4.067969e-01(9.800779e-02)
WFG9	iMOACO _R	2.093195e+08(2.540844e+07)	3.041535e+00(2.695595e-01)
	MOEA/D	1.827497e+08(8.375147e+07)	1.471417e+00(3.906610e-01)
	NSGA-III	3.764141e+08(1.318983e+07)	5.459089e-01(5.712510e-02)

Tabla B.17: Comparación de iMOACO_R con respecto a SMS-EMOA, MOEA/D, NSGA-III y MOACO_R en los problemas WFG con 10 objetivos. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de iMOACO_R en la prueba Wilcoxon con un nivel del 5%. Los mejores dos valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor.

Problema	Algoritmo	HV	IGD+
WFG1	iMOACO _R	3.961333e+09(2.039211e+07)	1.038277e+00(1.221703e-01)
	MOEA/D	3.802791e+09(5.153363e+07)#	6.107899e-02(1.435852e-02)
	NSGA-III	3.273061e+09(5.361818e+07)#	3.946059e-01(1.701356e-02)
WFG2	iMOACO _R	9.437415e+09(1.712836e+08)	9.497545e-01(7.075744e-02)
	MOEA/D	8.086769e+09(3.775121e+08)#	2.368147e+00(3.475890e-01)#
	NSGA-III	1.117947e+10(8.590828e+08)	8.249208e-01(9.370814e-01)
WFG3	iMOACO _R	4.742821e+09(7.999123e+07)	2.039118e+00(9.069703e-02)
	MOEA/D	2.394173e+09(1.504879e+08)#	1.311043e+00(2.552686e-02)
	NSGA-III	5.662479e+09(3.371602e+08)	3.069705e+00(7.192726e-01)#
WFG4	iMOACO _R	7.170558e+09(2.850698e+08)	2.815883e+00(2.478266e-01)
	MOEA/D	6.385061e+09(7.696642e+08)#	1.240225e+00(2.006889e-02)
	NSGA-III	8.950776e+09(2.529560e+08)	9.860570e-01(1.236551e-01)
WFG5	iMOACO _R	4.317068e+09(1.650082e+08)	3.836405e+00(2.880983e-01)
	MOEA/D	6.192197e+09(7.343777e+08)	8.299703e-01(1.921598e-02)
	NSGA-III	8.883282e+09(1.562287e+08)	6.795388e-01(4.928624e-02)
WFG6	iMOACO _R	4.790208e+09(2.524902e+08)	2.174708e+00(1.602587e-01)
	MOEA/D	5.170353e+09(1.858406e+08)	1.153931e+00(1.470494e-02)
	NSGA-III	8.885712e+09(2.249019e+08)	5.371587e-01(5.946452e-02)
WFG7	iMOACO _R	6.963838e+09(2.330565e+08)	1.555666e+00(1.116714e-01)
	MOEA/D	5.243514e+09(2.561860e+08)#	1.120330e+00(1.898160e-02)
	NSGA-III	9.386685e+09(1.513259e+08)	5.434857e-01(5.101777e-02)
WFG8	iMOACO _R	5.339008e+09(3.843827e+08)	2.049588e+00(1.115740e-01)
	MOEA/D	2.549207e+09(8.836877e+08)#	1.670862e+00(1.425014e-01)
	NSGA-III	8.394377e+09(1.694206e+08)	5.230700e-01(8.613283e-02)
WFG9	iMOACO _R	4.188941e+09(3.585450e+08)	3.543821e+00(2.839773e-01)
	MOEA/D	3.122641e+09(1.626959e+09)#	1.517764e+00(4.994526e-01)
	NSGA-III	7.329193e+09(3.111879e+08)	6.791161e-01(6.389943e-02)

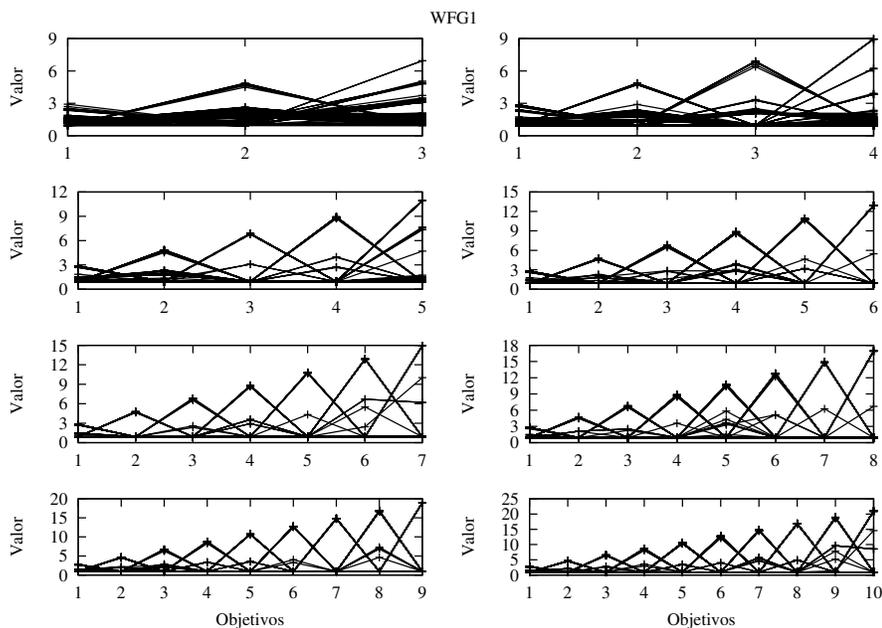


Figura B.12: Gráficas de las aproximaciones obtenidas por iMOACO_ℝ para el problema WFG1 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes.

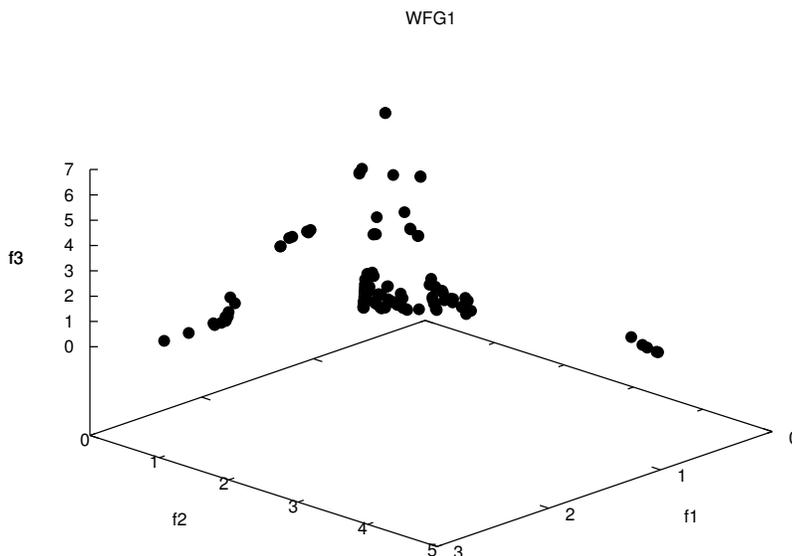


Figura B.13: Gráfica de la aproximación obtenida por iMOACO_ℝ para WFG1 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen.

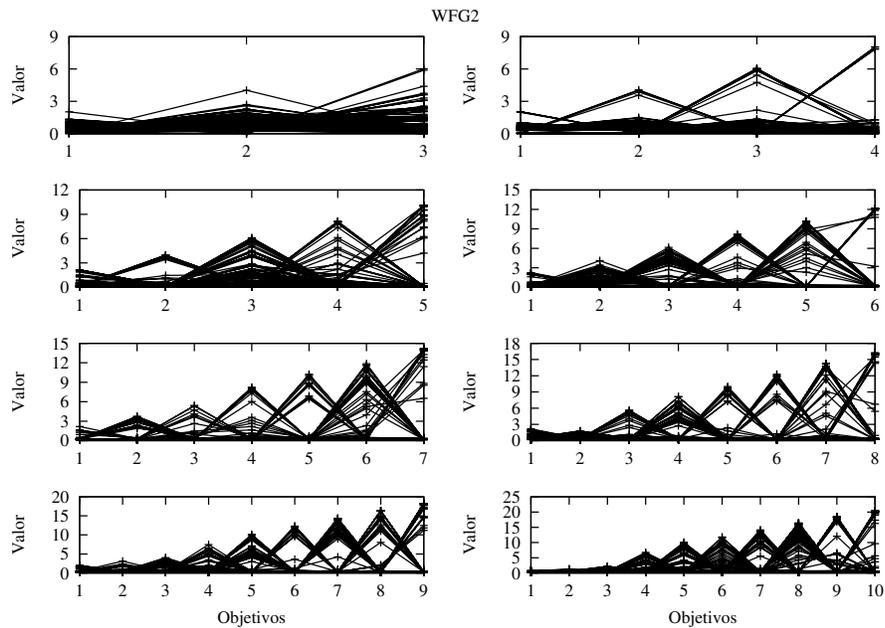


Figura B.14: Gráficas de las aproximaciones obtenidas por $iMOACO_{\mathbb{R}}$ para el problema WFG2 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes.

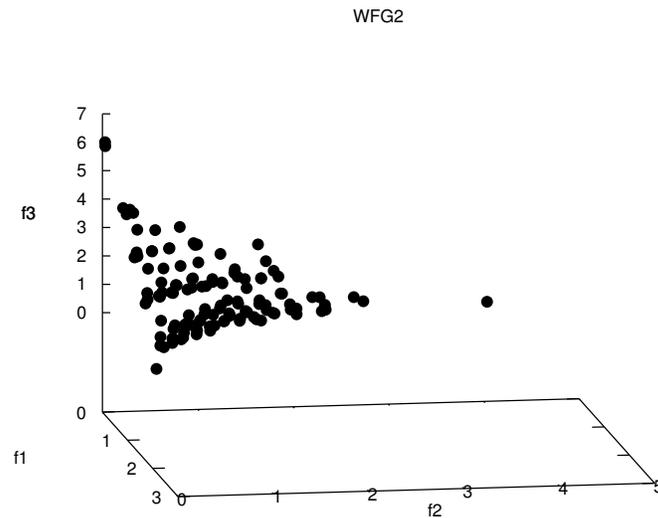


Figura B.15: Gráfica de la aproximación obtenida por $iMOACO_{\mathbb{R}}$ para WFG2 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen.

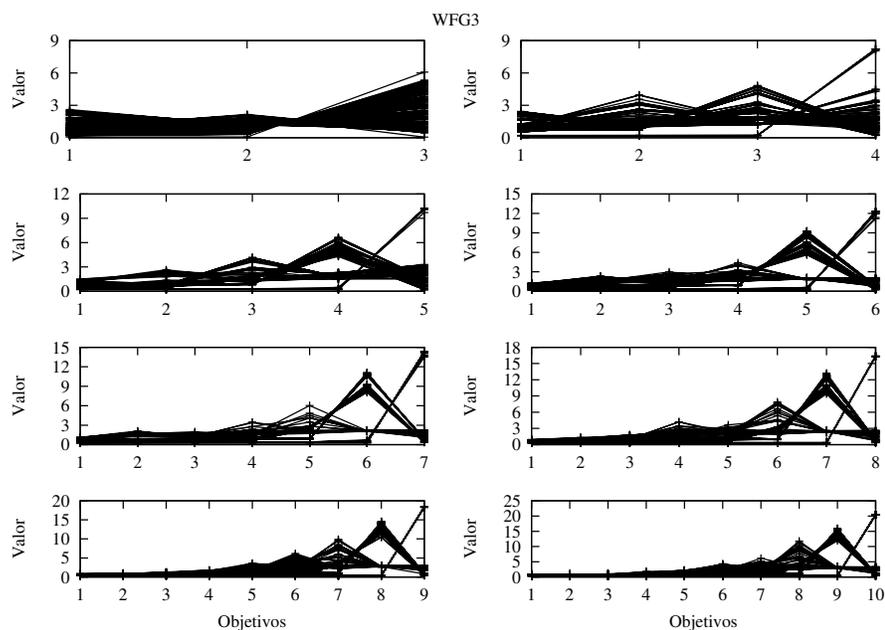


Figura B.16: Gráficas de las aproximaciones obtenidas por $iMOACO_{\mathbb{R}}$ para el problema WFG3 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes.

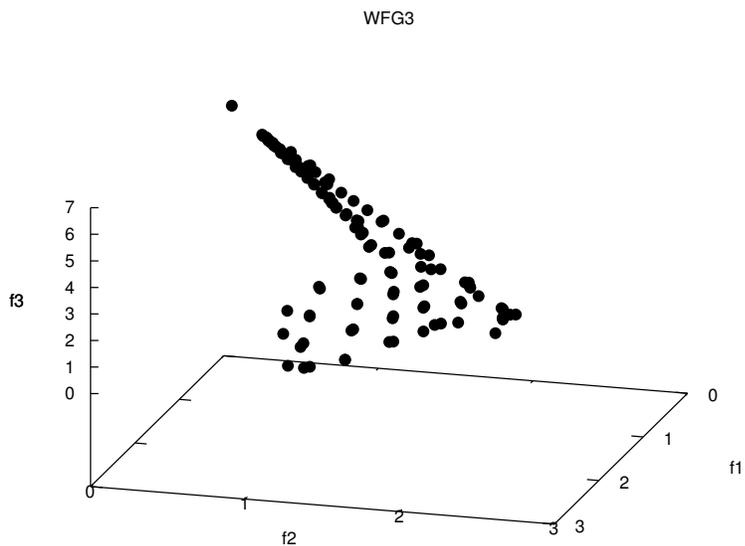


Figura B.17: Gráfica de la aproximación obtenida por $iMOACO_{\mathbb{R}}$ para WFG3 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen.

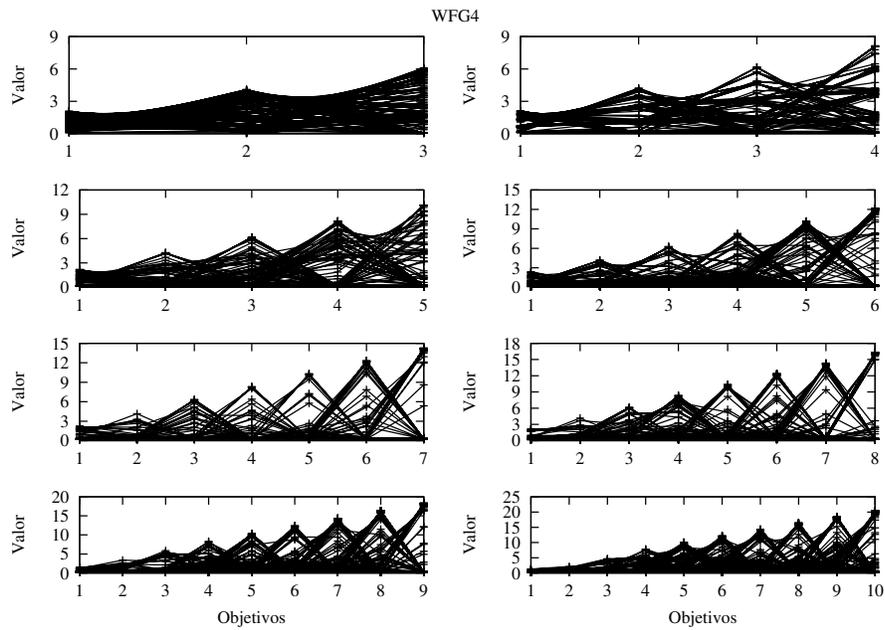


Figura B.18: Gráficas de las aproximaciones obtenidas por $iMOACO_{\mathbb{R}}$ para el problema WFG4 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes.

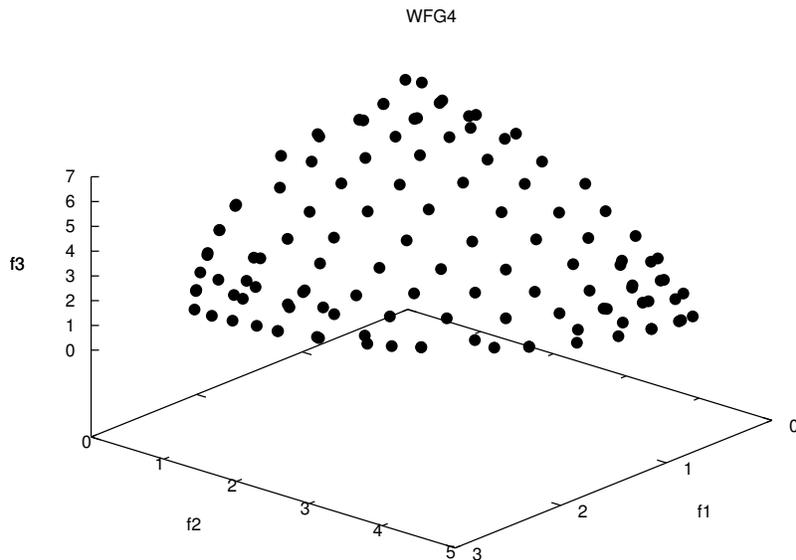


Figura B.19: Gráfica de la aproximación obtenida por $iMOACO_{\mathbb{R}}$ para WFG4 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen.

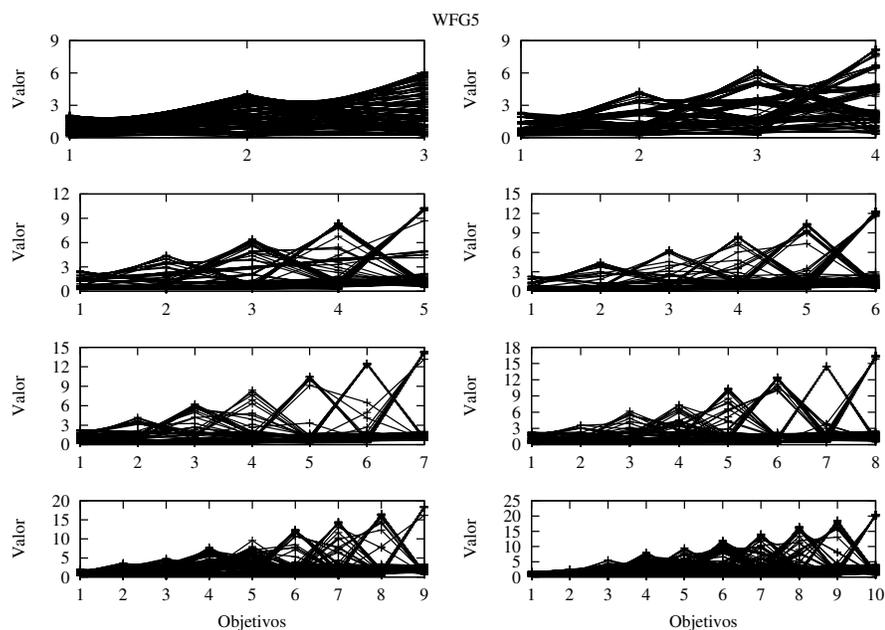


Figura B.20: Gráficas de las aproximaciones obtenidas por $iMOACO_{\mathbb{R}}$ para el problema WFG5 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes.

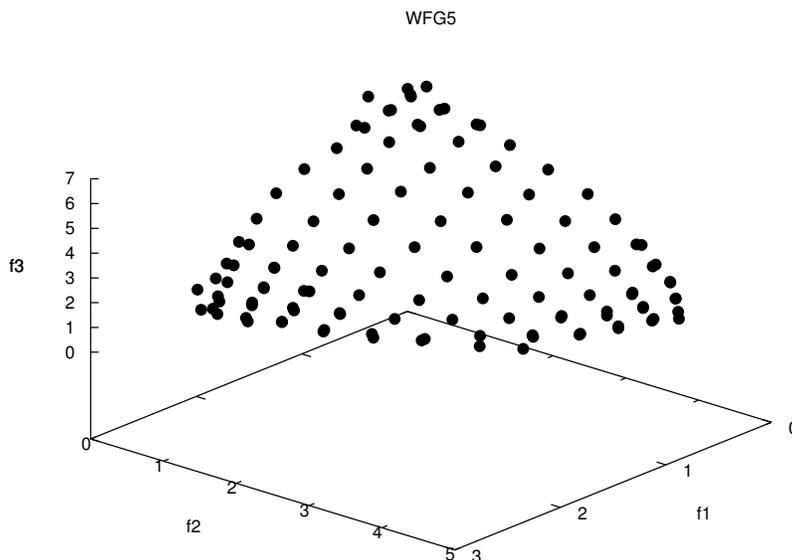


Figura B.21: Gráfica de la aproximación obtenida por $iMOACO_{\mathbb{R}}$ para WFG5 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen.

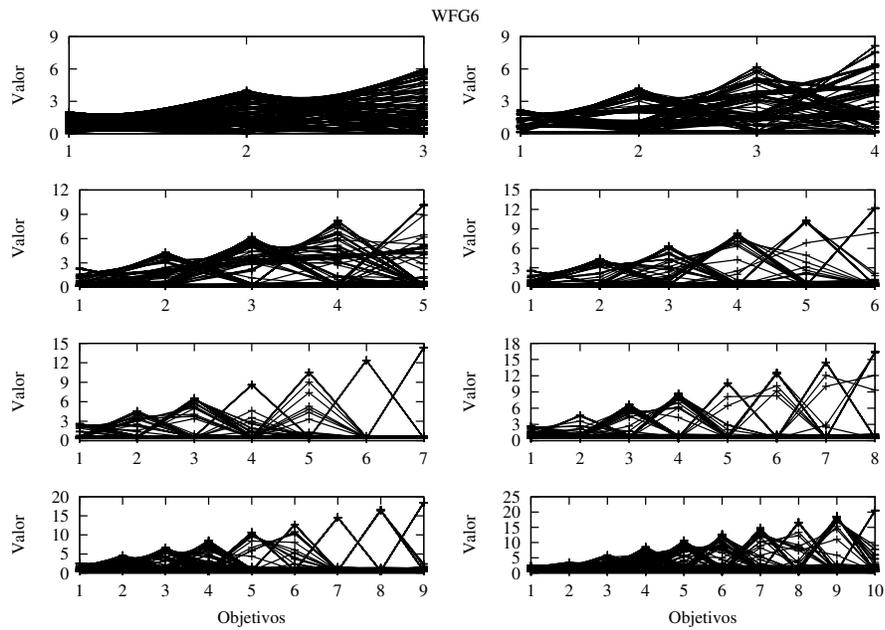


Figura B.22: Gráficas de las aproximaciones obtenidas por $iMOACO_{\mathbb{R}}$ para el problema WFG6 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes.

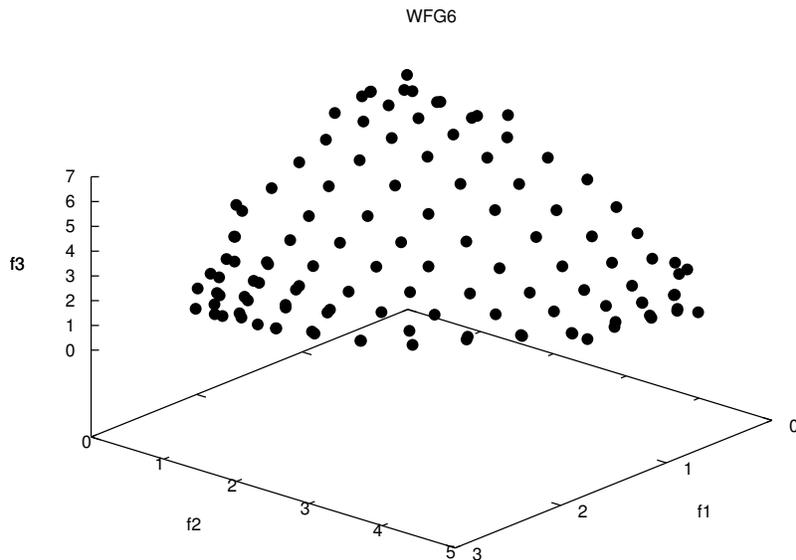


Figura B.23: Gráfica de la aproximación obtenida por $iMOACO_{\mathbb{R}}$ para WFG6 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen.

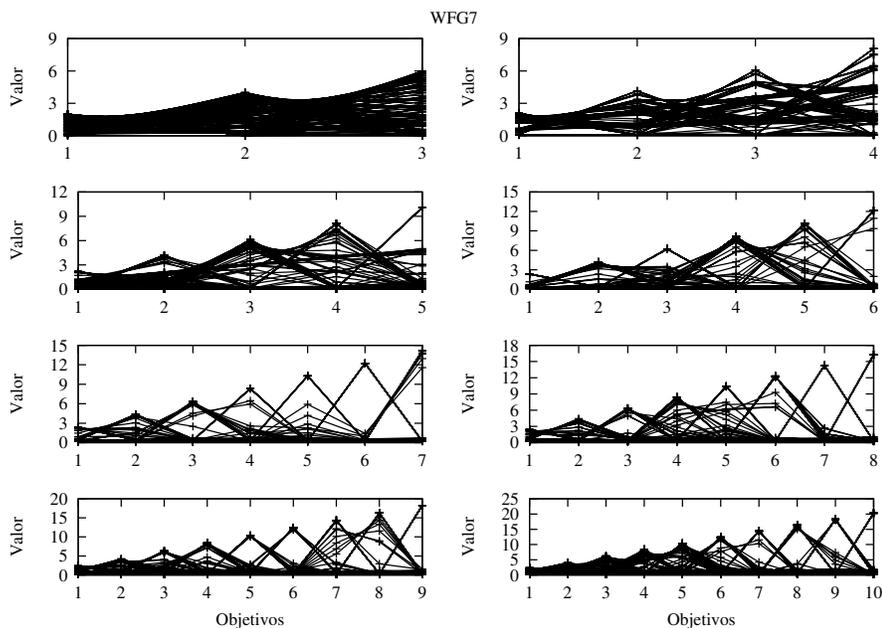


Figura B.24: Gráficas de las aproximaciones obtenidas por $iMOACO_{\mathbb{R}}$ para el problema WFG7 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes.

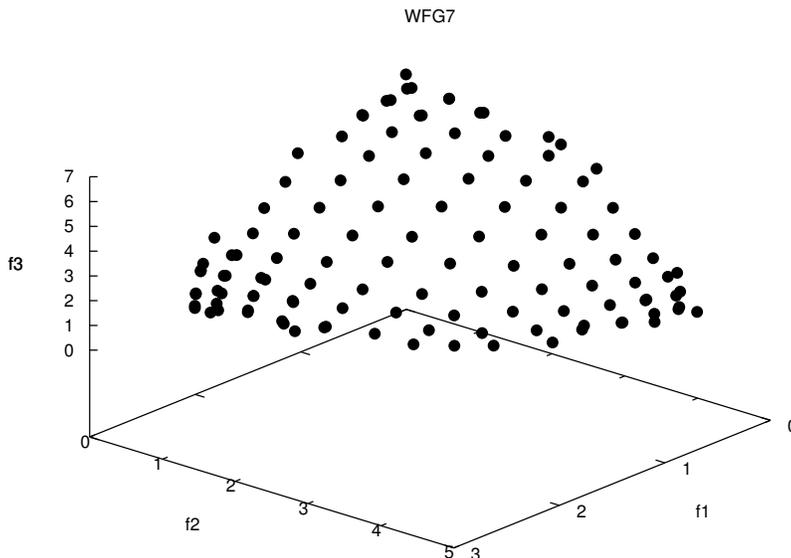


Figura B.25: Gráfica de la aproximación obtenida por $iMOACO_{\mathbb{R}}$ para WFG7 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen.

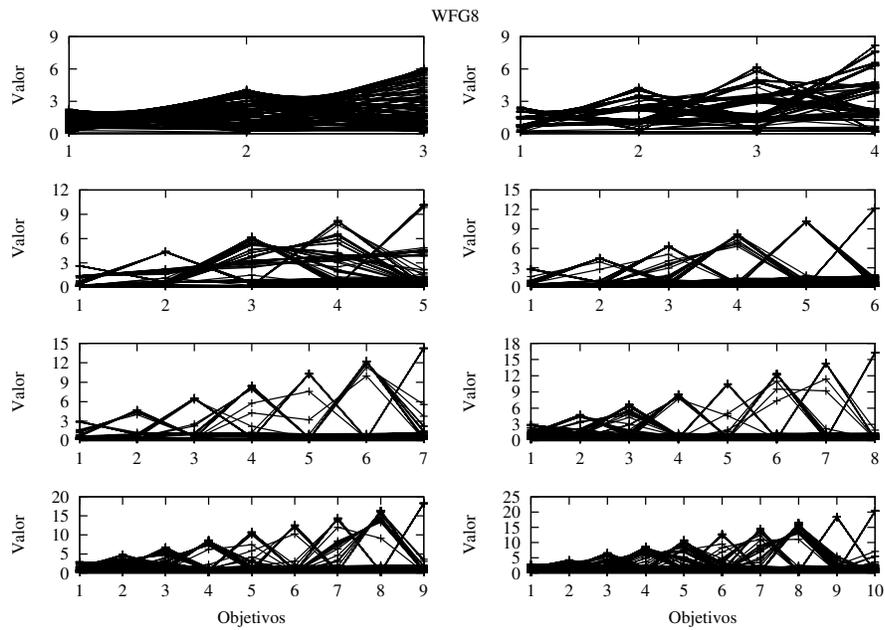


Figura B.26: Gráficas de las aproximaciones obtenidas por $iMOACO_{\mathbb{R}}$ para el problema WFG8 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes.

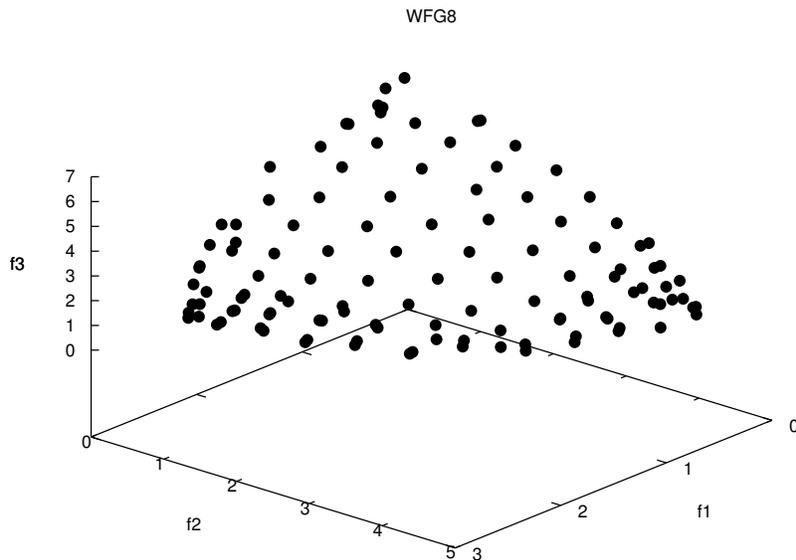


Figura B.27: Gráfica de la aproximación obtenida por $iMOACO_{\mathbb{R}}$ para WFG8 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen.

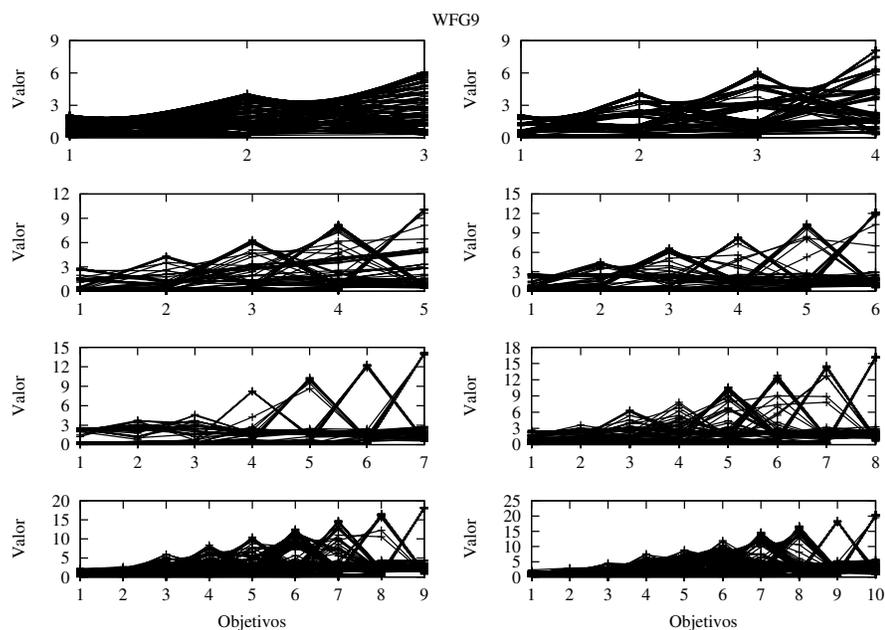


Figura B.28: Gráficas de las aproximaciones obtenidas por $iMOACO_{\mathbb{R}}$ para el problema WFG9 desde 3 hasta 10 objetivos. Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes.

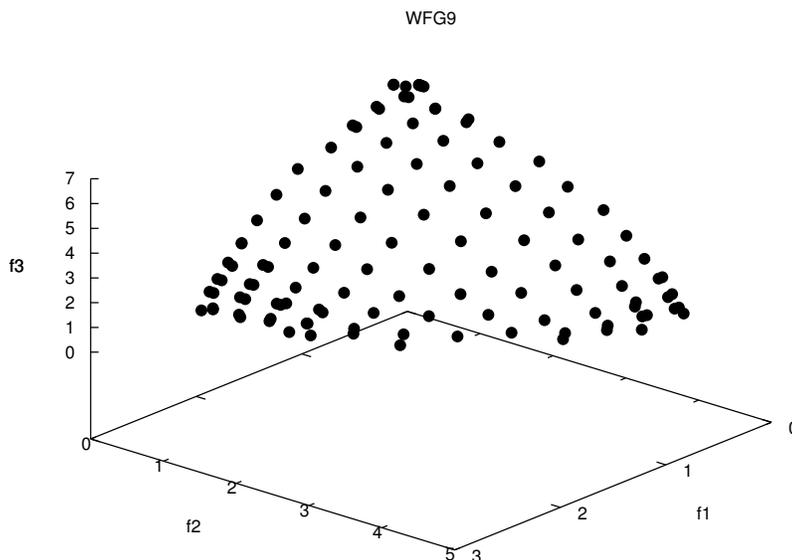


Figura B.29: Gráfica de la aproximación obtenida por $iMOACO_{\mathbb{R}}$ para WFG9 en 3 dimensiones. La aproximación corresponde con la mediana del valor de hipervolumen.

B.2. Resultados del experimento 2

Esta sección muestra los diagramas de caja y las representaciones de los intervalos de confianza de la prueba Tukey de los problemas elegidos en el estudio de sensibilidad de parámetros de iMOACO_ℝ. Sólo para DTLZ5 no se muestra el diagrama de intervalos de confianza al no hallarse pruebas estadísticamente significativas que indiquen sensibilidad ante la variación de parámetros.

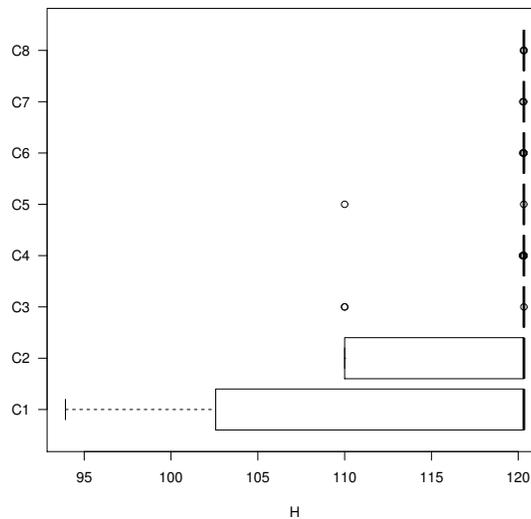


Figura B.30: Datos de estadística descriptiva para el problema ZDT2 empleando iMOACO_ℝ con diferentes combinaciones de parámetros.

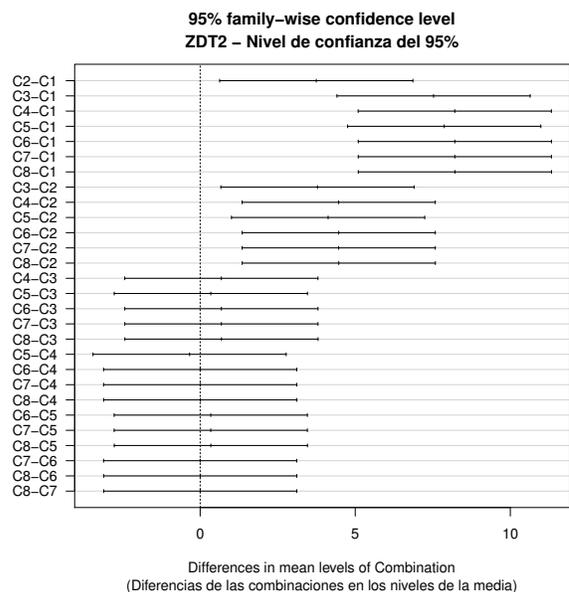


Figura B.31: Intervalos de confianza generados por la prueba Tukey en el problema ZDT2. Aquellos intervalos que intersectan el valor cero indican que no hay una diferencia estadísticamente significativa entre las combinaciones pareadas.

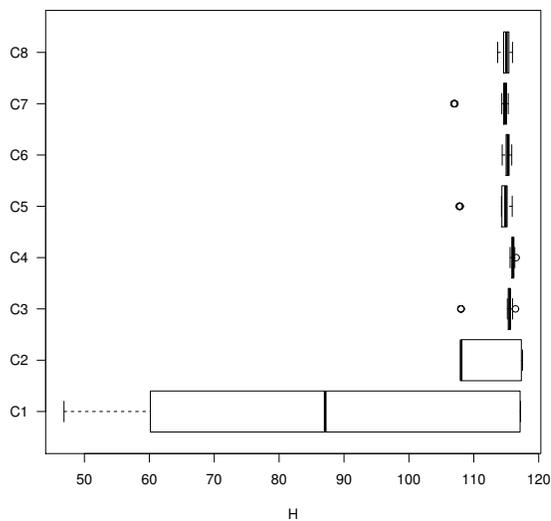


Figura B.32: Datos de estadística descriptiva para el problema ZDT6 empleando iMOACO_R con diferentes combinaciones de parámetros.

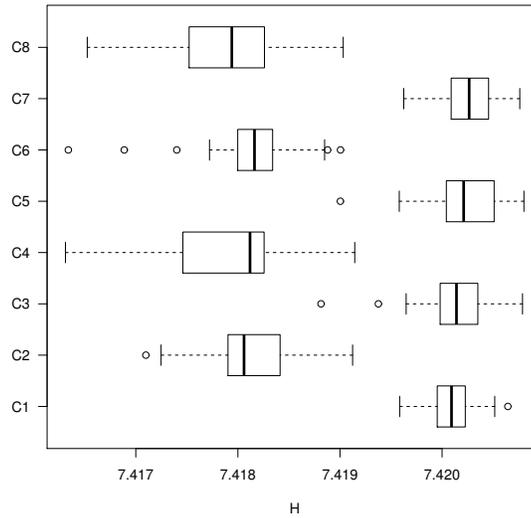


Figura B.34: Datos de estadística descriptiva para el problema DTLZ2 en tres dimensiones empleando iMOACO_ℝ con diferentes combinaciones de parámetros.

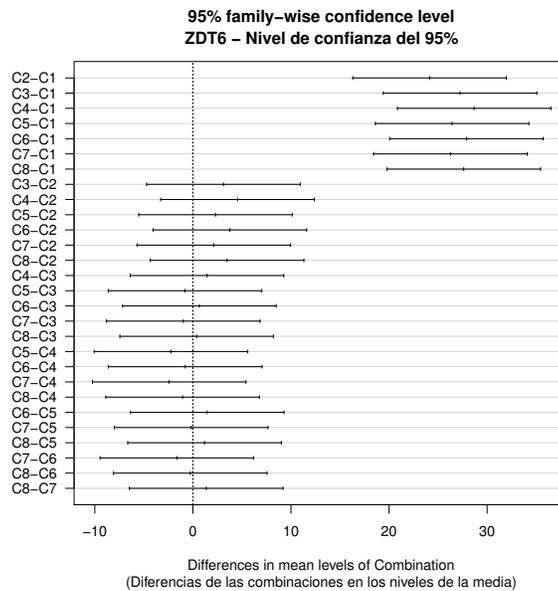


Figura B.33: Intervalos de confianza generados por la prueba Tukey en el problema ZDT6. Aquellos intervalos que intersectan el valor cero indican que no hay una diferencia estadísticamente significativa entre las combinaciones pareadas.

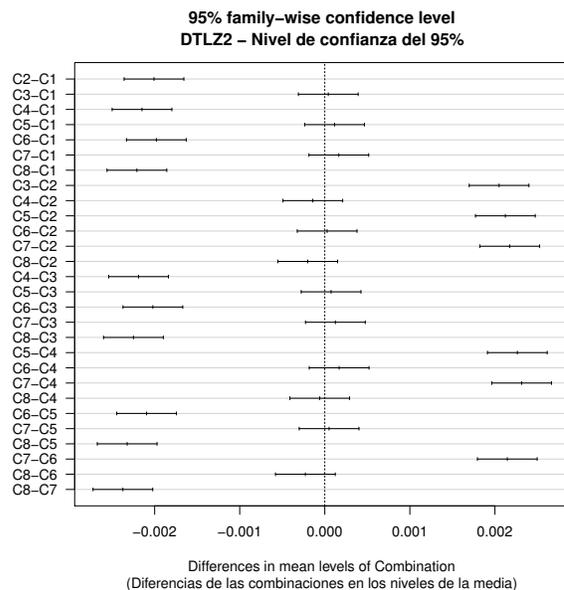


Figura B.35: Intervalos de confianza generados por la prueba Tukey en el problema DTLZ2 en tres dimensiones. Aquellos intervalos que intersectan el valor cero indican que no hay una diferencia estadísticamente significativa entre las combinaciones pareadas.

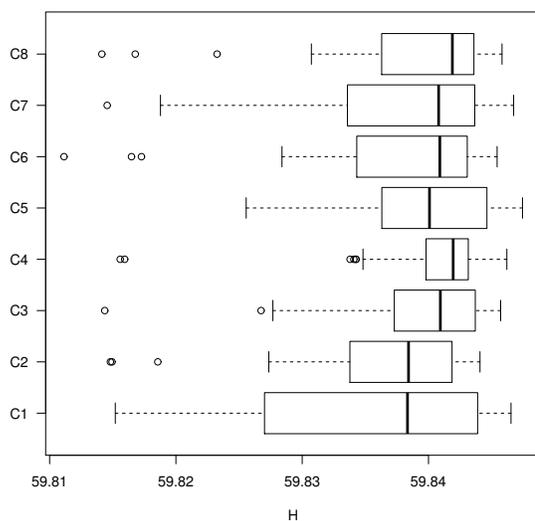


Figura B.36: Datos de estadística descriptiva para el problema DTLZ5 en tres dimensiones empleando iMOACO_R con diferentes combinaciones de parámetros.

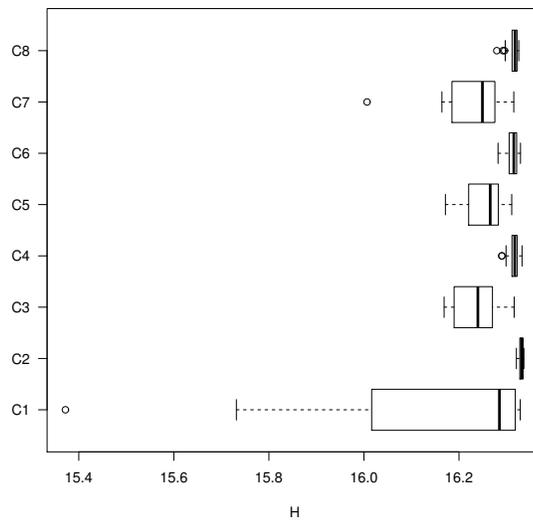


Figura B.37: Datos de estadística descriptiva para el problema DTLZ7 en tres dimensiones empleando iMOACO_R con diferentes combinaciones de parámetros.

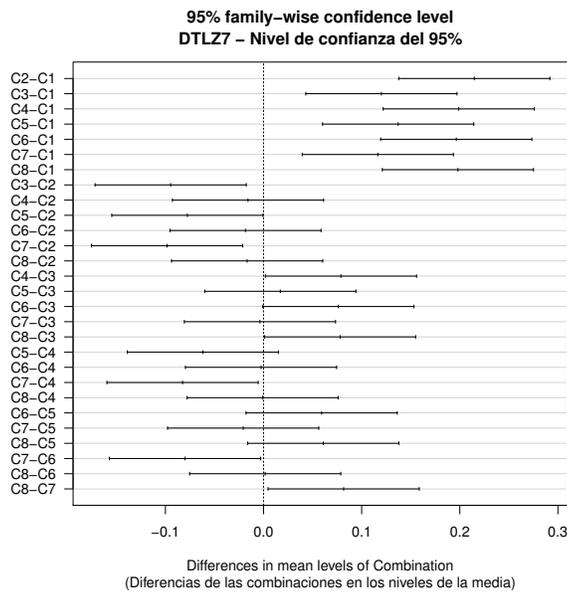


Figura B.38: Intervalos de confianza generados por la prueba Tukey en el problema DTLZ7 en tres dimensiones. Aquellos intervalos que intersectan el valor cero indican que no hay una diferencia estadísticamente significativa entre las combinaciones pareadas.

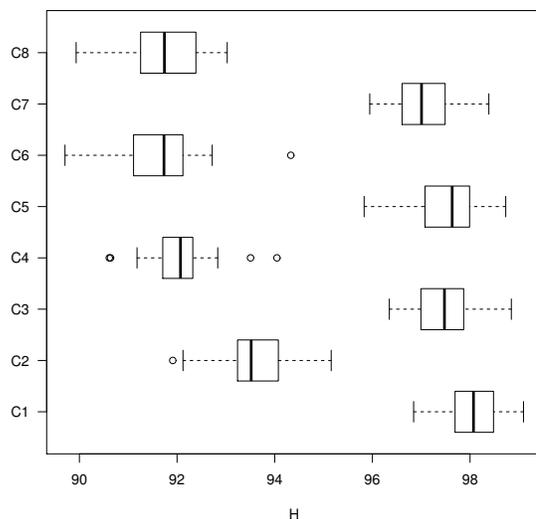


Figura B.39: Datos de estadística descriptiva para el problema WFG2 en tres dimensiones empleando iMOACO_R con diferentes combinaciones de parámetros.

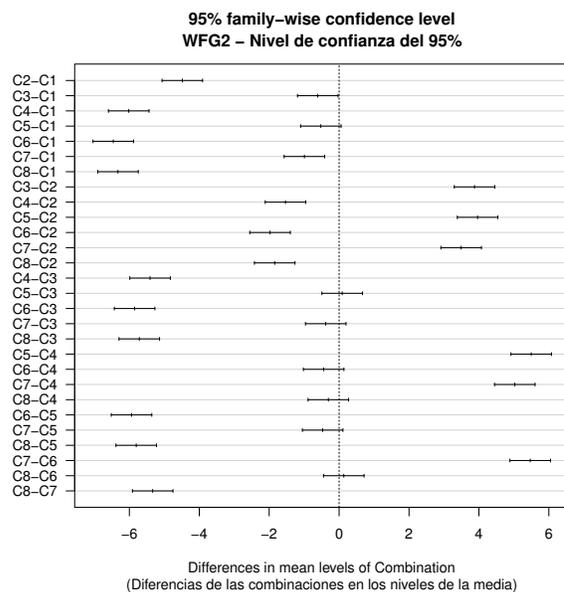


Figura B.40: Intervalos de confianza generados por la prueba Tukey en el problema WFG2 en tres dimensiones. Aquellos intervalos que intersectan el valor cero indican que no hay una diferencia estadísticamente significativa entre las combinaciones pareadas.

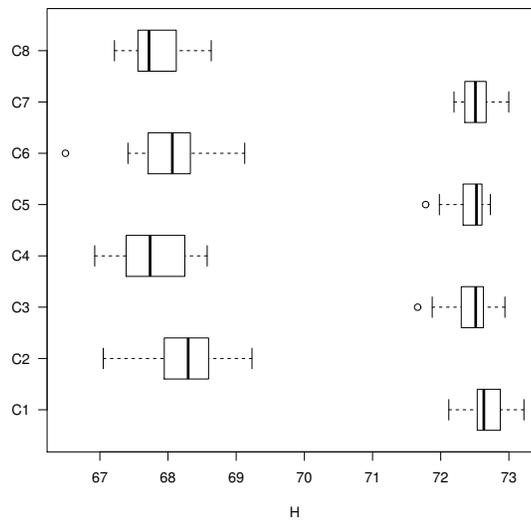


Figura B.41: Datos de estadística descriptiva para el problema WFG3 en tres dimensiones empleando iMOACO_R con diferentes combinaciones de parámetros.

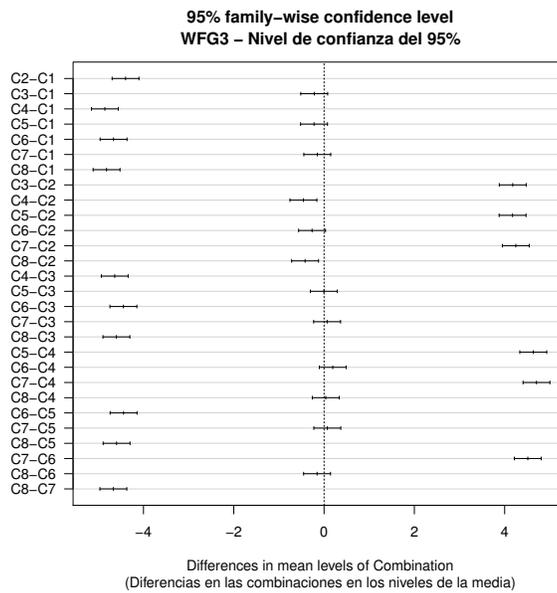


Figura B.42: Intervalos de confianza generados por la prueba Tukey en el problema WFG3 en tres dimensiones. Aquellos intervalos que intersectan el valor cero indican que no hay una diferencia estadísticamente significativa entre las combinaciones pareadas.

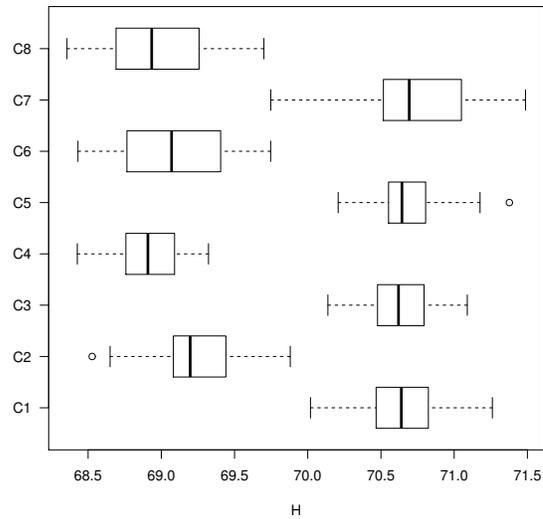


Figura B.43: Datos de estadística descriptiva para el problema WFG4 en tres dimensiones empleando iMOACO_R con diferentes combinaciones de parámetros.

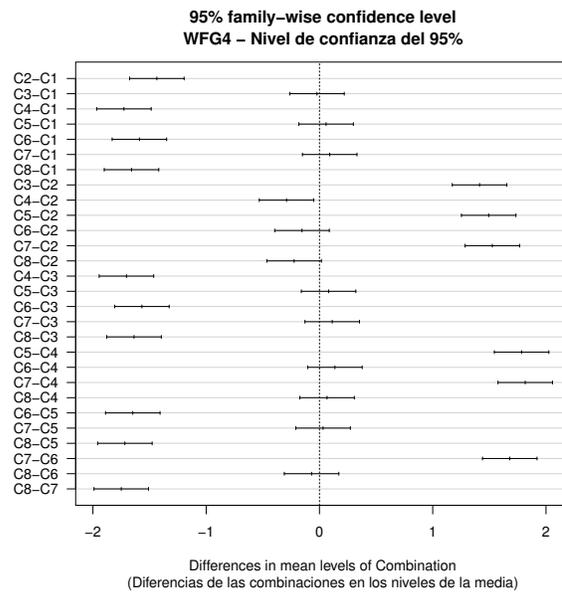


Figura B.44: Intervalos de confianza generados por la prueba Tukey en el problema WFG4 en tres dimensiones. Aquellos intervalos que intersectan el valor cero indican que no hay una diferencia estadísticamente significativa entre las combinaciones pareadas.

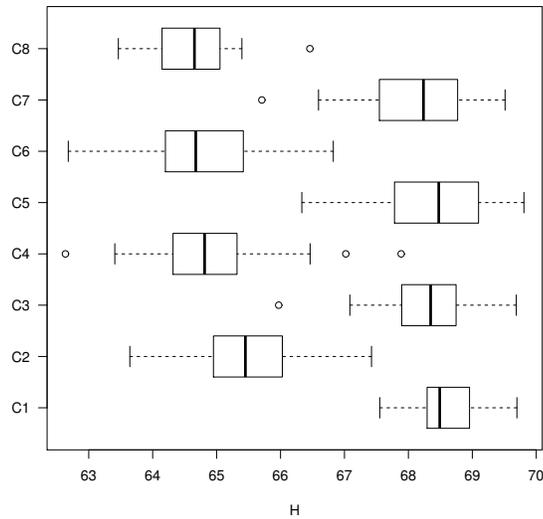


Figura B.45: Datos de estadística descriptiva para el problema WFG5 en tres dimensiones empleando iMOACO_R con diferentes combinaciones de parámetros.

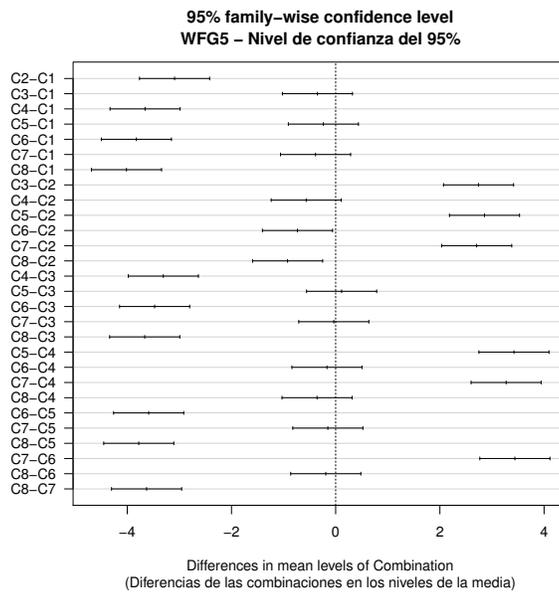


Figura B.46: Intervalos de confianza generados por la prueba Tukey en el problema WFG5 en tres dimensiones. Aquellos intervalos que intersectan el valor cero indican que no hay una diferencia estadísticamente significativa entre las combinaciones pareadas.

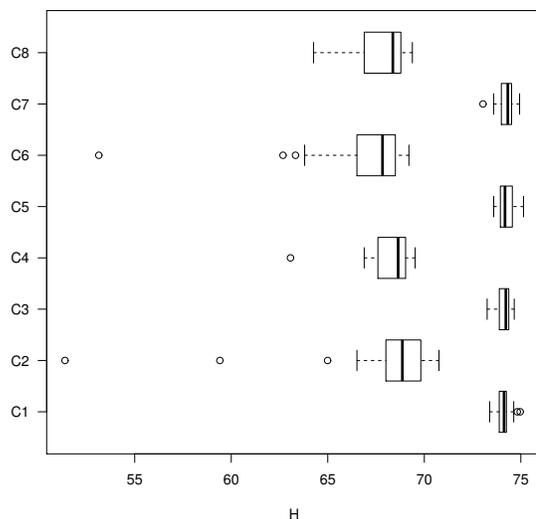


Figura B.47: Datos de estadística descriptiva para el problema WFG6 en tres dimensiones empleando iMOACO_R con diferentes combinaciones de parámetros.

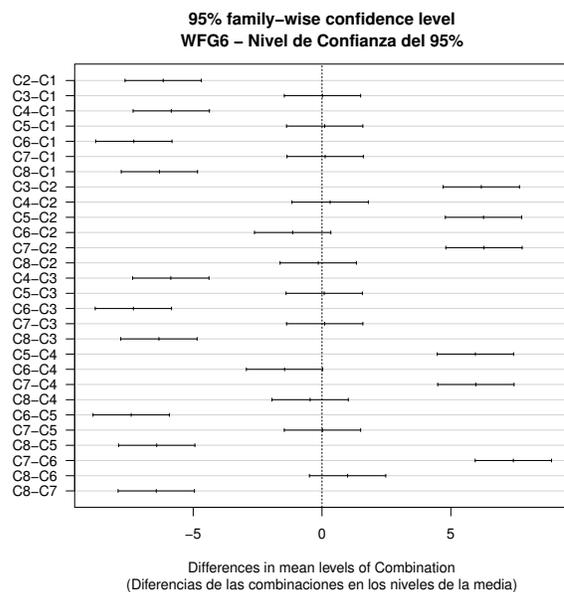


Figura B.48: Intervalos de confianza generados por la prueba Tukey en el problema WFG6 en tres dimensiones. Aquellos intervalos que intersectan el valor cero indican que no hay una diferencia estadísticamente significativa entre las combinaciones pareadas.

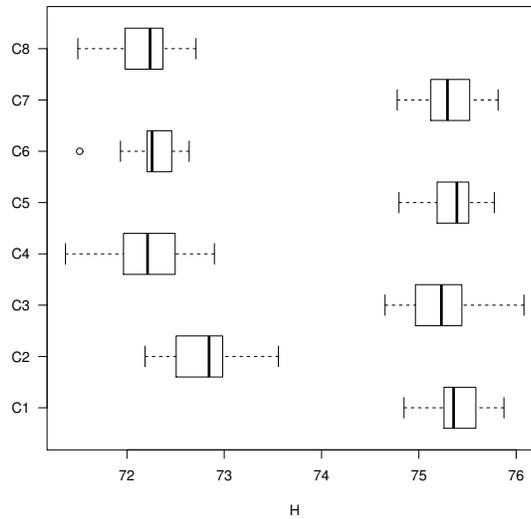


Figura B.49: Datos de estadística descriptiva para el problema WFG7 en tres dimensiones empleando iMOACO_R con diferentes combinaciones de parámetros.

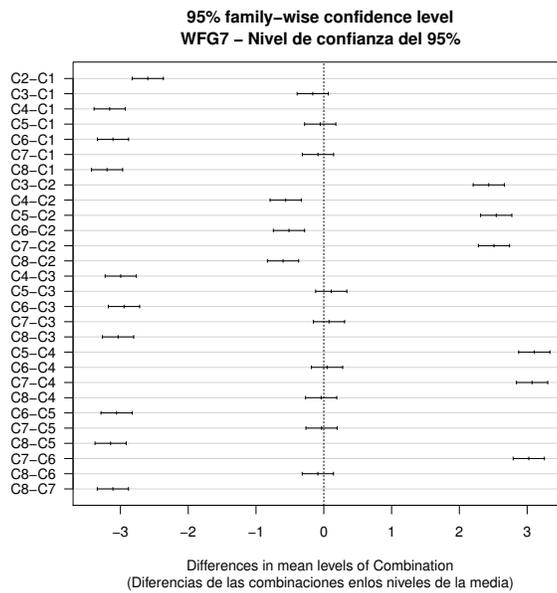


Figura B.50: Intervalos de confianza generados por la prueba Tukey en el problema WFG7 en tres dimensiones. Aquellos intervalos que intersectan el valor cero indican que no hay una diferencia estadísticamente significativa entre las combinaciones pareadas.

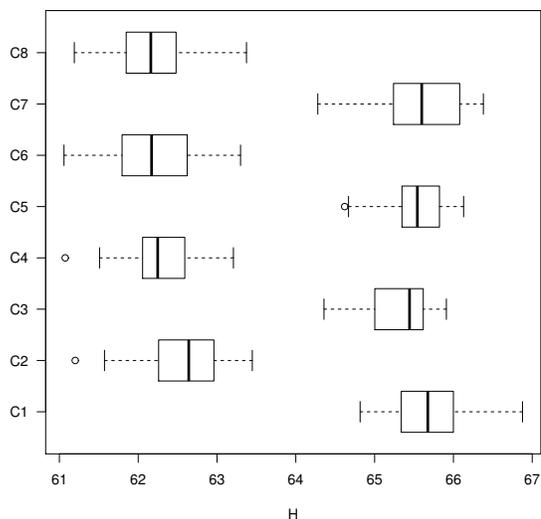


Figura B.51: Datos de estadística descriptiva para el problema WFG8 en tres dimensiones empleando iMOACO_R con diferentes combinaciones de parámetros.

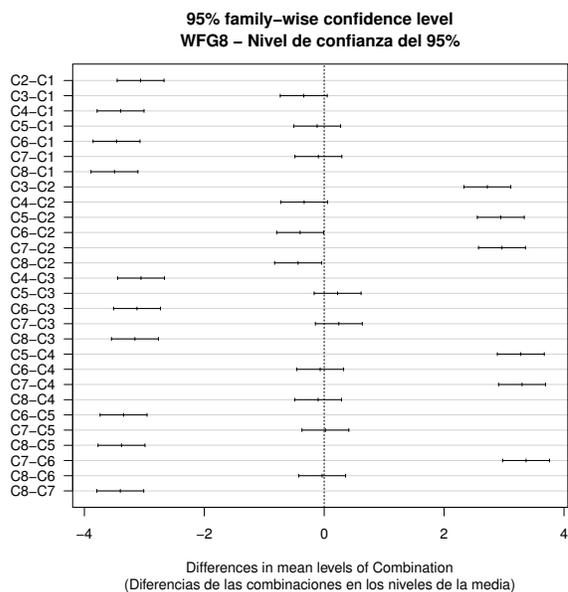


Figura B.52: Intervalos de confianza generados por la prueba Tukey en el problema WFG8 en tres dimensiones. Aquellos intervalos que intersectan el valor cero indican que no hay una diferencia estadísticamente significativa entre las combinaciones pareadas.

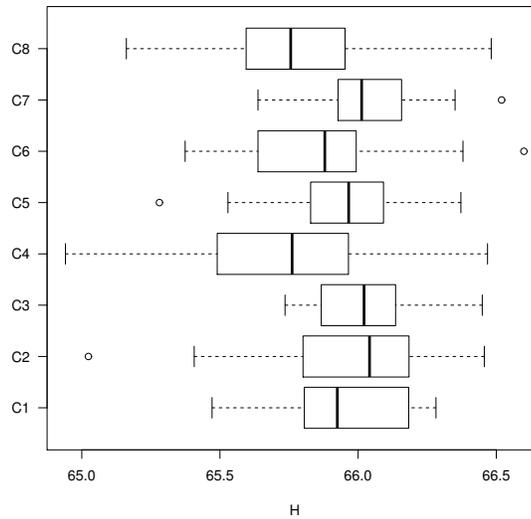


Figura B.53: Datos de estadística descriptiva para el problema WFG9 en tres dimensiones empleando iMOACO_R con diferentes combinaciones de parámetros.

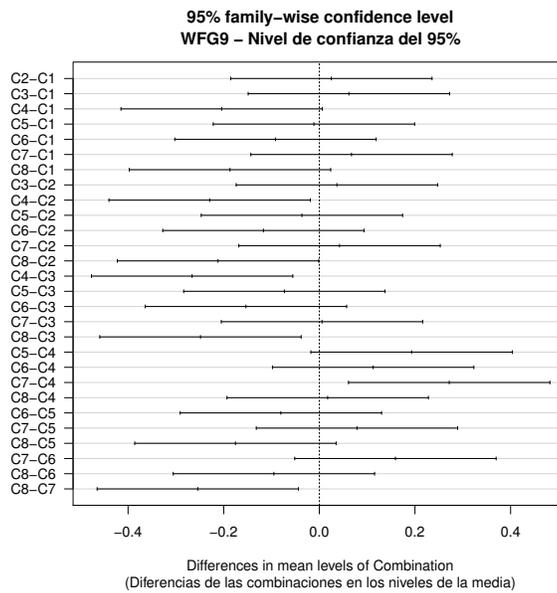


Figura B.54: Intervalos de confianza generados por la prueba Tukey en el problema WFG9 en tres dimensiones. Aquellos intervalos que intersectan el valor cero indican que no hay una diferencia estadísticamente significativa entre las combinaciones pareadas.

Bibliografía

- [1] Emile Aarts and Jan Korst. *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. John Wiley & Sons, Inc., New York, NY, USA, 1989.
- [2] I. Alaya, C. Solnon, and K. Ghedira. Ant colony optimization for multi-objective optimization problems. In *19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)*, volume 1, pages 450–457, Oct 2007.
- [3] D. Angus. Crowding population-based ant colony optimisation for the multi-objective travelling salesman problem. In *2007 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*, pages 333–340, Los Alamitos, 2007. IEEE.
- [4] D. Angus. Population-based ant colony optimisation for multi-objective function optimisation. In M. Randall, H. A. Abbass, and J. Wiles, editors, *Progress in Artificial Life*, volume 4828 of *Lecture Notes in Computer Science*, pages 232–244. Springer, 2007.
- [5] Daniel Angus and Clinton Woodward. Multiple objective ant colony optimisation. *Swarm Intelligence*, 3(1):69–85, 2009.
- [6] Tom M. Apostol. *Calculus, vol. II*. John Wiley & Sons, 1969.
- [7] Tom M. Apostol. *Calculus, volume I*. John Wiley & Sons, 2007.
- [8] David L. Applegate, Robert E. Bixby, Vasek Chvatal, and William J. Cook. *The Traveling Salesman Problem: A Computational Study (Princeton Series in Applied Mathematics)*. Princeton University Press, Princeton, NJ, USA, 2007.
- [9] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.
- [10] Johannes Bader and Eckart Zitzler. HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. *Evolutionary Computation*, 19(1):45–76, Spring, 2011.

- [11] Johannes M. Bader. *Hypervolume-Based Search for Multiobjective Optimization: Theory and Methods*. PhD thesis, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology Zürich, Zürich, Switzerland, 2009.
- [12] Benjamín Barán and Matilde Schaerer. A multiobjective ant colony system for vehicle routing problem with time windows. In *Applied Informatics*, pages 97–102, 2003.
- [13] Stefano Benedettini, Andrea Roli, and Luca Di Gaspero. Two-level aco for haplotype inference under pure parsimony. In Marco Dorigo, Mauro Birattari, Christian Blum, Maurice Clerc, Thomas Stützle, and Alan F. T. Winfield, editors, *Ant Colony Optimization and Swarm Intelligence: 6th International Conference, ANTS 2008*, pages 179–190, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
- [14] G. Beni. The concept of cellular robotic system. In *IEEE International Symposium on Intelligent Control*, pages 57–62. IEEE Press, August 1988.
- [15] G. Beni and J. Wang. Theoretical problems for the realization of distributed robotic systems. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 1914–1920, April 1991.
- [16] Gerardo Beni and Jing Wang. Swarm intelligence in cellular robotic systems. In Paolo Dario, Giulio Sandini, and Patrick Aebischer, editors, *Robots and Biological Systems: Towards a New Bionics?*, pages 703–712. Springer Berlin Heidelberg, Berlin, Heidelberg, 1993.
- [17] J. L. Bentley, H. T. Kung, M. Schkolnick, and C. D. Thompson. On the Average Number of Maxima in a Set of Vectors and Applications. *Journal of the ACM*, 25(4):536–543, oct 1978.
- [18] Nicola Beume, Boris Naujoks, and Michael Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 16 September 2007.
- [19] G. Bilchev and I. C. Parmee. The Ant Colony Metaphor for Searching Continuous Design Spaces. In T. C. Fogarty, editor, *Evolutionary Computing*, pages 25–39, Sheffield, UK, April 1995. AISB Workshop, Springer.
- [20] Christian Blum. Beam-ACO: Hybridizing Ant Colony Optimization with Beam Search: An Application to Open Shop Scheduling. *Computers and Operations Research*, 32(6):1565–1591, jun 2005.
- [21] Christian Blum and Maria J. Blesa. New Metaheuristic Approaches for the Edge-weighted K-cardinality Tree Problem. *Computers and Operations Research*, 32(6):1355–1377, jun 2005.

-
- [22] Christian Blum and Andrea Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computint Surveys*, 35(3):268–308, sep 2003.
- [23] Christian Blum, Mateu Yábar Vallès, and Maria J Blesa. An ant colony optimization algorithm for dna sequencing by hybridization. *Computers & Operations Research*, 35(11):3620–3635, 2008.
- [24] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, Inc., New York, NY, USA, 1999.
- [25] George EP Box, Mervin E Muller, et al. A note on the generation of random normal deviates. *The annals of mathematical statistics*, 29(2):610–611, 1958.
- [26] Gilles Brassard and Paul Bratley. *Algorithmics: Theory & Practice*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.
- [27] Dimo Brockhoff, Tobias Friedrich, and Frank Neumann. Analyzing Hypervolume Indicator Based Algorithms. In Günter Rudolph, Thomas Jansen, Simon Lucas, Carlo Poloni, and Nicola Beume, editors, *Parallel Problem Solving from Nature–PPSN X*, pages 651–660. Springer. Lecture Notes in Computer Science Vol. 5199, Dortmund, Germany, September 2008.
- [28] Dimo Brockhoff, Tobias Wagner, and Heike Trautmann. On the Properties of the $R2$ Indicator. In *2012 Genetic and Evolutionary Computation Conference (GECCO'2012)*, pages 465–472, Philadelphia, USA, July 2012. ACM Press. ISBN: 978-1-4503-1177-9.
- [29] Stephen P Brooks and Byron JT Morgan. Optimization using simulated annealing. *The Statistician*, pages 241–257, 1995.
- [30] Scott Camazine. *Self-organization in biological systems*. Princeton University Press, 2003.
- [31] Ling Chen, Jie Shen, Ling Qin, and Hongjian Chen. An improved ant colony algorithm in continuous optimization. *Journal of Systems Science and Systems Engineering*, 12(2):224–235, 2003.
- [32] Carlos A. Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, second edition, September 2007. ISBN 978-0-387-33254-3.
- [33] Carlos A. Coello Coello and Margarita Reyes Sierra. A Study of the Parallelization of a Coevolutionary Multi-Objective Evolutionary Algorithm. In Raúl Monroy, Gustavo Arroyo-Figueroa, Luis Enrique Sucar, and Humberto Sossa, editors, *Proceedings of the Third Mexican International Conference on Artificial*

- Intelligence (MICAI'2004)*, pages 688–697. Springer Verlag. Lecture Notes in Artificial Intelligence Vol. 2972, April 2004.
- [34] Carlos A. Coello Coello and Maximino Salazar Lechuga. MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization. In *Congress on Evolutionary Computation (CEC'2002)*, volume 2, pages 1051–1056, Piscataway, New Jersey, May 2002. IEEE Service Center.
- [35] Alberto Coloni, Marco Dorigo, Vittorio Maniezzo, et al. Distributed optimization by ant colonies. In *Proceedings of the first European conference on artificial life*, volume 142, pages 134–142. Paris, France, 1991.
- [36] John A. Cornell. *Experiments with mixtures: designs, models, and the analysis of mixture data*, volume 895. Wiley-Interscience, New York, 1990.
- [37] Daniele Costa and Alain Hertz. Ants can colour graphs. *Journal of the operational research society*, 48(3):295–305, 1997.
- [38] Luis M. de Campos, Juan M. Fernández-Luna, José A. Gámez, and José M. Puerta. Ant colony optimization for learning bayesian networks. *International Journal of Approximate Reasoning*, 31(3):291 – 311, 2002.
- [39] Luis M. de Campos, Martín Gámez, José Antonio, and José Miguel Puerta Castellón. Learning bayesian networks by ant colony optimisation: searching in two different spaces. *Mathware & soft computing. 2002*, 9(2), 2002.
- [40] Kenneth Alan De Jong. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, University of Michigan, Ann Arbor, MI, USA, 1975. AAI7609381.
- [41] Kalyanmoy Deb. Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. *Evolutionary Computation*, 7(3):205–230, Fall 1999.
- [42] Kalyanmoy Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001. ISBN 0-471-87339-X.
- [43] Kalyanmoy Deb and Ram B Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9(3):115–148, 1995.
- [44] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, Juan Julian Merelo, and Hans-Paul Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.

-
- [45] Kalyanmoy Deb and Himanshu 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, August 2014.
- [46] Kalyanmoy Deb and Abhishek Kumar. Interactive Evolutionary Multi-Objective Optimization and Decision-Making using Reference Direction Method. In Dirk Thierens, editor, *2007 Genetic and Evolutionary Computation Conference (GECCO'2007)*, volume 1, pages 781–788, London, UK, July 2007. ACM Press.
- [47] Kalyanmoy Deb, Kaisa Miettinen, and Shamik Chaudhuri. Toward an Estimation of Nadir Objective Vector Using a Hybrid of Evolutionary and Local Search Approaches. *IEEE Transactions On Evolutionary Computation*, 14(6):821–841, December 2010.
- [48] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable Multi-Objective Optimization Test Problems. In *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 825–830, Piscataway, New Jersey, May 2002. IEEE Service Center.
- [49] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable Test Problems for Evolutionary Multiobjective Optimization. In Ajith Abraham, Lakhmi Jain, and Robert Goldberg, editors, *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, pages 105–145. Springer, USA, 2005.
- [50] Matthijs den Besten, Thomas Stützle, and Marco Dorigo. Ant colony optimization for the total weighted tardiness problem. In Marc Schoenauer, Kalyanmoy Deb, Günther Rudolph, Xin Yao, Evelyne Lutton, Juan Julian Merelo, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature PPSN VI: 6th International Conference Paris, France, September 18–20, 2000 Proceedings*, pages 611–620, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.
- [51] J-L Deneubourg, Serge Aron, Simon Goss, and Jacques M Pasteels. The self-organizing exploratory pattern of the argentine ant. *Journal of insect behavior*, 3(2):159–168, 1990.
- [52] Francesco di Pierro, Shoon-Thiam Khu, and Dragan A. Savić. An Investigation on Preference Order Ranking Scheme for Multiobjective Evolutionary Optimization. *IEEE Transactions on Evolutionary Computation*, 11(1):17–45, February 2007.
- [53] Alan Díaz-Manríquez, Gregorio Toscano-Pulido, Carlos A. Coello Coello, and Ricardo Landa-Becerra. A Ranking Method Based on the $R2$ Indicator for Many-Objective Optimization. In *2013 IEEE Congress on Evolutionary*

- Computation (CEC'2013)*, pages 1523–1530, Cancún, México, 20-23 June 2013. IEEE Press. ISBN 978-1-4799-0454-9.
- [54] M. Dorigo and L. M. Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, Apr 1997.
- [55] Marco Dorigo. *Optimization, Learning and Natural Algorithms (in Italian)*. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy, 1992.
- [56] Marco Dorigo and Christian Blum. Ant colony optimization theory: A survey. *Theoretical Computer Science*, 344(2-3):243–278, November 2005.
- [57] Marco Dorigo and Luca Maria Gambardella. Ant colonies for the travelling salesman problem. *Biosystems*, 43(2):73 – 81, 1997.
- [58] Marco Dorigo, Vittorio Maniezzo, and A. Colorni. Positive feedback as a search strategy. Technical Report 91-016, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy, 1991.
- [59] Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. The Ant System: An Autocatalytic Optimizing Process. Technical Report 91-016, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy, 1991.
- [60] Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. The Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man and Cybernetics-Part B*, 26(1):29–41, February 1996.
- [61] Marco Dorigo and Thomas Stützle. *Ant Colony Optimization*. Bradford Company, Scituate, MA, USA, 2004.
- [62] Marco Dorigo and Thomas Stützle. Ant Colony Optimization: Overview and Recent Advances. In Michel Gendreau and Jean-Yves Potvin, editors, *Handbook of Metaheuristics*, pages 227–263. Springer US, Boston, MA, 2010.
- [63] J. Dréo and P. Siarry. Continuous Interacting Ant Colony Algorithm Based on Dense Heterarchy. *Future Generation Computer Systems*, 20(5):841–856, june 2004.
- [64] Richard Durbin. An analogue approach to the travelling salesman. *Nature*, 326:16, 1987.
- [65] Francis Ysidro Edgeworth. *Mathematical psychics: An essay on the application of mathematics to the moral sciences*, volume 10. Kegan Paul, 1881.

-
- [66] Jesús Guillermo Falcón-Cardona and Carlos A. Coello Coello. iMOACO_R: A New Indicator-Based Multi-objective Ant Colony Optimization Algorithm for Continuous Search Spaces. In Julia Handl, Emma Hart, R. Peter Lewis, Manuel López-Ibáñez, Gabriela Ochoa, and Ben Paechter, editors, *Parallel Problem Solving from Nature – PPSN XIV: 14th International Conference, Edinburgh, UK, September 17-21, 2016, Proceedings*, pages 389–398, Cham, 2016. Springer International Publishing.
- [67] M. Farina and P. Amato. On the Optimal Solution Definition for Many-criteria Optimization Problems. In *Proceedings of the NAFIPS-FLINT International Conference'2002*, pages 233–238, Piscataway, New Jersey, June 2002. IEEE Service Center.
- [68] M. Farina and P. Amato. A fuzzy definition of “optimality” for many-criteria optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics Part A—Systems and Humans*, 34(3):315–326, May 2004.
- [69] Daniela Favaretto, Elena Moretti, and Paola Pellegrini. Ant colony system for a vrp with multiple time windows and multiple visits. *Journal of Interdisciplinary Mathematics*, 10(2):263–284, 2007.
- [70] William Feller. *An introduction to probability theory and its applications*, volume I. Wiley, New York, 1968.
- [71] David B Fogel. Applying evolutionary programming to selected traveling salesman problems. *Cybernetics and systems*, 24(1):27–36, 1993.
- [72] Carlos M. Fonseca, Luís Paquete, and Manuel López-Ibáñez. An Improved Dimension-Sweep Algorithm for the Hypervolume Indicator. In *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, pages 3973–3979, Vancouver, BC, Canada, July 2006. IEEE.
- [73] Günther Füllerer, Karl F. Doerner, Richard F. Hartl, and Manuel Iori. Ant Colony Optimization for the Two-dimensional Loading Vehicle Routing Problem. *Computers and Operations Research*, 36(3):655–673, March 2009.
- [74] L. M. Gambardella, E. Taillard, and G. Agazzi. MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows. Technical report, Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale, 1999.
- [75] Luca Maria Gambardella and Marco Dorigo. An Ant Colony System Hybridized with a New Local Search for the Sequential Ordering Problem. *INFORMS Journal on Computing*, 12(3):237–255, July 2000.
- [76] Carlos García-Martínez, Oscar Cordón, and Francisco Herrera. A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria tsp. *European Journal of Operational Research*, 180(1):116–148, 2007.

- [77] A. Garcia-Najera and J. A. Bullinaria. Extending $ACO_{\mathbb{R}}$ to Solve Multi-Objective Problems. In G. M. Coghill, editor, *Proceedings of the UK Workshop on Computational Intelligence*. Imperial College United Kingdom, 2007.
- [78] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.
- [79] Fred Glover. Tabu search: part I. *ORSA Journal on computing*, 1(3):190–206, 1989.
- [80] Fred Glover. Tabu search: part II. *ORSA Journal on computing*, 2(1):4–32, 1990.
- [81] Fred Glover and Manuel Laguna. *Tabu Search*. Springer, 2013.
- [82] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.
- [83] David E. Goldberg and Jon Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application*, pages 41–49, Hillsdale, NJ, USA, 1987. L. Erlbaum Associates Inc.
- [84] Oded Goldreich. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, New York, NY, USA, 1 edition, 2008.
- [85] Simon Goss, Serge Aron, Jean-Louis Deneubourg, and Jacques Marie Pasteels. Self-organized shortcuts in the argentine ant. *Naturwissenschaften*, 76(12):579–581, 1989.
- [86] Pierre-Paul Grasse. Termitologia, tome ii. *Fondation des Sociétés. Construction*, 1984.
- [87] Plerre-P Grassé. La reconstruction du nid et les coordinations interindividuelles chezbellicositermes natalensis etcubitermes sp. la théorie de la stigmergie: Essai d’interprétation du comportement des termites constructeurs. *Insectes sociaux*, 6(1):41–80, 1959.
- [88] Ralph P Grimaldi. *Matemáticas discreta y combinatoria: introducción y aplicaciones*. Pearson Educación, 1998.
- [89] Viviane Grunert da Fonseca, Carlos M. Fonseca, and Andreia O. Hall. Inferential Performance Assessment of Stochastic Optimisers and the Attainment Function. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 213–225. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.

-
- [90] Michael Guntsch and Martin Middendorf. A Population Based Approach for ACO. In Stefano Cagnoni, Jens Gottlieb, Emma Hart, Martin Middendorf, and Günther R. Raidl, editors, *Applications of Evolutionary Computing: Evo-Workshops 2002: EvoCOP, EvoIASP, EvoSTIM/EvoPLAN Kinsale*, pages 72–81, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.
- [91] Michael Guntsch and Martin Middendorf. Applying population based aco to dynamic optimization problems. In *Proceedings of the Third International Workshop on Ant Algorithms*, ANTS '02, pages 111–122, London, UK, UK, 2002. Springer-Verlag.
- [92] Herman Haken. Synergetics. *Physics Bulletin*, 28(9):412, 1977.
- [93] Thomas Hanne. Global Multiobjective Optimization with Evolutionary Algorithms: Selection Mechanisms and Mutation Control. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 197–212. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.
- [94] Michael Pilegaard Hansen and Andrzej Jaszkievicz. *Evaluating the quality of approximations to the non-dominated set*. IMM, Department of Mathematical Modelling, Technical University of Denmark, 1998.
- [95] Michael Pilegaard Hansen and Andrzej Jaszkievicz. *Evaluating the quality of approximations to the non-dominated set*. IMM, Department of Mathematical Modelling, Technical University of Denmark, 1998.
- [96] Georges Harik. Finding multimodal solutions using restricted tournament selection. In *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 24–31. Morgan Kaufmann, 1995.
- [97] Zhenan He, Gary G. Yen, and Jun Zhang. Fuzzy-Based Pareto Optimality for Many-Objective Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, 18(2):269–285, April 2014.
- [98] Hugo Hernández and Christian Blum. Ant colony optimization for multicasting in static wireless ad-hoc networks. *Swarm Intelligence*, 3(2):125–148, 2009.
- [99] Raquel Hernández Gómez. A New Multi-Objective Evolutionary Algorithm based on the R2 Indicator. Master’s thesis, Departamento de Computación, CINVESTAV-IPN, 2013.
- [100] 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 (CEC’2013)*, pages 2488–2495, Cancún, México, 20–23 June 2013. IEEE Press. ISBN 978-1-4799-0454-9.

- [101] Raquel Hernández Gómez and Carlos A. Coello Coello. Improved Metaheuristic Based on the $R2$ Indicator for Many-Objective Optimization. In *2015 Genetic and Evolutionary Computation Conference (GECCO 2015)*, pages 679–686, Madrid, Spain, July 11-15 2015. ACM Press. ISBN 978-1-4503-3472-3.
- [102] Frederick S. Hillier and Gerald J. Lieberman. *Introduction to Operations Research, 4th Ed.* Holden-Day, Inc., San Francisco, CA, USA, 1986.
- [103] John H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence.* MIT Press, Cambridge, MA, USA, 1992.
- [104] Simon Huband, Phil Hingston, Luigi Barone, and Lyndon While. A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, October 2006.
- [105] Philip Husbands. Genetic algorithms in optimisation and adaptation. In *Advances in Parallel Algorithms*, pages 227–276. John Wiley & Sons, Inc., 1992.
- [106] C. L. Hwang and Abu Syed Md Masud. *Multiple objective decision making methods and applications: a state of the art survey*, volume 164. Springer Science and Business Media, 2012.
- [107] Steffen Iredi, Daniel Merkle, and Martin Middendorf. Bi-Criterion Optimization with Multi Colony Ant Algorithms. In Eckart Zitzler, Lothar Thiele, Kalyanmoy Deb, Carlos Artemio Coello Coello, and David Corne, editors, *Evolutionary Multi-Criterion Optimization: First International Conference, EMO 2001*, pages 359–372, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.
- [108] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima. Difficulties in specifying reference points to calculate the inverted generational distance for many-objective optimization problems. In *2014 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM'2014)*, pages 170–177. IEEE Press, December 2014.
- [109] Hisao Ishibuchi, Hiroyuki Masuda, and Yusuke Nojima. A Study on Performance Evaluation Ability of a Modified Inverted Generational Distance Indicator. In *2015 Genetic and Evolutionary Computation Conference (GECCO 2015)*, pages 695–702, Madrid, Spain, July 11-15 2015. ACM Press. ISBN 978-1-4503-3472-3.
- [110] Hisao Ishibuchi, Hiroyuki Masuda, Yuki Tanigaki, and Yusuke Nojima. Modified Distance Calculation in Generational Distance and Inverted Generational Distance. In António Gaspar-Cunha, Carlos Henggeler Antunes, and Carlos Coello Coello, editors, *Evolutionary Multi-Criterion Optimization, 8th International Conference, EMO 2015*, pages 110–125. Springer. Lecture Notes in Computer Science Vol. 9019, Guimarães, Portugal, March 29 - April 1 2015.

-
- [111] Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. Iterative Approach to Indicator-Based Multiobjective Optimization. In *2007 IEEE Congress on Evolutionary Computation (CEC'2007)*, pages 3967–3974, Singapore, September 2007. IEEE Press.
- [112] Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. Evolutionary many-objective optimization: A short review. In *2008 Congress on Evolutionary Computation (CEC'2008)*, pages 2424–2431, Hong Kong, June 2008. IEEE Service Center.
- [113] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, September 1999.
- [114] Robert L Jeanne. The evolution of the organization of work in social insects. *Monitore Zoologico Italiano-Italian Journal of Zoology*, 20(2):119–133, 1986.
- [115] R. Johnson and M. G. Pilcher. The traveling salesman problem. *Networks*, pages 253–254, 1989.
- [116] James Kennedy, James F Kennedy, Russell C Eberhart, and Yuhui Shi. *Swarm intelligence*. Morgan Kaufmann, 2001.
- [117] Joshua Knowles and David Corne. On Metrics for Comparing Nondominated Sets. In *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 711–716, Piscataway, New Jersey, May 2002. IEEE Service Center.
- [118] Joshua Knowles and David Corne. Quantifying the Effects of Objective Space Dimension in Evolutionary Multiobjective Optimization. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007*, pages 757–771, Matshushima, Japan, March 2007. Springer. Lecture Notes in Computer Science Vol. 4403.
- [119] Joshua D. Knowles. *Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization*. PhD thesis, The University of Reading, Department of Computer Science, Reading, UK, January 2002.
- [120] Ikeda Kokolo, Kita Hajime, and Kobayashi Shigenobu. Failure of Pareto-based MOEAs: Does Non-dominated Really Mean Near to Optimal? In *Proceedings of the Congress on Evolutionary Computation 2001 (CEC'2001)*, volume 2, pages 957–962, Piscataway, New Jersey, May 2001. IEEE Service Center.
- [121] Min Kong and Peng Tian. A direct application of ant colony optimization to function optimization problem in continuous domain. In *Proceedings of the 5th International Conference on Ant Colony Optimization and Swarm Intelligence, ANTS'06*, pages 324–331, Berlin, Heidelberg, 2006. Springer-Verlag.

- [122] Oliver Korb, Thomas Stützle, and Thomas E. Exner. Plants: Application of ant colony optimization to structure-based drug design. In Marco Dorigo, Luca Maria Gambardella, Mauro Birattari, Alcherio Martinoli, Riccardo Poli, and Thomas Stützle, editors, *Ant Colony Optimization and Swarm Intelligence: 5th International Workshop, ANTS 2006*, pages 247–258, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
- [123] Oliver Korb, Thomas Stützle, and Thomas E. Exner. An ant colony optimization approach to flexible protein–ligand docking. *Swarm Intelligence*, 1(2):115–134, 2007.
- [124] Gary B. Lamont, Scott M. Brown, and George H. Gates. Evolutionary algorithms combined with deterministic search. In V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, editors, *Evolutionary Programming VII: 7th International Conference*, pages 515–526, San Diego, California, March 1998. Springer Berlin Heidelberg.
- [125] Marco Laumanns, Lothar Thiele, Kalyanmoy Deb, and Eckart Zitzler. Combining Convergence and Diversity in Evolutionary Multi-objective Optimization. *Evolutionary Computation*, 10(3):263–282, Fall 2002.
- [126] G. Leguizamón and Z. Michalewicz. A New Version of Ant System for Subset Problems. In *IEEE Congress on Evolutionary Computation (CEC'1999)*, volume 2, page 1464 Vol. 2, 1999.
- [127] Guillermo Leguizamón and Carlos A. Coello Coello. Multi-Objective Ant Colony Optimization: A Taxonomy and Review of Approaches. In Satchidanada Dehuri, Susmita Ghosh, and Sung Bae Cho, editors, *Integration of Swarm Intelligence and Artificial Neural Networks*, chapter 3, pages 67–94. World Scientific, Singapore, 2011.
- [128] Lucas Lessing, Irina Dumitrescu, and Thomas Stützle. A comparison between aco algorithms for the set covering problem. In Marco Dorigo, Mauro Birattari, Christian Blum, Luca Maria Gambardella, Francesco Mondada, and Thomas Stützle, editors, *Ant Colony Optimization and Swarm Intelligence: 4th International Workshop, ANTS 2004*, pages 1–12, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
- [129] K. Li, K. Deb, Q. Zhang, and S. Kwong. An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition. *IEEE Transactions on Evolutionary Computation*, 19(5):694–716, Oct 2015.
- [130] Miqing Li, Jinhua Zheng, Ruimin Shen, Ke Li, and Qizhao Yuan. A Grid-Based Fitness Strategy for Evolutionary Many-Objective Optimization. In *Proceedings of the 12th annual conference on Genetic and Evolutionary Computation (GECCO'2010)*, pages 463–470, Portland, Oregon, USA, July 7–11 2010. ACM Press. ISBN 978-1-4503-0072-8.

-
- [131] Tianjun Liao, Krzysztof Socha, Marco A. Montes de Oca, and Thomas Stützle. Ant Colony Optimization for Mixed-Variable Optimization Problems. *IEEE Transactions on Evolutionary Computation*, 18(4):503–518, 2014.
- [132] Shen Lin. Computer solutions of the traveling salesman problem. *The Bell System Technical Journal*, 44(10):2245–2269, 1965.
- [133] H. L. Liu, F. Gu, and Q. Zhang. Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. *IEEE Transactions on Evolutionary Computation*, 18(3):450–455, June 2014.
- [134] Nan Liu, Bo Huang, and Xiaohong Pan. Using the ant algorithm to derive pareto fronts for multiobjective siting of emergency service facilities. *Transportation Research Record: Journal of the Transportation Research Board*, (1935):120–129, 2005.
- [135] Manuel López-Ibáñez, Christian Blum, Dhananjay Thiruvady, Andreas T. Ernst, and Bernd Meyer. *Beam-ACO Based on Stochastic Sampling for Makespan Optimization Concerning the TSP with Time Windows*, pages 97–108. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [136] Manuel López-Ibáñez, Luís Paquete, and Thomas Stützle. On the Design of ACO for the Biobjective Quadratic Assignment Problem. In Marco Dorigo, Mauro Birattari, Christian Blum, Luca Maria Gambardella, Francesco Mondada, and Thomas Stützle, editors, *Ant Colony Optimization and Swarm Intelligence: 4th International Workshop, ANTS 2004*, pages 214–225, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
- [137] Samir W Mahfoud. *Niching methods for genetic algorithms*. PhD thesis, University of Illinois, 1995.
- [138] Vittorio Maniezzo. Exact and Approximate Nondeterministic Tree-Search Procedures for the Quadratic Assignment Problem. *INFORMS Journal on Computing*, 11(4):358–369, April 1999.
- [139] Vittorio Maniezzo and Antonella Carbonaro. An Ants Heuristic for the Frequency Assignment Problem. *Future Generation Computer Systems*, 16(9):927–935, June 2000.
- [140] D. Martens, M. De Backer, R. Haesen, J. Vanthienen, M. Snoeck, and B. Baeens. Classification with ant colony optimization. *IEEE Transactions on Evolutionary Computation*, 11(5):651–665, Oct 2007.
- [141] William Scheaffer Mendenhall, Richard L Wackerly, and D Dennis. *Estadística matemática con aplicaciones*. Grupo Editorial Iberoamérica,, 1994.

- [142] D. Merkle, M. Middendorf, and H. Schmeck. Ant colony optimization for resource-constrained project scheduling. *IEEE Transactions on Evolutionary Computation*, 6(4):333–346, Aug 2002.
- [143] Kaisa Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publisher, 1999.
- [144] Sanaz Mostaghim and Hartmut Schmeck. Distance Based Ranking in Many-Objective Particle Swarm Optimization. In Günter Rudolph, Thomas Jansen, Simon Lucas, Carlo Poloni, and Nicola Beume, editors, *Parallel Problem Solving from Nature—PPSN X*, pages 753–762. Springer. Lecture Notes in Computer Science Vol. 5199, Dortmund, Germany, September 2008.
- [145] Gregoire Nicolis, Ilya Prigogine, et al. *Self-organization in nonequilibrium systems*, volume 191977. Wiley, New York, 1977.
- [146] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.
- [147] Shigeru Obayashi and Daisuke Sasaki. Visualization and Data Mining of Pareto Solutions Using Self-Organizing Map. In Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele, editors, *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, pages 796–809, Faro, Portugal, April 2003. Springer. Lecture Notes in Computer Science. Volume 2632.
- [148] Fernando E. B. Otero, Alex A. Freitas, and Colin G. Johnson. cant-miner: An ant colony classification algorithm to cope with continuous attributes. In Marco Dorigo, Mauro Birattari, Christian Blum, Maurice Clerc, Thomas Stützle, and Alan F. T. Winfield, editors, *Ant Colony Optimization and Swarm Intelligence: 6th International Conference (ANTS 2008)*, pages 48–59, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
- [149] V. Pareto. *Manual of political economy*. Scholars Book Shelf, 1971.
- [150] Vilfredo Pareto. *Cours d'économie politique*. Librairie Droz, 1964.
- [151] R. S. Parpinelli, H. S. Lopes, and A. A. Freitas. Data mining with an ant colony optimization algorithm. *IEEE Transactions on Evolutionary Computation*, 6(4):321–332, Aug 2002.
- [152] Judea Pearl. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1984.
- [153] Robin C. Purshouse and Peter J. Fleming. Evolutionary Multi-Objective Optimisation: An Exploratory Analysis. In *Proceedings of the 2003 Congress on*

-
- Evolutionary Computation (CEC'2003)*, volume 3, pages 2066–2073, Canberra, Australia, December 2003. IEEE Press.
- [154] Chandrasekharan Rajendran and Hans Ziegler. Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs. *European Journal of Operational Research*, 155(2):426–438, 2004.
- [155] Marc Reimann, Karl Doerner, and Richard F. Hartl. D-Ants: Savings Based Ants Divide and Conquer the Vehicle Routing Problem. *Computers and Operations Research*, 31(4):563–591, April 2004.
- [156] Gerhard Reinelt. *The Traveling Salesman: Computational Solutions for TSP Applications*. Springer-Verlag, Berlin, Heidelberg, 1994.
- [157] Edward M. Reingold, Jurg Nievergelt, and Narsingh Deo. *Combinatorial Algorithms: Theory and Practice*. Prentice Hall College Div, 1977.
- [158] Gene E Robinson. Regulation of division of labor in insect societies. *Annual review of entomology*, 37(1):637–665, 1992.
- [159] Cynthia A. Rodríguez Villalobos and Carlos A. Coello Coello. A New Multi-Objective Evolutionary Algorithm Based on a Performance Assessment Indicator. In *2012 Genetic and Evolutionary Computation Conference (GECCO'2012)*, pages 505–512, Philadelphia, USA, July 2012. ACM Press. ISBN: 978-1-4503-1177-9.
- [160] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition, 2003.
- [161] Lamjed Ben Said, Slim Bechikh, and Khaled Ghédira. The r-Dominance: A New Dominance Relation for Interactive Evolutionary Multicriteria Decision Making. *IEEE Transactions on Evolutionary Computation*, 14(5):801–818, October 2010.
- [162] Hiroyuki Sato, Hernán E. Aguirre, and Kiyoshi Tanaka. Self-Controlling Dominance Area of Solutions in Evolutionary Many-Objective Optimization. In Kalyanmoy Deb, Arnab Bhattacharya, Nirupam Chakraborti, Partha Chakraborty, Swagatam Das, Joydeep Dutta, Santosh K. Gupta, Ashu Jain, Varun Aggarwal, Jürgen Branke, Sushil J. Louis, and Kay Chen Tan, editors, *Simulated Evolution and Learning, 8th International Conference, SEAL 2010*, pages 455–465, Kanpur, India, December 1-4 2010. Springer. Lecture Notes in Computer Science Vol. 6457.
- [163] Dhish Kumar Saxena, Jo ao A. Duro, Ashutosh Tiwari, Kalyanmoy Deb, and Qingfu Zhang. Objective Reduction in Many-Objective Optimization: Linear and Nonlinear Algorithms. *IEEE Transactions on Evolutionary Computation*, 17(1):77–99, February 2013.

- [164] Henry Scheffe. Experiments with Mixtures. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, pages 344–360, 1958.
- [165] Henry Scheffe. The Simplex-Centroid Design for Experiments with Mixtures. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, pages 235–263, 1963.
- [166] Jason R. Schott. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. Master’s thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts, May 1995.
- [167] Richard J. Brand Scott D. Chasalow. Algorithm as 299: Generation of simplex lattice points. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 44(4):534–545, 1995.
- [168] Pratyush Sen and Jian-Bo Yang. *Multiple criteria decision support in engineering design*. Springer Science & Business Media, 2012.
- [169] Alena Shmygelska and Holger H. Hoos. An ant colony optimisation algorithm for the 2d and 3d hydrophobic polar protein folding problem. *BMC Bioinformatics*, 6(1):1–22, 2005.
- [170] Krzysztof Socha and Christian Blum. An ant colony optimization algorithm for continuous optimization: application to feed-forward neural network training. *Neural Computing and Applications*, 16(3):235–247, 2007.
- [171] Krzysztof Socha and Marco Dorigo. Ant colony optimization for continuous domains. *European Journal of Operational Research*, 185:1155–1173, 2008.
- [172] Krzysztof Socha, Joshua Knowles, and Michael Sampels. A MAX-MIN Ant System for the University Course Timetabling Problem. In Michael Sampels Marco Dorigo, Gianni Di Caro, editor, *Ant Algorithms: Third International Workshop, ANTS 2002*, volume 2463 2002 of *Lecture Notes in Computer Science*, pages 1–13, Brussels, Belgium, September 2002. Springer Berlin Heidelberg.
- [173] Krzysztof Socha, Michael Sampels, and Max Manfrin. Ant Algorithms for the University Course Timetabling Problem with Regard to the State-of-the-Art. In Stefano Cagnoni, Colin G. Johnson, Juan J. Romero Cardalda, Elena Marchiori, David W. Corne, Jean-Arcady Meyer, Jens Gottlieb, Martin Middendorf, Agnès Guillot, Günther R. Raidl, and Emma Hart, editors, *Applications of Evolutionary Computing: EvoWorkshops 2003: EvoBIO, EvoCOP, EvoIASP, EvoMUSART, EvoROB, and EvoSTIM Essex, UK, April 14–16, 2003 Proceedings*, pages 334–345, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.
- [174] Christine Solnon. Combining two pheromone structures for solving the car sequencing problem with ant colony optimization. *European Journal of Operational Research*, 191(3):1043–1055, 2008.

-
- [175] Christine Solnon and Serge Fenet. A study of aco capabilities for solving the maximum clique problem. *Journal of Heuristics*, 12(3):155–180, 2006.
- [176] R.R. Stoll. *Set Theory and Logic*. Dover books on advanced mathematics. Dover Publications, 1979.
- [177] T. Stutzle and H. Hoos. MAX-MIN Ant System and local search for the traveling salesman problem. In *IEEE International Conference on Evolutionary Computation (CEC'1997)*, pages 309–314, April 1997.
- [178] Thomas Stützle et al. An ant approach to the flow shop problem. In *Proceedings of the 6th European Congress on Intelligent Techniques & Soft Computing (EUFIT 98)*, volume 3, pages 1560–1564, 1998.
- [179] Thomas Stützle and Holger H. Hoos. MAX-MIN Ant System. *Future Generation Computer Systems*, 16(9):889–914, June 2000.
- [180] Twelfth Intern. Conf. on Machine Learning. *Ant-Q: A Reinforcement Learning approach to the traveling salesman problem*. Morgan Kaufmann Publishers, 1995.
- [181] Dŭng H. Phan and Junichi Suzuki. R2-IBEA: R2 Indicator Based Evolutionary Algorithm for Multiobjective Optimization. In *2013 IEEE Congress on Evolutionary Computation (CEC'2013)*, pages 1836–1845, Cancún, México, 20-23 June 2013. IEEE Press. ISBN 978-1-4799-0454-9.
- [182] David A. Van Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio, May 1999.
- [183] John Von Neumann and Oskar Morgenstern. *Theory of games and economic behavior*. Princeton university press, New Jersey, 1944.
- [184] Tobias Wagner, Nicola Beume, and Boris Naujoks. Pareto-, Aggregation-, and Indicator-Based Methods in Many-Objective Optimization. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007*, pages 742–756, Matshushima, Japan, March 2007. Springer. Lecture Notes in Computer Science Vol. 4403.
- [185] Ronald E Walpole, Raymond H Myers, and Sharon L Myers. *Probabilidad y estadística para ingenieros*. Pearson Educación, 1999.
- [186] Lyndon While, Lucas Bradstreet, and Luigi Barone. A Fast Way of Calculating Exact Hypervolumes. *IEEE Transactions on Evolutionary Computation*, 16(1):86–95, February 2012.

- [187] Andrzej P. Wierzbicki. Basic Properties of Scalarizing Functionals for Multiobjective Optimization. *Mathematische Operationsforschung und Statistik, Ser. Optimization*, 8(1):55–60, 1977.
- [188] Andrzej P. Wierzbicki. A methodological guide to multiobjective optimization. In K. Iracki, K. Malanowski, and S. Walukiewicz, editors, *Optimization Techniques: Proceedings of the 9th IFIP Conference on Optimization Techniques*, pages 99–123, Berlin, Heidelberg, 1980. Springer Berlin Heidelberg.
- [189] Andrzej P. Wierzbicki. The use of reference objectives in multiobjective optimization. In Günter Fandel and Tomas Gal, editors, *Multiple Criteria Decision Making Theory and Application: Proceedings of the Third Conference Hagen/Königswinter*, pages 468–486, Berlin, Heidelberg, 1980. Springer Berlin Heidelberg.
- [190] Tomohiro Yoshikawa, Daisuke Yamashiro, and Takeshi Furuhashi. A Proposal of Visualization of Multi-Objective Pareto Solutions-Development of Mining Technique for Solutions-. In *Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Multicriteria Decision Making (MCDM'2007)*, pages 172–177, Honolulu, Hawaii, USA, April 2007. IEEE Press.
- [191] Qingfu Zhang and Hui Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, December 2007.
- [192] Eckart Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, November 1999.
- [193] Eckart Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, November 1999.
- [194] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. Technical Report 70, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, December 1999.
- [195] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, Summer 2000.
- [196] Eckart Zitzler and Simon Künzli. Indicator-based Selection in Multiobjective Search. In Xin Yao et al., editor, *Parallel Problem Solving from Nature - PPSN VIII*, pages 832–842, Birmingham, UK, September 2004. Springer-Verlag. Lecture Notes in Computer Science Vol. 3242.

- [197] Eckart Zitzler, Marco Laumanns, and Stefan Bleuler. A Tutorial on Evolutionary Multiobjective Optimization. In Xavier Gandibleux, Marc Sevaux, Kenneth Sörensen, and Vincent T'kindt, editors, *Metaheuristics for Multiobjective Optimisation*, pages 3–37. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.
- [198] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty, editors, *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, pages 95–100, Athens, Greece, 2002.
- [199] Eckart Zitzler and Lothar Thiele. Multiobjective Optimization Using Evolutionary Algorithms—A Comparative Study. In A. E. Eiben, editor, *Parallel Problem Solving from Nature V*, pages 292–301, Amsterdam, September 1998. Springer-Verlag.
- [200] Eckart Zitzler and Lothar Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, November 1999.