



UNIVERSIDAD DE GUANAJUATO

CAMPUS IRAPUATO-SALAMANCA

DIVISIÓN DE INGENIERÍAS

**UNA VARIANTE DE ALGORITMO BASADO
EN CÚMULOS DE PARTÍCULAS PARA
OPTIMIZACIÓN MULTI-OBJETIVO**

TESIS PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN MECATRÓNICA

PRESENTA:

JESÚS ALFREDO RODRÍGUEZ HERNÁNDEZ

DIRECTORES:

DR. CARLOS ARTEMIO COELLO COELLO

DR. VÍCTOR AYALA RAMÍREZ

SALAMANCA, GUANAJUATO, MÉXICO. DICIEMBRE 2017

Resumen

Existen muchos problemas del mundo real cuya solución requiere optimizar simultáneamente varios objetivos que están en conflicto uno con respecto al otro; es decir, el mejoramiento de un objetivo implica el deterioro de otro. Estos son llamados Problemas Multi-Objetivo (POMs). Los Algoritmos Evolutivos Multi-Objetivo (AEMOs) han sido muy exitosos cuando se resuelven Problemas Multi-Objetivo, principalmente debido a que no requieren ningún conocimiento específico acerca del problema. Esto quiere decir que pueden ser aplicados a varios problemas del mundo real con poca o ninguna modificación. El uso de Algoritmos Evolutivos Multi-Objetivo es particularmente adecuado para resolver Problemas Multi-Objetivo ya que trabajan simultáneamente con un conjunto de soluciones potenciales (la población). Esta característica les permite encontrar varias soluciones del Conjunto de Óptimos de Pareto en cada ejecución.

La optimización mediante cúmulos de partículas (PSO por sus siglas en inglés) es una técnica de búsqueda heurística que simula el movimiento del vuelo de un grupo de aves cuyo objetivo es encontrar comida. Una característica bien conocida de los algoritmos basados en PSO es su rápida convergencia. Además, es una técnica poblacional lo cual permite que sea un candidato natural para ser extendido a Problemas de optimización Multi-objetivo. El *Speed-constrained Multi-objective Particle Swarm Optimizer (SMPSO)* es uno de los algoritmos multi-objetivo basados en cúmulo de partículas más destacados en la literatura. *SMPSO* hace uso de un mecanismo para limitar la velocidad de las partículas cuando ésta se vuelve muy grande, lo que permite una búsqueda más eficaz en el espacio de las variables. *SMPSO_{hv}* es una versión del *SMPSO* que adopta al indicador de hipervolumen como mecanismo de selección, la cual muestra un mejor desempeño que *SMPSO* y es capaz de escalar en el número de funciones objetivo. Sin embargo, el costo computacional que implica el cálculo del hipervolumen es exponencial con el número de objetivos lo cual limita seriamente su uso. Debido a esto, en esta tesis se propone un nuevo mecanismo de selección basado en el hipervolumen que no realiza el cálculo exacto de la contribución al hipervolumen,

sino que calcula una aproximación que ayuda a reducir el costo computacional. El mecanismo de selección propuesto es integrado a *SMPSO* y el algoritmo resultante es llamado *SMPSO_{FHV}*. Este algoritmo es competitivo en cuanto a los tiempos de ejecución al resolver los conjuntos de problemas de prueba Zitzler-Deb-Thiele (ZDT) y Deb-Thiele-Laumanns-Zitzler (DTLZ, en su versión de 3 objetivos) con respecto a los algoritmos del estado del arte utilizados en este trabajo (*SMPSO*, *NSGA-II* y *MOEA/D*). Los resultados de este trabajo muestran que nuestra propuesta produce mejores resultados con respecto a tres indicadores del estado del arte (*HV*, *IGD+* y *Espaciado*). Adicionalmente, el algoritmo propuesto puede resolver problemas de hasta cinco funciones objetivo con un tiempo de ejecución menor a dos minutos. Éste es un tiempo considerablemente bueno, ya que si se utilizara el cálculo exacto de la contribución al hipervolumen en el mecanismo de selección, el tiempo de ejecución sería de horas.

Abstract

There are many real world problems that require optimizing several objectives concurrently, furthermore, these objectives are in conflict, in other words, the improvement of one objective implies the deterioration of another in the problem. These type of problems are called multi-objective optimization problems (MOPs). The multi-objective evolutionary algorithms (MOEAs) have been very successful when solving a MOP, mainly since they need no previous knowledge about the problem. Therefore, MOEAs can be applied to solve many real world problems with a few or without modifications. The use of MOEAs is particularly adequate for solving MOPs due to the fact that the solution of a MOP represents a complete set of points which can be represented by the population. Then, a MOEA is able to find an approximation of the solution set (Pareto set and Pareto front) in every single execution.

The particle swarm optimization (PSO) is a heuristic search technique that simulates the movement of a swarm finding food. One of the remarkable features of the algorithms based on PSO is that the fast rate of convergence. Furthermore, since it is a set-based technique is then, a natural candidate for being adapted in order to solve MOPs. The *Speed-constrained Multi-objective Particle Swarm Optimizer (SMPSO)* is one of the most widely used techniques within literature. The *SMPSO* has a specialized mechanism to control the velocity of the particles when it becomes too high, this feature allows performing an effective search over the decision variable space. The *SMPSO_{hv}* is a modified version of *SMPSO* that adapts the hypervolume indicator as a selection mechanism. The modification allows *SMPSO_{hv}* to have a better performance against the original version, and also the new version is able to scale the number of objectives. However, the computational cost that implies the calculation of the hypervolume becomes exponential as the number of objectives increase which limits the use of this indicator. Due to the previous fact, in this thesis work, we propose a new selection mechanism that does not require the exact calculation of the hypervolume contribution, but it calculates an approximation in order to reduce the computational cost. The proposed selection mechanism, that is integrated

into the *SMPSO*, is called *SMPSO_{FHV}*. The *SMPSO_{FHV}* is competitive with the execution time when solving the test benchmark problems, included in Zitzler-Deb-Thiele (ZDT) and Deb-Thiele-Laumanns-Zitzler (DTLZ) (3 objectives), against the following state of the art algorithms: *SMPSO*, *NSGA-II* y *MOEA/D*. Numerical results show that *SMPSO_{FHV}* obtain better results with respect to three performance indicators (*HV*, *IGD+* and *Spacing*). Finally, the proposed algorithm is able to obtain a reliable approximation of the solution set of problems that have up to five objective functions in less than two minutes. The execution time is acceptable since if the algorithm computes the exact hypervolume contribution into the selection mechanism, the execution time grows from minutes to hours.

Agradecimientos

En primer lugar, quiero agradecer a mis padres Alfredo Rodríguez Hernández y Ma. Beatriz Hernández Flores, así como a mis hermanos Ivan Abdiel Rodríguez Hernández y Luis Alberto Rodríguez Hernández por su apoyo incondicional a lo largo de todo el proceso de mi carrera universitaria. Ellos han sido un gran soporte para alcanzar cada uno de mis logros. También, agradezco el apoyo de mis abuelos, tíos y primos.

Quiero agradecer de forma especial al Dr. Carlos Artemio Coello Coello y al Dr. Víctor Ayala Ramírez por su guía y apoyo durante el desarrollo de mi tesis. Además, agradezco cada uno de sus consejos que para mí son invaluable. Sin duda, ellos son un gran ejemplo de inspiración en mi vida profesional por sus grandes logros. Gracias a ellos, tuve la oportunidad de estar en tan honorable institución como lo es el Centro de Investigación y de Estudios Avanzados del IPN (CINVESTAV-IPN) al momento de realizar mi tesis. Particularmente en el Departamento de Computación, conocí a los estudiantes de maestría y doctorado quienes jamás dudaron en ayudarme y les estoy completamente agradecido.

A mis revisores de tesis el Dr. Mario Alberto Ibarra Manzano y el Dr. Raúl Enrique Sánchez Yáñez, por su tiempo y valiosas aportaciones para la obtención de un mejor trabajo.

A el Dr. René de Jesús Romero Troncoso, porque es una fuente de inspiración para mí desde mi llegada a la Universidad de Guanajuato y considero que es el mejor profesor en esta institución. A el Dr. Horacio Rostro González y el Dr. Roberto Rojas Laguna por sus consejos y apoyo durante mi carrera universitaria.

A mis amigos de la Universidad de Guanajuato por hacer de mi experiencia universitaria la mejor.

Por último, quiero agradecer enormemente a la Universidad de Guanajuato por todo el conocimiento que implantó en mí y por todas las habilidades que me ayudó a desarrollar. Prometo devolverle lo mucho que me ha otorgado.

VIII

¡Soy orgullosamente UG!

Índice general

| | |
|---|-----------|
| Índice de Figuras | xI |
| Índice de Tablas | xv |
| Índice de Algoritmos | xvii |
| Lista de Acrónimos | xx |
| 1. Introducción | 1 |
| 1.1. Objetivo de la tesis | 2 |
| 1.2. Contribuciones | 2 |
| 1.3. Organización de la tesis | 3 |
| 2. Conceptos de optimización multi-objetivo | 5 |
| 2.1. El Problema de Optimización Multi-Objetivo | 5 |
| 2.2. Dominancia y Optimalidad de Pareto | 6 |
| 2.3. Vectores de referencia | 8 |
| 2.3.1. Vector objetivo ideal | 8 |
| 2.3.2. Vector objetivo utópico | 9 |
| 2.3.3. Vector objetivo de Nadir | 10 |
| 3. Optimización mediante cúmulos de partículas | 11 |

| | | |
|-----------|---|-----------|
| 3.1. | Algoritmo PSO mono-objetivo | 11 |
| 3.2. | Algoritmo PSO multi-objetivo (MOPSO) | 13 |
| 4. | Metaheurística $SMPSO_{FHV}$ | 17 |
| 4.1. | Algoritmo $SMPSO$ | 17 |
| 4.2. | Algoritmos Evolutivos Multi-objetivo basados en Indicadores de Desempeño | 20 |
| 4.2.1. | Mecanismos de selección basados en Hipervolumen | 22 |
| 4.3. | Propuesta para el cálculo de la contribución al Hipervolumen | 22 |
| 4.3.1. | Nuevo mecanismo de selección basado en hipervolumen | 27 |
| 4.4. | Funcionamiento de los mecanismos de selección | 28 |
| 4.4.1. | Mecanismo de selección propuesto en [7] | 28 |
| 4.4.2. | Mecanismo de selección propuesto en [8] | 29 |
| 4.4.3. | Nuevo método de selección propuesto | 30 |
| 4.5. | Nuestro algoritmo $SMPSO_{FHV}$ | 32 |
| 5. | Estudio Experimental | 35 |
| 5.1. | Problemas de prueba | 36 |
| 5.2. | Indicadores de evaluación de desempeño | 36 |
| 5.3. | Experimento 1: Comparación de desempeño en problemas ZDT1-4, ZDT6 y DTLZ1-7 (3 objetivos) | 39 |
| 5.3.1. | Configuración de parámetros de los algoritmos | 39 |
| 5.3.2. | Parámetros de evaluación | 41 |
| 5.3.3. | Discusión de resultados | 41 |
| 5.4. | Experimento 2: Prueba de escabilidad utilizando el problema DTLZ2 (2-5 objetivos) | 45 |
| 5.4.1. | Configuración de parámetros de los algoritmos | 45 |
| 5.4.2. | Parámetros de evaluación | 45 |

| | |
|---|-----------|
| <i>ÍNDICE GENERAL</i> | XI |
| 5.4.3. Discusión de resultados | 46 |
| 6. Resultados y trabajo futuro | 49 |
| A. Problemas de Prueba | 51 |
| A.1. Conjunto de problemas Zitzler-Deb-Thiele | 51 |
| A.2. Conjunto de problemas Deb-Thiele-Laumanns-Zitzler | 57 |
| B. Resultados Numéricos | 65 |
| B.1. Tablas de comparación de los AEMOs (experimento 1) | 65 |
| B.2. Tablas de comparación de los AEMOs (experimento 2) | 68 |
| B.3. Gráficas de aproximación de los AEMOs (2D y 3D) | 69 |
| Bibliografía | 69 |

Índice de figuras

| | | |
|------|---|----|
| 2.1. | Ejemplo de dominancia de Pareto para dos funciones objetivo. | 8 |
| 2.2. | Vectores de referencia: vector ideal (\vec{z}^*), vector de Nadir (\vec{z}^{nad}) y vector utópico (\vec{z}^{**}). | 9 |
| 4.1. | Si $\mathbf{A} = \{\vec{z}_1, \vec{z}_2, \dots, \vec{z}_6\}$ es la aproximación del frente óptimo de Pareto y \vec{y}_{ref} es el punto de referencia. El área gris sin rayas representa el hipervolumen del conjunto \mathbf{A} y el área gris con rayas representa la contribución al hipervolumen de la solución \vec{z}_2 | 21 |
| 4.2. | Si $\mathbf{A} = \{\vec{z}_1, \vec{z}_2, \dots, \vec{z}_6\}$ es el frente de Pareto aproximado y movemos \vec{z}_2 entre \vec{z}_1 y \vec{z}_3 , el espacio cubierto por $\{\mathbf{A} - \vec{z}_2\}$ no es afectado y sólo la contribución al hipervolumen de \vec{z}_2 es afectada. | 23 |
| 4.3. | Si $\mathbf{A} = \{\vec{z}_1, \vec{z}_2, \dots, \vec{z}_6\}$ es el frente de Pareto aproximado, tenemos dos regiones vistas por \vec{z}_2 , donde se encuentran todos los demás elementos del frente. En cada una de las regiones se toma el elemento más cercano a \vec{z}_2 (Ec. 4.10). Los elementos más cercanos en cada región están representados por un triángulo. | 26 |
| 4.4. | $\mathbf{C} = \{\vec{z}_1, \vec{z}_2, \vec{z}_3\}$ es el frente de Pareto entregado por nuestro método para calcular la contribución al hipervolumen de \vec{z}_2 | 27 |
| 4.5. | Contribución al hipervolumen de todas las soluciones. | 29 |
| 4.6. | Contribución al hipervolumen de B , New y D | 29 |
| 4.7. | Contribución al hipervolumen de todas las soluciones. | 30 |
| 4.8. | Contribución al hipervolumen de B , New y C | 31 |

| | |
|---|----|
| 5.1. Clasificación de desempeño de las metaheurísticas probadas en el conjunto ZDT y DTLZ (3 objetivos) con base en el indicador <i>Espaciado</i> . Por cada algoritmo, se muestra el número de POMs en los que obtiene una clasificación dada. | 42 |
| 5.2. Clasificación de desempeño de las metaheurísticas probadas en el conjunto ZDT y DTLZ (3 objetivos) con base en el indicador IGD+. Por cada algoritmo, se muestra el número de POMs en los que obtiene una clasificación dada. | 43 |
| 5.3. Clasificación de desempeño de las metaheurísticas probadas en el conjunto ZDT y DTLZ (3 objetivos) con base en el indicador Hipervolumen. Por cada algoritmo, se muestra el número de POMs en los que obtiene una clasificación dada. | 44 |
| 5.4. Valores del indicador Hipervolumen para el problema DTLZ2 cuando el número de objetivos es variado de 2 a 5. | 47 |
| 5.5. Tiempo de ejecución para el problema DTLZ2 cuando el número de objetivos es variado de 2 a 5. | 47 |
| A.1. Frente óptimo de Pareto para el problema ZDT1. | 52 |
| A.2. Frente óptimo de Pareto para el problema ZDT2. | 53 |
| A.3. Frente óptimo de Pareto para el problema ZDT3. | 54 |
| A.4. Frente óptimo de Pareto para el problema ZDT4. | 55 |
| A.5. Frente óptimo de Pareto para el problema ZDT6. | 56 |
| A.6. Frente óptimo de Pareto para el problema DTLZ1 con tres objetivos. | 58 |
| A.7. Frente óptimo de Pareto para el problema DTLZ2 con tres objetivos. | 59 |
| A.8. Frente óptimo de Pareto para el problema DTLZ3 con tres objetivos. | 60 |
| A.9. Frente óptimo de Pareto para el problema DTLZ4 con tres objetivos. | 61 |
| A.10. Frente óptimo de Pareto para el problema DTLZ5 con tres objetivos. | 62 |
| A.11. Frente óptimo de Pareto para el problema DTLZ6 con tres objetivos. | 63 |
| A.12. Frente óptimo de Pareto para el problema DTLZ7 con tres objetivos. | 64 |

| | |
|--|----|
| B.1. Problema de prueba ZDT1. | 69 |
| B.2. Problema de prueba ZDT2. | 70 |
| B.3. Problema de prueba ZDT3. | 71 |
| B.4. Problema de prueba ZDT4. | 72 |
| B.5. Problema de prueba ZDT6. | 73 |
| B.6. Problema de prueba DTLZ1 con tres objetivos. | 74 |
| B.7. Problema de prueba DTLZ2 con tres objetivos. | 75 |
| B.8. Problema de prueba DTLZ3 con tres objetivos. | 76 |
| B.9. Problema de prueba DTLZ4 con tres objetivos. | 77 |
| B.10. Problema de prueba DTLZ5 con tres objetivos. | 78 |
| B.11. Problema de prueba DTLZ6 con tres objetivos. | 79 |
| B.12. Problema de prueba DTLZ7 con tres objetivos. | 80 |

Índice de tablas

| | |
|--|----|
| 5.1. Puntos de referencia para el Experimento 1. | 37 |
| 5.2. Puntos de referencia para el Experimento 2. | 37 |
| 5.3. Parámetros comunes entre las metaheurísticas en el experimento 1. | 41 |
| 5.4. Parámetros comunes entre las metaheurísticas en el experimento 2. | 45 |
| B.1. Comparación de resultados obtenidos del indicador HV por $MOEA/D$, $NSGA-II$, $SMPSO$ y nuestro algoritmo $SMPSO_{FHV}$ para el conjunto de problemas de prueba ZDT y DTLZ (3 objetivos). | 66 |
| B.2. Comparación de resultados obtenidos del indicador $IGD+$ por $MOEA/D$, $NSGA-II$, $SMPSO$ y nuestro algoritmo $SMPSO_{FHV}$ para el conjunto de problemas de prueba ZDT y DTLZ (3 objetivos). | 66 |
| B.3. Comparación de resultados obtenidos del indicador $Espaciado$ por $MOEA/D$, $NSGA-II$, $SMPSO$ y nuestro algoritmo $SMPSO_{FHV}$ para el conjunto de problemas de prueba ZDT y DTLZ (3 objetivos). | 67 |
| B.4. Media del tiempo de ejecución necesario por $MOEA/D$, $NSGA-II$, $SMPSO$ y nuestro algoritmo $SMPSO_{FHV}$ para resolver cada problema del conjunto ZDT y DTLZ (3 objetivos). | 67 |
| B.5. Comparación de resultados obtenidos del indicador HV por $MOEA/D$, $NSGA-II$, $SMPSO$ y nuestro algoritmo $SMPSO_{FHV}$ para el problema DTLZ2 usando de 2 a 5 objetivos. El tiempo de ejecución promedio también es reportado. | 68 |

Índice de Algoritmos

| | | |
|----|---|----|
| 1. | Pseudocódigo de PSO | 12 |
| 2. | Pseudocódigo de MOPSO | 14 |
| 3. | Pseudocódigo de SMPSO | 19 |
| 4. | Pseudocódigo de nuestro $SMPSO_{FHV}$ | 33 |

Lista de Acrónimos

POMs Problemas de Optimización Multi-objetivo

AEMOs Algoritmos Evolutivos Multi-objetivo

PSO Optimizadores Mediante Cúmulos de Partículas

MOPSOs Algoritmos Multi-Objetivo basado en Optimización mediante Cúmulos de Partículas

1 — Introducción

Existen muchos problemas de la industria y la ingeniería cuya solución requiere optimizar simultáneamente varios objetivos que están en conflicto uno con respecto al otro, es decir, el mejoramiento de un objetivo implica el deterioro de otro en el problema. A éstos se les conoce como Problemas Multi-Objetivo (POMs). Los Algoritmos Evolutivos Multi-Objetivo (AEMOs) han sido muy exitosos cuando se resuelven Problemas Multi-Objetivos, principalmente debido a que presentan las siguientes ventajas:

- No requieren ningún conocimiento específico acerca del problema.
- Pueden ser utilizados como optimizadores efectivos, robustos y globales.
- Son fáciles de entender e implementar ya sea en plataformas secuenciales o paralelas.
- Pueden ser hibridizados con técnicas de programación matemática e incluso con otras metaheurísticas.
- Son menos susceptibles de ser afectados por la forma y continuidad del Frente de Pareto en comparación con los métodos de programación matemática.

El uso de Algoritmos Evolutivos Multi-Objetivo es particularmente adecuado para resolver Problemas Multi-Objetivo ya que trabajan simultáneamente con un conjunto de soluciones potenciales (la población). Esta característica les permite encontrar varias soluciones del Conjunto de Óptimos de Pareto en cada ejecución.

Particularmente, los Optimizadores Mediante Cúmulos de Partículas (PSO) han sido ampliamente utilizados en la última década debido a su simplicidad y buen

desempeño. El éxito de este algoritmo como un optimizador mono-objetivo (principalmente cuando se tienen espacios de búsqueda continuos) ha motivado a los investigadores a extender el uso de esta técnica bio-inspirada a otras áreas. Una de ellas es la Optimización Multi-Objetivo. Un número considerable de Optimizadores Multi-Objetivo basados en Cúmulos de Partículas han sido propuestos, pero existen aun muchos aspectos que deben ser mejorados.

Estamos seguros de que un buen MOPSO sería útil no sólo para resolver problemas de prueba estándar encontrados en la literatura, sino también para aplicaciones del mundo real.

1.1. Objetivo de la tesis

Diseñar una variante del algoritmo Multi-Objetivo basado en Optimización mediante Cúmulo de Partículas (MOPSO) que sea mejor o al menos competitivo comparado con los algoritmos existentes en el estado del arte, considerando para su comparación un conjunto de problemas de prueba estándar e indicadores reportados en la literatura especializada.

1.2. Contribuciones

Las principales contribuciones de esta tesis se resumen a continuación:

- Un nuevo mecanismo de selección basado en el indicador de desempeño conocido como hipervolumen que muestra ser rápido sin perder calidad significativa en la distribución y convergencia de las soluciones.
- Un nuevo Algoritmo Multi-Objetivo basado en Optimización mediante Cúmulo de Partículas que utiliza el mecanismo de selección basado en hipervolumen indicado en el punto anterior, el cual es llamado $SMPSO_{FHV}$.
- Un estudio comparativo detallado de nuestra propuesta con respecto a otros AEMOs del estado del arte para espacios continuos en baja y alta dimensionalidad.

1.3. Organización de la tesis

Además de este capítulo introductorio, esta tesis contiene seis capítulos y dos apéndices.

El capítulo 2 contiene una serie de conceptos y definiciones que son necesarios para un mejor entendimiento de la optimización multi-objetivo.

El capítulo 3 introduce brevemente la optimización mediante cúmulos de partículas tanto mono-objetivo como multi-objetivo, así como las principales características a ser consideradas en estos dos tipos de problemas.

El capítulo 4 muestra nuestro mecanismo de selección basado en hipervolumen, así como el algoritmo propuesto.

El capítulo 5 muestra los resultados obtenidos al comparar nuestra propuesta con respecto a tres AEMOs del estado del arte para espacios continuos. Se emplean problemas de prueba estándar e indicadores de la literatura especializada para evaluar el desempeño de cada AEMO. Nuestro algoritmo es puesto a prueba tanto en baja como en alta dimensionalidad en el espacio de los objetivos. Además, se muestra una discusión de los resultados obtenidos.

Las conclusiones obtenidas, así como también las posibles líneas de trabajo futuro son descritas en el capítulo 6. El apéndice A contiene una descripción completa de los problemas de prueba estándar empleados por las metaheurísticas adoptadas en este trabajo de tesis.

Por último, el apéndice B contiene los resultados numéricos obtenidos en los experimentos, los cuales son organizados en tablas y figuras.

2 — Conceptos de optimización multi-objetivo

En este capítulo se definen las características de un problema de optimización multi-objetivo, así como los conceptos de dominancia y optimalidad de Pareto.

2.1. El Problema de Optimización Multi-Objetivo

La Optimización Multi-Objetivo se refiere a optimizar simultáneamente un conjunto de dos o más funciones objetivo. Dichos objetivos normalmente se encuentran en conflicto unos con otros, por lo que mejorar uno implica empeorar el desempeño de los demás.

Considerando que se minimizarán todos los objetivos (en general, cualquier problema de maximización puede transformarse fácilmente en un problema de minimización), las siguientes expresiones definen matemáticamente a un Problema de Optimización Multi-Objetivo.

Definición 1. Problema de Optimización Multi-Objetivo:

Encontrar un vector \vec{x}^* que satisfaga las m restricciones de desigualdad:

$$g_i(\vec{x}^*) \leq 0; i = \{1, \dots, m\} \quad (2.1)$$

las n restricciones de igualdad:

$$h_i(\vec{x}^*) = 0; i = \{1, \dots, n\} \quad (2.2)$$

y optimice el vector de funciones objetivo:

$$f(\vec{x}^*) = [f_1(\vec{x}^*), f_2(\vec{x}^*), \dots, f_k(\vec{x}^*)] \quad (2.3)$$

Las restricciones son las que definen la región factible del problema y cualquier vector \vec{x}^* que se encuentre en dicha región se considera como una solución factible.

Optimización Mono-Objetivo y Multi-Objetivo

En Optimización, cuando tratamos con Problemas Mono-Objetivo sólo se busca una solución óptima, mientras que en Problemas Multi-Objetivo se busca un conjunto de soluciones que representen los mejores compromisos posibles entre los objetivos, ya que normalmente no existe solución única. Se hace notar que en Optimización Multi-Objetivo se tienen claramente dos tareas: progresar hacia el frente óptimo de Pareto y mantener diversidad en el conjunto de soluciones (o sea, obtener una distribución uniforme a lo largo del frente).

Cabe mencionar que el realizar un proceso adecuado de optimización no implica que las soluciones obtenidas tendrán una distribución uniforme y vice versa. Por tanto, mecanismos explícitos o implícitos deben ser introducidos para enfatizar convergencia hacia el Frente Óptimo de Pareto y para mantener diversidad dentro del conjunto de soluciones producidas.

2.2. Dominancia y Optimalidad de Pareto

La mayoría de algoritmos de optimización Multi-Objetivo utilizan el concepto de dominancia de Pareto en su búsqueda. En estos algoritmos, dos soluciones son comparadas en base a si una domina o no domina a la otra. Para describir el concepto de optimalidad, introduciremos las siguientes definiciones.

Definición 2. (Dominancia de Pareto) Si se tienen dos vectores $\vec{x}, \vec{y} \in R^k$, se dice que $\vec{x} \leq \vec{y}$ si $x_i \leq y_i$ para $i = \{1, \dots, k\}$ y que \vec{x} domina a \vec{y} ($\vec{x} \prec \vec{y}$) si $\vec{x} \leq \vec{y}$ y $\vec{x} \neq \vec{y}$.

Definición 3. (Dominancia débil de Pareto) Si se tienen dos vectores $\vec{x}, \vec{y} \in R^k$, y se cumple $\forall i, x_i \leq y_i$. Entonces se dice que \vec{x} domina débilmente a \vec{y} ($\vec{x} \preceq \vec{y}$).

Definición 4. Se dice que un vector de variables de decisión $\vec{x} \in \mathbf{X} \subset R^n$ es no-dominado con respecto a \mathbf{X} , si no existe otro $\vec{x}' \in \mathbf{X}$ tal que $f(\vec{x}') \prec f(\vec{x})$.

Definición 5. Se dice que un vector de variables de decisión $\vec{x}^* \in \mathcal{F} \subset R^n$ (\mathcal{F} es la región factible) es óptimo de Pareto si es no dominado con respecto a \mathcal{F} .

Definición 6. El conjunto de óptimos de Pareto \mathcal{P}^* es definido como:

$$\mathcal{P}^* = \{\vec{x} \in \mathcal{F} | \vec{x} \text{ es un óptimo de Pareto}\} \quad (2.4)$$

Definición 7. El frente de Pareto \mathcal{PF}^* es definido por:

$$\mathcal{PF}^* = \{f(\vec{x}) \in R^k | \vec{x} \in \mathcal{P}^*\} \quad (2.5)$$

Definición 8. (Dominancia de Pareto para conjuntos) Si se tienen dos conjuntos de vectores solución A y B , se dice que B es dominado por A ($A \prec B$) si se cumple que $\forall b_j \in B, \exists a_i \in A : a_i \prec b_j$.

Definición 9. (Dominancia débil de Pareto para conjuntos) Si se tienen dos conjuntos de vectores solución A y B , se dice que B es dominado débilmente por A ($A \preceq B$) si se cumple que $\forall b_j \in B, \exists a_i \in A : a_i \preceq b_j$.

Al final se desea determinar el Conjunto de Óptimos de Pareto del conjunto \mathcal{F} de todos los vectores de variables de decisión que satisfacen las definiciones 4 y 5.

En la figura 2.1 podemos ver gráficamente la dominancia de Pareto considerando un problema de minimización con dos objetivos. Por ejemplo, la solución A domina a B, ya que A mejora a la solución B tanto en la función objetivo f_1 como en la función objetivo f_2 . La solución C es no dominada con respecto a A, ya que C es mejor que la solución A en f_2 , pero A es mejor que C en f_1 . Las soluciones no dominadas forman el Frente de Pareto.

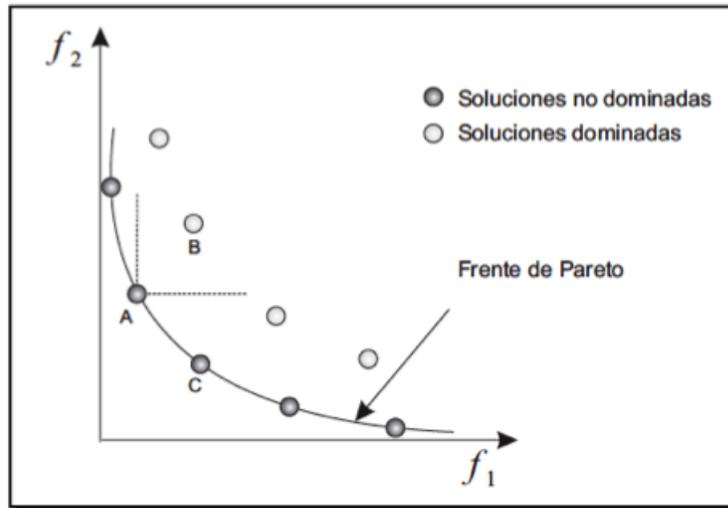


Figura 2.1: Ejemplo de dominancia de Pareto para dos funciones objetivo.

2.3. Vectores de referencia

En esta sección se presentan algunos vectores que son utilizados como soluciones de referencia (figura 2.2) y que pueden ayudar a ciertos métodos a resolver problemas multi-objetivo.

2.3.1. Vector objetivo ideal

El vector objetivo ideal denotado por \vec{z}^* , representa los límites inferiores de cada objetivo en el espacio de búsqueda factible. Está constituido por los valores óptimos individuales para cada función objetivo y se define como:

$$z_i^* = \min f_i(\vec{x}), \vec{x} \in \mathbf{X} \quad (2.6)$$

$$\vec{z}^* = (z_1^*, z_2^*, \dots, z_m^*)^T$$

donde $m =$ número de objetivos. En general, el vector objetivo ideal corresponde a una solución no factible en el espacio objetivo. La única forma en que un vector objetivo ideal sea factible es que no exista conflicto entre los objetivos. Usualmente es

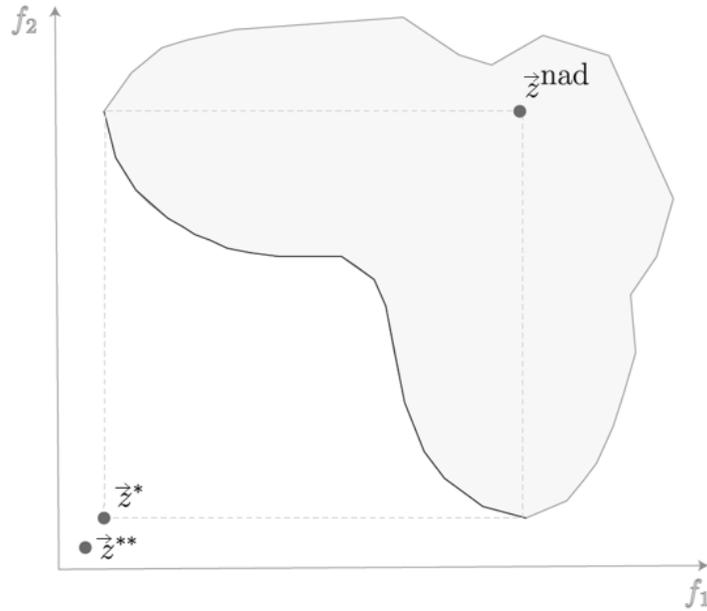


Figura 2.2: Vectores de referencia: vector ideal (\vec{z}^*), vector de Nadir (\vec{z}^{nad}) y vector utópico (\vec{z}^{**}).

utilizado como un punto de referencia, dado que las soluciones factibles más cercanas a éste representan el mejor compromiso entre los objetivos.

2.3.2. Vector objetivo utópico

Algunas veces se desea un vector objetivo que domine estrictamente a todas las soluciones del conjunto de óptimos de Pareto. A este vector se le denomina vector objetivo utópico, se le denota mediante $\vec{z}^{**} = (z_1^{**}, z_2^{**}, \dots, z_m^{**})^T$ y cada uno de sus componentes se define de la siguiente forma:

$$z_i^{**} = z_i^* - \epsilon_i, \forall i = \{1, \dots, m\}, \epsilon_i \geq 0 \quad (2.7)$$

2.3.3. Vector objetivo de Nadir

El vector objetivo de Nadir, denotado por \vec{z}^{nad} , representa los límites superiores de cada objetivo en el conjunto de óptimos de Pareto (\mathcal{P}^*) y se define como:

$$\begin{aligned} z_i^{nad} &= \max f_i(\vec{x}), \vec{x} \in \mathcal{P}^* \\ \vec{z}^{nad} &= (z_1^{nad}, z_2^{nad}, \dots, z_m^{nad})^T \end{aligned} \tag{2.8}$$

3 — Optimización mediante cúmulos de partículas

3.1. Algoritmo PSO mono-objetivo

James Kennedy y Russell C. Eberhart [1] propusieron el algoritmo de cúmulos de partículas (particle swarm optimization, o PSO) para optimización mono-objetivo. PSO es una técnica de búsqueda heurística que simula el movimiento del vuelo de un grupo de aves cuyo objetivo es encontrar comida. Aunque originalmente fue adoptado para balancear pesos en redes neuronales, el PSO pronto se convirtió en un optimizador global muy popular, principalmente en problemas en los cuales las variables de decisión son números reales.

El [algoritmo 1](#) muestra la forma en que trabaja el PSO general. Primeramente, la población es inicializada. Esta inicialización incluye tanto a las velocidades como a las posiciones bajo las siguientes fórmulas:

$$\vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t) \quad (3.1)$$

$$\vec{v}_i(t) = w \cdot \vec{v}_i(t-1) + C_1 \cdot r_1 \cdot (\vec{x}_{pbest_i} - \vec{x}_i) + C_2 \cdot r_2 \cdot (\vec{x}_{leader_i} - \vec{x}_i) \quad (3.2)$$

donde $\vec{v}_i(t)$ es la velocidad de la partícula i en la iteración t , $\vec{x}_i(t)$ es la posición de la partícula i en la iteración t , w es el factor de inercia de la velocidad, c_1 y c_2 son los factores cognitivos y social respectivamente (w , c_1 y c_2 generalmente son constantes), r_1 y r_2 son números aleatorios, $pbest$ y $leader$ representan la mejor

Algoritmo 1 Pseudocódigo de PSO

```
1: Inicializar la población
2: Elegir el líder
3: generación = 0
4: while generación <  $G_{max}$  do
5:   for cada partícula do
6:     Actualizar la posición // Ecuaciones (3.1) - (3.2)
7:     Evaluación
8:     Actualizar  $pbest$ 
9:   end for
10:  Actualizar el líder
11:  generación ++
12: end while
13: return Partícula líder
```

solución encontrada por cada partícula y la mejor solución encontrada por toda la población, respectivamente.

El $pbest$ correspondiente de cada partícula es inicializado y el líder es seleccionado (usualmente la solución $gbest$ es seleccionada como *leader*). Después, para un número máximo de iteraciones, cada partícula emprende su vuelo a través del espacio de búsqueda actualizando su posición (con las ecuaciones (3.1) y (3.2)), su $pbest$ y finalmente el líder es actualizado también.

La simplicidad relativa del PSO y el hecho de que es una técnica basada en el uso de una población han provocado que sea un candidato natural para ser extendido para Optimización Multi-Objetivo.

Moore y Chapman propusieron la primera extensión de la estrategia del PSO para resolver Problemas Multi-Objetivo en un manuscrito no publicado de 1999. Después de este intento temprano, un gran interés por extender el PSO surgió entre los investigadores, aunque, la siguiente propuesta fue publicada hasta el 2002. Sin embargo, actualmente existe un gran número de MOPSOs reportados en la literatura especializada.

3.2. Algoritmo PSO multi-objetivo (MOPSO)

Para poder aplicar la estrategia del PSO para resolver Problemas de Optimización Multi-Objetivo, es obvio que el esquema original debe ser modificado. Como se mencionó antes, el conjunto solución de un problema Multi-Objetivo consiste de varias soluciones no dominadas (el Conjunto de Óptimos de Pareto).

Dada la naturaleza poblacional del PSO, es deseable producir varias (diferentes) soluciones no dominadas con una sola ejecución. Así que, como con otros algoritmos evolutivos, las tres cuestiones que deben ser consideradas cuando se extiende PSO a Optimización Multi-Objetivo son:

1. ¿Cómo seleccionar a las partículas que se utilizarán como líderes para dar preferencia a las soluciones no dominadas sobre aquellas que son dominadas?
2. ¿Cómo conservar las soluciones no dominadas encontradas durante el proceso de búsqueda para contar con soluciones no dominadas con respecto a todas las poblaciones pasadas y no únicamente con respecto a la actual? También es deseable que estas soluciones estén bien distribuidas a lo largo del Frente de Pareto.
3. ¿Cómo mantener la diversidad en el cúmulo para evitar converger a sólo una solución?

Cuando se resuelven Problemas de Optimización Mono-Objetivo, el líder que cada partícula utiliza para actualizar su posición está completamente determinado una vez que la topología de vecindad está establecida (la forma en cómo se comunica el cúmulo). Sin embargo, en el caso de Problemas de Optimización Multi-Objetivo, cada partícula puede tener un conjunto de líderes diferentes de los cuales sólo uno puede ser elegido para actualizar su posición. Tal conjunto de líderes es usualmente almacenado en un lugar diferente a donde se encuentra el cúmulo (el denominado archivo externo): este archivo es un repositorio de tamaño definido inicialmente por el usuario (generalmente, es igual al tamaño de la población inicial) en el cual se almacenan las soluciones no dominadas encontradas hasta el momento. Después de esto, las soluciones no dominadas son filtradas por algún otro indicador para ajustar el número de soluciones no dominadas al tamaño máximo del archivo externo (en caso de ser necesario). Las soluciones contenidas en el archivo externo son usadas como líderes cuando las posiciones de las partículas del cúmulo necesitan ser actualizadas.

Además, el contenido del archivo externo es también reportado frecuentemente como la salida final del algoritmo.

Algoritmo 2 Pseudocódigo de MOPSO

```
1: Inicializar la población
2: Inicializar líderes en un archivo externo
3: Calidad (líderes)
4: generación = 0
5: while generación <  $G_{max}$  do
6:   for cada partícula do
7:     Seleccionar líder
8:     Actualizar la posición // Ecuaciones (3.1) - (3.2)
9:     Mutación
10:    Evaluación
11:    Actualizar pbest
12:   end for
13:   Actualizar el líderes en el archivo externo
14:   Calidad (líderes)
15:   generación ++
16: end while
17: return Archivo externo
```

El **algoritmo 2** muestra la forma en la cual trabaja un MOPSO general. Se marca con letras itálicas y subrayado a los procesos que hacen a este algoritmo diferente de un algoritmo PSO general para Optimización Mono-Objetivo.

Primero, el cúmulo es inicializado. Posteriormente, un conjunto de líderes es también inicializado con las partículas no dominadas del cúmulo. Como se mencionó antes, el conjunto de líderes es usualmente almacenado en un archivo externo. Después, alguna clase de medida de calidad es calculada para todos los líderes a fin de seleccionar (usualmente) un líder para cada partícula del cúmulo. En cada generación, para cada partícula, un líder es seleccionado y el vuelo es llevado a cabo. La mayoría de los MOPSOs existentes aplican algún tipo de operador de mutación después de haber realizado el vuelo. Posteriormente, la partícula es evaluada y su *pbest* correspondiente es actualizado (la forma de actualizar el *pbest* es un aspecto que debe ser

estudiado a fondo ya que afecta considerablemente el desempeño del algoritmo). Una forma sencilla de actualizarlo es: una nueva partícula reemplaza su partícula *pbest* cuando esta última es dominada o si ambas son incomparables (por ejemplo cuando son no dominadas una con respecto de la otra). Después de que todas las partículas han sido actualizadas, el conjunto de líderes es actualizado también. Finalmente, la medida de calidad del conjunto de líderes es calculada nuevamente. Este proceso es repetido por un cierto número de iteraciones (generalmente fijo).

Una vez mencionadas las características del algoritmo PSO, las cuestiones que surgen cuando tratamos con Problemas Multi-Objetivo están relacionadas principalmente con dos aspectos de diseño algorítmico:

1. Selección y actualización de líderes:

- ¿Cómo seleccionar un líder del conjunto de soluciones no dominadas tomando en cuenta que tales soluciones son igualmente buenas? ¿Deberíamos seleccionar este líder de manera aleatoria o debería utilizarse un criterio adicional (para promover diversidad, por ejemplo)?
- ¿Cómo seleccionar las partículas que deberían permanecer en el archivo externo de una iteración a otra?

2. Creación de nuevas soluciones:

- ¿Cómo promover diversidad a través de los dos mecanismos principales para crear nuevas soluciones: actualización de posiciones y el uso de un operador de mutación?

4 — Metaheurística $SMPSO_{FHV}$

Este capítulo presenta el $SMPSO_{FHV}$ propuesto, llamado *Speed-constrained Multi-objective PSO with fast hypervolume*. Esta nueva metaheurística hace uso de nuestro método propuesto para el cálculo de una aproximación de la contribución al hipervolumen, lo cual ayuda a reducir su costo computacional. La contribución se utiliza como medida de calidad tanto para podar el archivo externo (cuando éste sobrepasa su capacidad máxima) como para elegir el líder de las partículas del cúmulo. $SMPSO_{FHV}$ está basado en el algoritmo $SMPSO$ [2]. Este último es una metaheurística que utiliza distancia de agrupamiento como medida de calidad para podar su archivo externo y para elegir el líder de las partículas del cúmulo, lo cual no permite que $SMPSO$ tenga un buen desempeño en alta dimensionalidad en el espacio de los objetivos. $SMPSO_{hv}$ [3] es otro algoritmo basado en $SMPSO$, que utiliza la contribución al hipervolumen como medida de calidad aunque no presenta ninguna mejora en el cálculo de esta contribución. Como es bien conocido, el cálculo de la contribución al hipervolumen tiene un gran costo computacional, [4].

4.1. Algoritmo $SMPSO$

El algoritmo *Speed-constrained Multi-objective PSO* se caracteriza por el uso de un mecanismo para limitar la velocidad de las partículas. Este mecanismo permite producir nuevas posiciones de las partículas de manera efectiva en aquellos casos en los que la velocidad se vuelve muy alta.

Retomando las fórmulas para actualizar la posición \vec{x}_i y velocidad \vec{v}_i en una generación t de un algoritmo PSO tenemos:

$$\vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t) \quad (4.1)$$

$$\vec{v}_i(t) = w \cdot \vec{v}_i(t-1) + C_1 \cdot r_1 \cdot (\vec{x}_{pbest_i} - \vec{x}_i) + C_2 \cdot r_2 \cdot (\vec{x}_{leader_i} - \vec{x}_i) \quad (4.2)$$

Para controlar la velocidad de las partículas se adopta un *coeficiente de restricción* (ecuación (4.3)) obtenido del factor de restricción χ originalmente desarrollado por Clerc y Kennedy (ecuación 2) en [5].

$$\chi = \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4\varphi}} \quad (4.3)$$

donde

$$\varphi = \begin{cases} C_1 + C_2 & \text{Si } C_1 + C_2 > 4 \\ 1 & \text{Si } C_1 + C_2 \leq 4 \end{cases} \quad (4.4)$$

Además, se tiene un mecanismo para que la velocidad acumulada de cada variable j (en cada partícula) sea limitada por medio de la siguiente ecuación de *restricción de velocidad* :

$$v_{i,j}(t) = \begin{cases} \delta_j & \text{Si } v_{i,j}(t) > \delta_j \\ -\delta_j & \text{Si } v_{i,j}(t) \leq -\delta_j \\ v_{i,j}(t) & \text{en cualquier otro caso} \end{cases} \quad (4.5)$$

donde

$$\delta_j = \frac{\lim_{superior_j} - \lim_{inferior_j}}{2} \quad (4.6)$$

Resumiendo el procedimiento, la velocidad de las partículas es calculada con la ecuación (4.2). El resultado de la velocidad es entonces multiplicado por el factor de restricción (ecuación (4.3)) y el valor resultante es limitado utilizando la ecuación (4.5).

El **algoritmo 3** toma los siguiente parámetros:

Algoritmo 3 Pseudocódigo de SMP SO

Entrada: G_{max} , $T_{cúmulo}$, $T_{AExterno}$ **Salida:** Aproximación al frente óptimo de Pareto

```

1: Inicializar Cúmulo de Partículas
2: Inicializar Archivo de Líderes
3: generación = 0
4: while generación <  $G_{max}$  do
5:   Calcular velocidad de las partículas // Ecuaciones (4.2) - (4.6)
6:   Actualizar posición de las partículas // Ecuación (4.1)
7:   Mutación // Turbulencia
8:   Evaluación de las funciones objetivo de cada partícula
9:   Actualizar el archivo de líderes
10:  Actualizar memoria de las Partículas ( $pbest$ )
11:  generación ++
12: end while
13: return Archivo de líderes

```

- r_1 y r_2 números aleatorios uniformemente distribuidos en el rango $[0, 1]$
- C_1 y C_2 valores en el rango $[1.5, 2.5]$
- $w = 0.1$

Pseudocódigo del **algoritmo 3** :

Comienza inicializando el cúmulo (Línea 1), lo cual incluye la posición, velocidad, y $pbest$ de las partículas. El archivo de líderes es inicializado con las soluciones no dominadas en el cúmulo (Línea 2). Posteriormente, el ciclo principal del algoritmo es ejecutado durante un cierto número de iteraciones definido por el usuario. Las velocidades y posiciones de las partículas son calculadas primero (Líneas 5 y 6) y un operador de mutación es aplicado con una probabilidad dada (Línea 7). Las partículas resultantes son evaluadas (Línea 8) y la memoria de las partículas y el archivo de líderes son actualizados (Líneas 9 y 10). El algoritmo regresa el archivo de líderes como la aproximación obtenida del conjunto de óptimos de Pareto (Línea 13).

Dado que el tamaño del archivo de líderes generalmente es fijo y éste puede llenarse rápidamente, se utiliza la distancia de agrupamiento (crowding distance) de *NSGA* –

II para decidir qué partículas deben permanecer en él. Como operador de turbulencia utiliza el operador de mutación polinomial [6], el cual es aplicado al 15 % de las partículas.

Para elegir la partícula líder al momento de aplicar la ecuación (4.2), se toman dos soluciones aleatoriamente del archivo de líderes y la solución que tenga el mayor valor de distancia de agrupamiento es seleccionada (torneo binario). Para seleccionar el $pbest_i$, éste siempre es actualizado a la nueva posición de la partícula x_i , a menos que el $pbest_i$ domine a x_i ; en tal caso $pbest_i$ no cambia. Después de aplicar la ecuación (4.1), SMPSO revisa si la posición resultante está fuera de los límites de las variables del problema. En ese caso, las posiciones son asignadas a los valores correspondientes de los límites superior o inferior; además, la dirección de la velocidad es invertida multiplicando por -1.

4.2. Algoritmos Evolutivos Multi-objetivo basados en Indicadores de Desempeño

Definición 10. (Indicador Pareto compatible) Si se tienen dos conjuntos de vectores solución A y B , y se cumple que $A \preceq B$ y $A \neq B$. Entonces siempre se mantiene la relación $I(A) > I(B)$.

Definición 11. (Indicador débilmente Pareto compatible) Si se tienen dos conjuntos de vectores solución A y B , y se cumple que $A \preceq B$. Entonces siempre se mantiene la relación $I(A) \geq I(B)$.

En estas definiciones, el indicador $I()$ es un mapeo de un conjunto de vectores objetivo a un número real.

Debido a que los AEMOs basados en dominancia de Pareto tienen dificultades cuando incrementan el número de funciones objetivo, AEMOs basados en indicadores de desempeño han sido populares en años recientes. Quizás el más popular ha sido el indicador de hipervolumen (I_H) porque es el único indicador unario conocido que es estrictamente Pareto compatible. I_H es definido como el tamaño del espacio cubierto por las soluciones del frente de Pareto, dado un punto de referencia \vec{y}_{ref} . La contribución al hipervolumen de una solución \vec{z}_i refleja las influencia de ese punto en

la calidad del frente aproximado de Pareto. La contribución de una solución \vec{z}_i al I_H es definida como sigue:

$$C_{HV}(\vec{z}_i, \mathbf{A}) = I_H(\mathbf{A}, \vec{y}_{ref}) - I_H(\mathbf{A} - \vec{z}_i, \vec{y}_{ref}) \quad (4.7)$$

donde

$$\vec{z}_i \in \mathbf{A}$$

Entonces la contribución de \vec{z}_i es el espacio que sólo es cubierto por \vec{z}_i . La figura 4.1 muestra los dos conceptos.

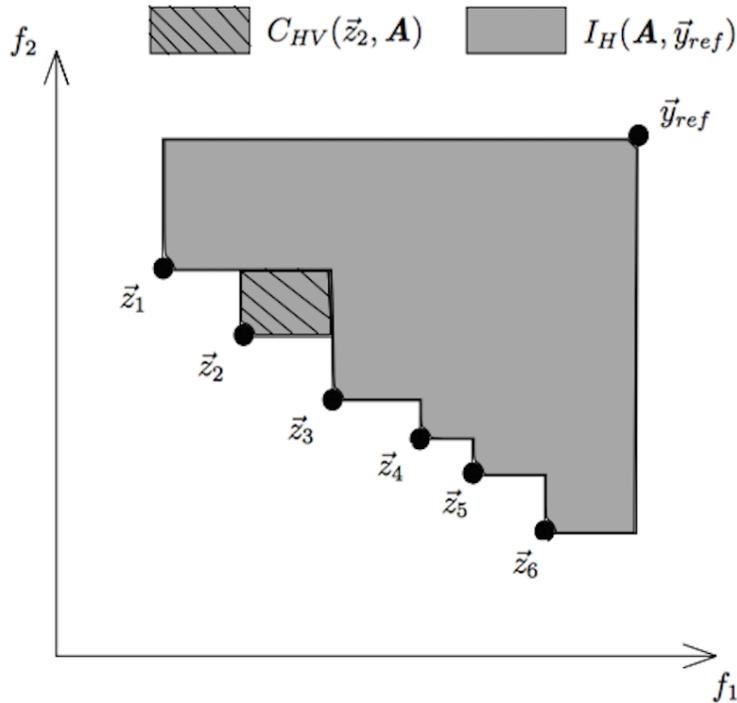


Figura 4.1: Si $\mathbf{A} = \{\vec{z}_1, \vec{z}_2, \dots, \vec{z}_6\}$ es la aproximación del frente óptimo de Pareto y \vec{y}_{ref} es el punto de referencia. El área gris sin rayas representa el hipervolumen del conjunto \mathbf{A} y el área gris con rayas representa la contribución al hipervolumen de la solución \vec{z}_2 .

La principal desventaja del I_H es su alto costo computacional, ya que el problema de calcular C_{HV} es $NP - duro$ (C_{HV} no puede ser calculado en tiempo polinomial en el número de las funciones objetivo a menos que $P = NP$ [4]).

4.2.1. Mecanismos de selección basados en Hipervolumen

Existen varios AEMOs basados en I_H . Knowles y Corne [7] utilizaron un archivo para guardar las soluciones no dominadas encontradas en cada generación. Cuando el archivo se llenaba y la dominancia de Pareto ya no podía descartar soluciones, propusieron el uso de I_H como sigue: calcular la contribución de cada solución al I_H ; si la contribución de la nueva solución era mejor que la contribución de la solución con peor contribución, entonces la nueva solución podía reemplazar a la de peor valor; si no, el archivo permanecería igual. Si utilizamos este esquema de competencia en un AEMO y tomando una población \mathbf{P} , necesitamos calcular $|\mathbf{P}| + 1$ contribuciones al indicador del hipervolumen, y por ende, este AEMO no será capaz de lidiar con POMs con más de cinco funciones objetivo (AEMOs basados en hipervolumen requieren cerca de ocho horas por simulación para resolver un POM con cinco funciones objetivo). También, Adriana [8] propuso un mecanismo de selección donde calcula la contribución al hipervolumen de la nueva solución, la contribución al hipervolumen de la solución más cercana a la nueva solución en el espacio de los objetivos y la contribución al hipervolumen de una solución aleatoria diferente a las dos anteriores. Posteriormente, estas soluciones compiten entre sí.

4.3. Propuesta para el cálculo de la contribución al Hipervolumen

Experimentalmente nos dimos cuenta de que el cálculo de la contribución al hipervolumen de una solución perteneciente a un frente de Pareto, no requiere de todas las soluciones del frente de Pareto. Estos experimentos no son mostrados en este trabajo de tesis debido a que posteriormente se encontró una base teórica en los trabajos realizados por Auger [9, 10].

Auger en [9] mencionó una propiedad interesante de este indicador cuando $m = 2$ (dos funciones objetivo), llamada **localidad** la cual dice: *dados tres puntos consecutivos en el frente de Pareto, al mover el punto medio sólo afectará la contribución al hipervolumen que es solamente dedicada a este punto, pero las demás contribuciones del conjunto permanecerán fijas*. Ver la figura 4.2. Auger en [10] realizó un estudio

similar pero para $m = 3$ (tres funciones objetivo) y mencionó que la colocación óptima de una solución no está determinada por sólo dos vecinos, como es el caso para $m = 2$.

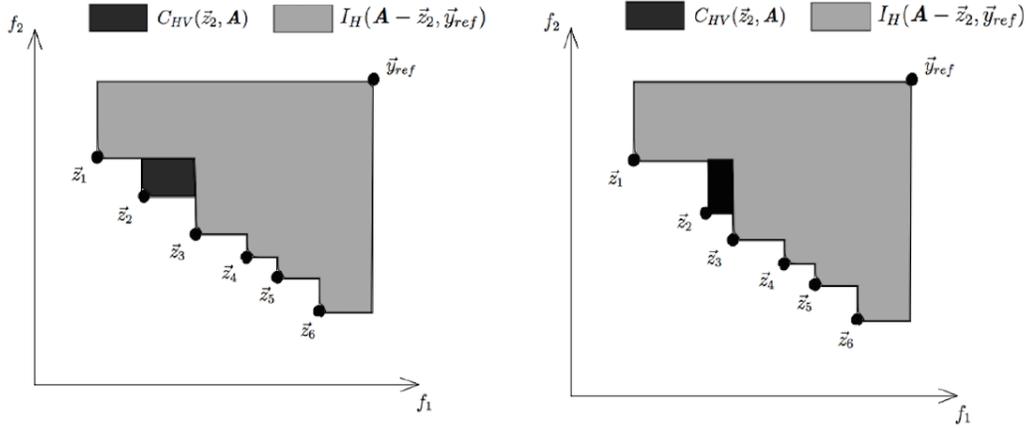


Figura 4.2: Si $\mathbf{A} = \{\vec{z}_1, \vec{z}_2, \dots, \vec{z}_6\}$ es el frente de Pareto aproximado y movemos \vec{z}_2 entre \vec{z}_1 y \vec{z}_3 , el espacio cubierto por $\{\mathbf{A} - \vec{z}_2\}$ no es afectado y sólo la contribución al hipervolumen de \vec{z}_2 es afectada.

En esta sección presentamos nuestra propuesta para el cálculo de una aproximación de la contribución al hipervolumen de una solución que se encuentra en el frente de Pareto aproximado. Nuestra técnica está basada en lo siguiente: el cálculo de la contribución de una solución \vec{z}_i , donde $\vec{z}_i \in \mathbf{A}$, no depende necesariamente de todas las soluciones del frente aproximado \mathbf{A} . Por tanto, proponemos una forma de elegir esas soluciones que sí son necesarias (de manera aproximada) para el cálculo de la contribución al hipervolumen de una solución dada. Con este esquema, pueden resolverse problemas que tienen 2,3,4, o 5 funciones objetivo.

El cálculo de la contribución del hipervolumen necesita un punto de referencia \vec{y}_{ref} en el espacio de los objetivos. Nosotros utilizamos $\vec{y}_{ref} = 1.5(\vec{y}_{nad})$. Recordemos que el vector de Nadir (\vec{y}_{nad}) fue definido en el capítulo 2 como:

$$\vec{y}_{nad} = (\max_{\vec{x} \in \mathbf{X}} f_1(\vec{x}), \dots, \max_{\vec{x} \in \mathbf{X}} f_m(\vec{x})) \quad (4.8)$$

donde m representa el número de objetivos. Durante el proceso evolutivo, el vector de Nadir debe ser actualizado siempre que el frente de Pareto aproximado cambie.

Por lo tanto, \vec{y}_{ref} también debe ser actualizado.

Si se tiene una solución $\vec{z}_i = \{f_1(\vec{x}_i), f_2(\vec{x}_i), \dots, f_m(\vec{x}_i)\}$ (en el espacio de los objetivos) que pertenece al frente de Pareto aproximado $\mathbf{A} = \{\vec{z}_1, \vec{z}_2, \dots, \vec{z}_n\}$, donde m es el número de objetivos y n el número de soluciones, entonces, el número de soluciones del frente que son necesarias para calcular la contribución al hipervolumen de \vec{z}_i son:

$$|\mathbf{B}| \leq (2^m - 2) \quad (4.9)$$

donde $\mathbf{B} \subset \mathbf{A}$ ya que $\vec{z}_i \notin \mathbf{B}$, $|\mathbf{B}|$ es el número de elementos de \mathbf{B} y $(2^m - 2)$ representa el número de regiones no dominadas vistas por la solución \vec{z}_i .

Un espacio m -dimensional se puede dividir en 2^m regiones con m ejes ortogonales entre sí, donde m es el número de objetivos. Una **región no dominada vista por una solución** \vec{z}_i es aquella región donde todos los puntos que conforman la región son no dominados con respecto a \vec{z}_i . En este caso, \vec{z}_i es considerado el origen.

Las soluciones necesarias para el cálculo de la contribución al hipervolumen de \vec{z}_i son aquellas que se encuentran más cercanas a \vec{z}_i en cada una de las regiones no dominadas. Es decir, por cada región existe una solución del frente que es la más cercana a \vec{z}_i . Si no es el caso, entonces no existen soluciones en esa región. Si existe al menos una solución $\vec{z} \in \mathbf{A}$ (con $\vec{z} \neq \vec{z}_i$) en cada una de las regiones no dominadas vistas por \vec{z}_i . Por tanto, el número de soluciones necesarias para calcular la contribución al hipervolumen de \vec{z}_i es $|\mathbf{B}| = (2^m - 2)$.

La distancia entre dos soluciones (\vec{z}_i y \vec{z}_j con m objetivos) está dada por:

$$d = \sum_{k=1}^m |f_k(\vec{z}_i) - f_k(\vec{z}_j)| \quad (4.10)$$

Una vez que se conoce \mathbf{B} , sumamos \vec{z}_i a éste y tenemos como resultado $\mathbf{C} = \mathbf{B} + \vec{z}_i$, donde $|\mathbf{C}| \leq (2^m - 2) + 1$.

Por último, calculamos la contribución de \vec{z}_i de forma habitual:

$$C_{HV}(\vec{z}_i, \mathbf{C}) = I_H(\mathbf{C}, \vec{y}_{ref}) - I_H(\mathbf{C} - \vec{z}_i, \vec{y}_{ref}) \quad (4.11)$$

Es importante resaltar que para calcular la contribución aproximada al hipervo-

lumen de una solución \vec{z}_i con m objetivos, sólo son necesarias a lo más $(2^m - 2) + 1$ soluciones (incluyendo \vec{z}_i) en comparación con utilizar el tamaño total del frente original.

Este método calcula la contribución al hipervolumen de las soluciones de un frente de Pareto dado con $m = 2$ de forma exacta. Para $m = 3$, $m = 4$ y $m = 5$ lo único que podemos asegurar es lo siguiente:

$$C_{HV}(\vec{z}_i, \mathbf{C}) = \begin{cases} C_{HV}(\vec{z}_i, \mathbf{C}) > C_{HV}(\vec{z}_i, \mathbf{A}) & \text{Si NSmétodo} \neq \text{NSRnecesarios} \\ C_{HV}(\vec{z}_i, \mathbf{C}) = C_{HV}(\vec{z}_i, \mathbf{A}) & \text{Si NSmétodo} = \text{NSRnecesarios} \end{cases} \quad (4.12)$$

donde NSmétodo es el número de soluciones que el método arroja como necesarias y NSRnecesarios es el número de soluciones realmente necesarias. Sin embargo, a pesar de que nuestro método no siempre calcula las contribuciones al hipervolumen de manera exacta de todas las soluciones del frente, el método ha mostrado en estudios preliminares que es capaz de calcular de manera exacta (en la mayoría de los casos) las contribuciones al hipervolumen de aquellas soluciones que menos contribuyen. De tal forma, este método se vuelve en un buen candidato para ser incluido en cualquier AEMO.

Ejemplo del método cuando $m = 2$

Se desea calcular la contribución al hipervolumen de $\vec{z}_2 \in \mathbf{A}$, donde el frente de Pareto aproximado es $\mathbf{A} = \{\vec{z}_1, \vec{z}_2, \dots, \vec{z}_6\}$.

Se elige el punto de referencia $\vec{y}_{ref} = 1.5(\vec{y}_{nadir})$. El número de soluciones necesarias $|\mathbf{B}|$ para calcular la contribución al hipervolumen de \vec{z}_2 es:

$$|\mathbf{B}| \leq 2 \quad (4.13)$$

Las regiones no dominadas vistas por \vec{z}_2 se representan por:

Región 1: $-\hat{i} + \hat{j}$

Región 2: $\hat{i} - \hat{j}$

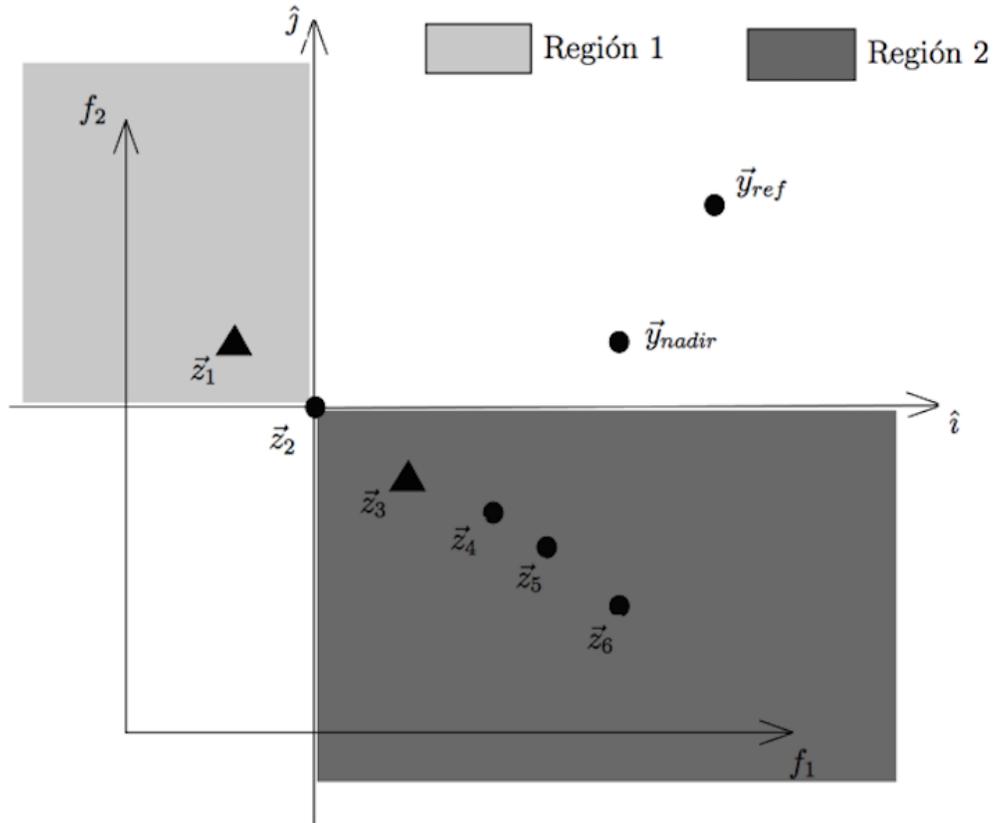


Figura 4.3: Si $\mathbf{A} = \{\vec{z}_1, \vec{z}_2, \dots, \vec{z}_6\}$ es el frente de Pareto aproximado, tenemos dos regiones vistas por \vec{z}_2 , donde se encuentran todos los demás elementos del frente. En cada una de las regiones se toma el elemento más cercano a \vec{z}_2 (Ec. 4.10). Los elementos más cercanos en cada región están representados por un triángulo.

La solución más cercana a \vec{z}_2 en la región 1 es \vec{z}_1 y la solución más cercana a \vec{z}_2 en la región 2 es \vec{z}_3 conforme la ecuación (4.10), por lo que $\mathbf{B} = \{\vec{z}_1, \vec{z}_3\}$. Los conceptos antes mencionados están representados en la figura 4.3.

Si sumamos \vec{z}_2 a \mathbf{B} , tenemos $\mathbf{C} = \mathbf{B} + \vec{z}_2 = \{\vec{z}_1, \vec{z}_2, \vec{z}_3\}$. Por último, se calcula la contribución al hipervolumen de \vec{z}_2 con:

$$C_{HV}(\vec{z}_2, \mathbf{C}) = I_H(\mathbf{C}, \vec{y}_{ref}) - I_H(\mathbf{C} - \vec{z}_2, \vec{y}_{ref}) \quad (4.14)$$

La figura 4.4 muestra el frente \mathbf{C} y la contribución al hipervolumen de \vec{z}_2 .

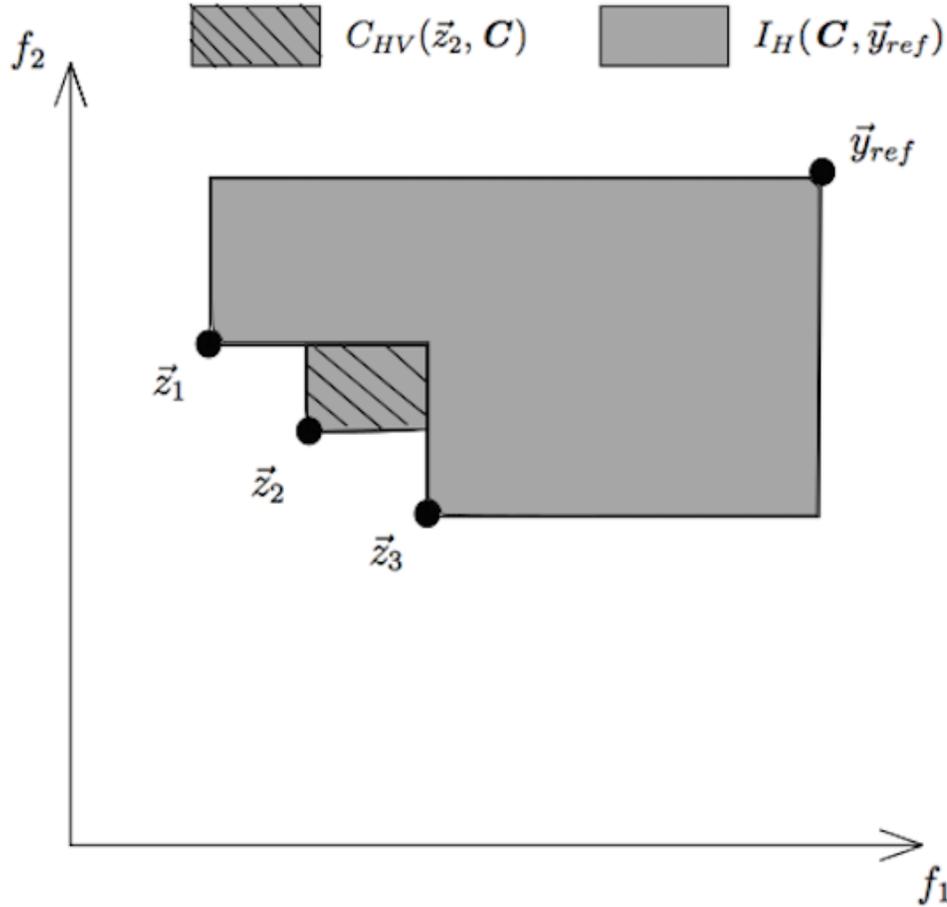


Figura 4.4: $C = \{\vec{z}_1, \vec{z}_2, \vec{z}_3\}$ es el frente de Pareto entregado por nuestro método para calcular la contribución al hipervolumen de \vec{z}_2 .

4.3.1. Nuevo mecanismo de selección basado en hipervolumen

Si se tiene un archivo de soluciones de tamaño máximo n donde sus soluciones conforman el frente de Pareto aproximado \mathbf{A} , entonces, cada vez que se llene el archivo y la dominancia de Pareto ya no puede descartar soluciones, se obtienen las contribuciones aproximadas al hipervolumen de todos los elementos del frente \mathbf{A} con nuestro método propuesto. Después de esto, si una nueva solución \vec{z}_{nueva} quiere entrar al archivo, se procede con lo siguiente: se agrega \vec{z}_{nueva} al frente \mathbf{A} (representado por $\{\mathbf{A} + \vec{z}_{nueva}\}$), se calcula la contribución aproximada al hipervolumen de \vec{z}_{nueva}

($C_{HV}(\vec{z}_{nueva}, \{\mathbf{A} + \vec{z}_{nueva}\})$) y también la de aquellas soluciones que fueron necesarias para su cálculo, representadas por el frente \mathbf{B} . Se localiza la solución que contribuye menos (la peor) al hipervolumen del frente $\{\mathbf{A} + \vec{z}_{nueva}\}$ denotada como \vec{z}_{peor} , y se identifican cuáles son las soluciones necesarias para el cálculo de su contribución al hipervolumen, representadas por el frente \mathbf{D} . Eliminamos \vec{z}_{peor} del frente $\{\mathbf{A} + \vec{z}_{nueva}\}$ y por último se calcula la contribución al hipervolumen de aquellas soluciones en \mathbf{D} .

Un defecto de este mecanismo de selección es que la contribución al hipervolumen de aquellas soluciones en las cuales el cálculo de la contribución al hipervolumen depende del punto de referencia (\vec{y}_{ref}), no siempre se actualiza. Por lo cual, cada determinado tiempo (nosotros proponemos cada generación) debe de calcularse la contribución aproximada al hipervolumen de todas las soluciones que pertenezcan al frente en un proceso de evolución para evitar que las soluciones cuyas contribuciones al hipervolumen son dependientes de \vec{y}_{ref} , se eliminen por error. Este nuevo mecanismo de selección ayuda a reducir el costo computacional enormemente cuando es añadido en un AEMO sin afectar su desempeño de manera considerable.

Si utilizamos este esquema de competencia en un AEMO y suponiendo el uso de una población \mathbf{P} , necesitamos calcular a lo más $2(2^m - 2) + 1$ contribuciones al indicador del hipervolumen comparado con $|\mathbf{P}| + 1$ contribuciones al indicador del hipervolumen que utilizaba el mecanismo de selección de la sección 4.2.1. En este caso, m es el número de objetivos.

4.4. Funcionamiento de los mecanismos de selección

4.4.1. Mecanismo de selección propuesto en [7]

Se supone que el archivo externo está lleno y contiene un frente \mathbf{P} . Si una solución *New* quiere entrar al archivo: se calculan $|\mathbf{P}| + 1$ contribuciones al hipervolumen y se elimina aquella solución que menos contribuya.

Para este caso se elimina la solución F (figura 4.5).

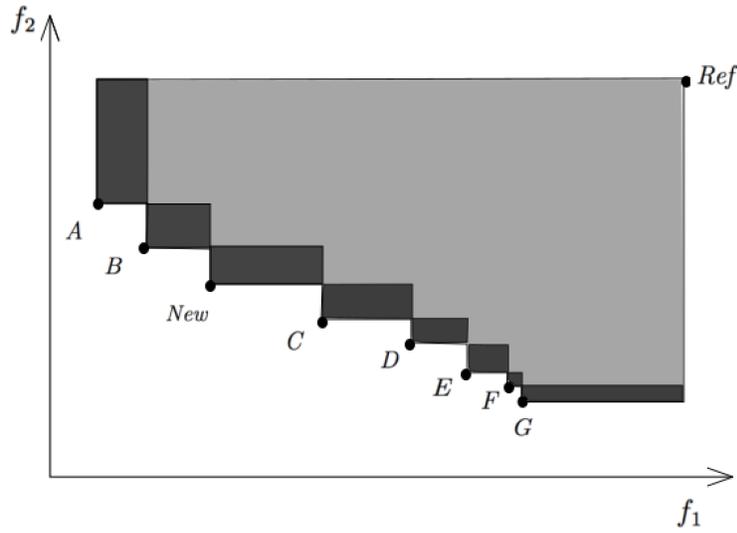


Figura 4.5: Contribución al hipervolumen de todas las soluciones.

4.4.2. Mecanismo de selección propuesto en [8]

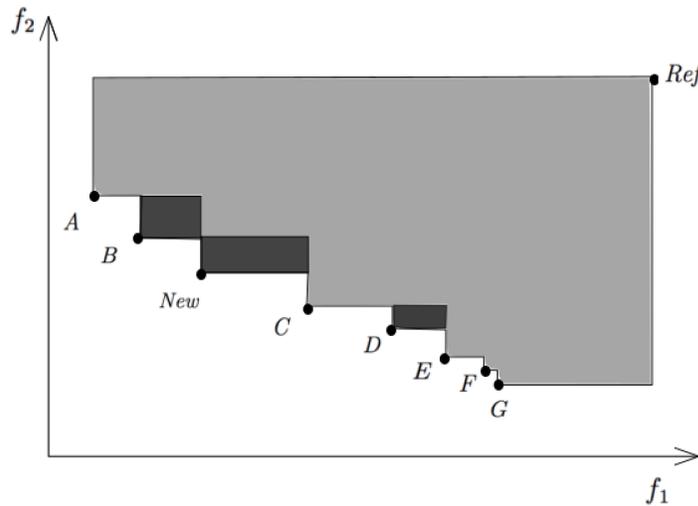


Figura 4.6: Contribución al hipervolumen de B , New y D .

Se supone que el archivo externo está lleno y contiene un frente \mathbf{P} . Si una solución New quiere entrar al archivo:

- Se calcula la contribución al hipervolumen de New .

- Se calcula la contribución al hipervolumen de B . Ya que es la solución más cercana a New .
- Se calcula la contribución al hipervolumen de D . Donde D es elegida aleatoriamente pero se debe cumplir que $D \neq New$ y $D \neq B$.
- Se elimina la solución con menor valor de la contribución al hipervolumen.

Para este caso se elimina la solución D (figura 4.6) .

4.4.3. Nuevo método de selección propuesto

Se supone que el archivo externo está lleno y contiene un frente \mathbf{P} .

Una vez que el archivo externo se llenó, se calculan las contribuciones al hipervolumen de todas las soluciones contenidas en éste (figura 4.7).

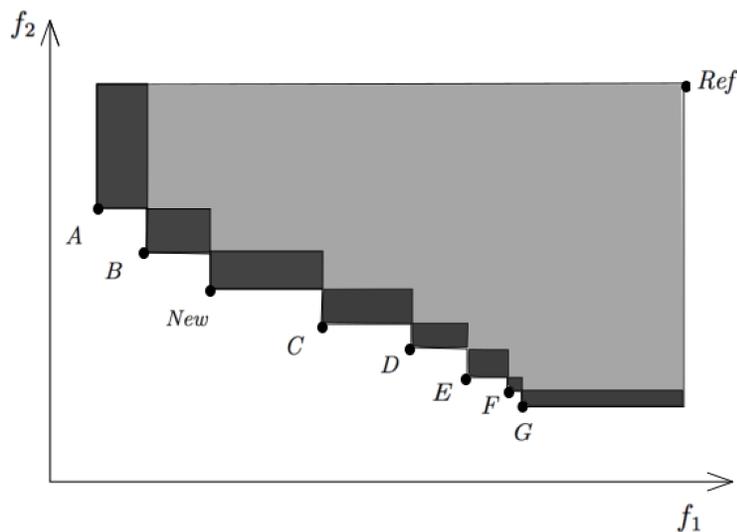


Figura 4.7: Contribución al hipervolumen de todas las soluciones.

Si una solución New quiere entrar al archivo:

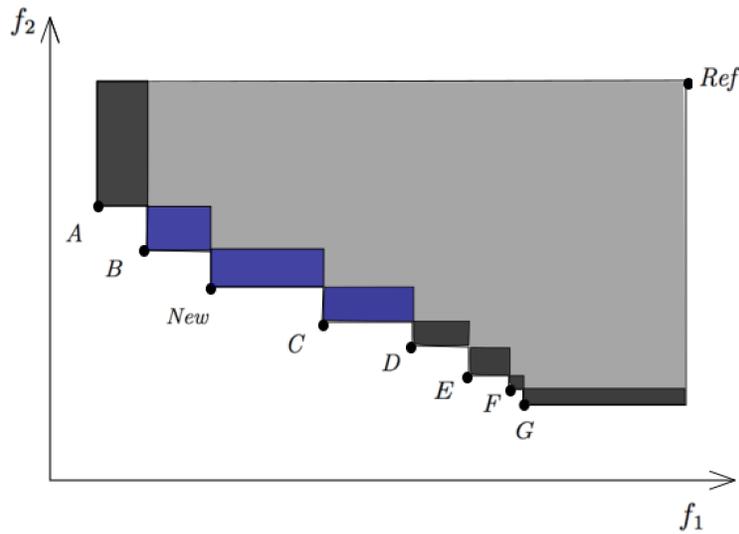


Figura 4.8: Contribución al hipervolumen de B , New y C .

- Se calcula la contribución al hipervolumen de New .
- Se calcula la contribución al hipervolumen de B , debido a que B es la solución más cercana en la dirección $-\hat{i} + \hat{j}$ (\hat{i} es la dirección X, \hat{j} es la dirección Y y New es el origen).
- Se calcula la contribución al hipervolumen de C , debido a que C es la solución más cercana en la dirección $\hat{i} - \hat{j}$ (\hat{i} es la dirección X, \hat{j} es la dirección Y y New es el origen).
- Se actualizan sólo las contribuciones de New , B y C y las contribuciones al hipervolumen de las soluciones restantes del frente \mathbf{P} permanecen sin cambios.
- Se elimina la solución con menor valor de la contribución al hipervolumen.

Para este caso se elimina la solución F (figura 4.8).

4.5. Nuestro algoritmo $SMPSO_{FHV}$

Esta sección presenta nuestra propuesta del algoritmo $SMPSO_{FHV}$, el cual se basa en el algoritmo $SMPSO$, pero en este caso se reemplaza el cálculo de la distancia de agrupamiento (*crowding distance*) por el cálculo de la contribución al hipervolumen como se propone en la sección 4.3. Adicionalmente, se modifica la forma en que se selecciona al líder. En el pseudocódigo de $SMPSO_{FHV}$ (algoritmo 4) se marca con letras *itálicas* y subrayado a los procesos que hacen a este algoritmo diferente del algoritmo $SMPSO$.

El **algoritmo 4** al igual que $SMPSO$ toma los siguientes parámetros ya que también hace uso del mecanismo para limitar la velocidad (sección 4.1):

- r_1 y r_2 números aleatorios uniformemente distribuidos en el rango $[0, 1]$
- C_1 y C_2 valores en el rango $[1.5, 2.5]$
- $w = 0.1$

Pseudocódigo del **algoritmo 4** :

Comienza inicializando el cúmulo (Línea 1), lo cual incluye la posición, velocidad, y *pbest* de las partículas. El archivo de líderes es inicializado con las soluciones no dominadas en el cúmulo (Línea 2). Posteriormente, el ciclo principal del algoritmo es ejecutado por un máximo número de iteraciones. Las velocidades y posiciones de las partículas son calculadas primero (Líneas 5 y 6) y un operador de mutación es aplicado con una probabilidad dada (Línea 7). Las partículas resultantes son evaluadas (Línea 8) y la memoria de las partículas y el archivo de líderes son actualizados (Líneas 9 y 10). Se calcula la contribución al hipervolumen de cada una de las partículas en el archivo de líderes (Línea 11). Por último, el algoritmo regresa el archivo de líderes como el conjunto de aproximación encontrado (Línea 13).

Las posiciones de las partículas se inicializan con números aleatorios dentro del rango de cada variable de decisión, las velocidades con $\vec{v}_i = \vec{0}$ y el *pbest* de una partícula es inicialmente ella misma.

Dado que el tamaño del archivo de líderes generalmente es fijo y éste puede llenarse rápidamente, se utiliza el mecanismo de selección basado en hipervolumen que se

Algoritmo 4 Pseudocódigo de nuestro $SMPSO_{FHV}$

Entrada: $G_{max}, T_{cúmulo}, T_{AExterno}$ **Salida:** Aproximación al frente óptimo de Pareto

- 1: Inicializar Cúmulo de Partículas
 - 2: Inicializar Archivo de Líderes
 - 3: generación = 0
 - 4: **while** generación < G_{max} **do**
 - 5: Calcular velocidad de las partículas // Ecuaciones (4.2)- (4.6)
 - 6: Actualizar posición de las partículas // Ecuación (4.1)
 - 7: Mutación // Turbulencia
 - 8: Evaluación de las funciones objetivo de cada partícula
 - 9: Actualizar el archivo de líderes
 - 10: Actualizar memoria de las Partículas ($pbest$)
 - 11: Calcular la contribución al HV de cada partícula
 - 12: generación ++
 - 13: **end while**
 - 14: **return** Archivo de líderes
-

propuso en la sección 4.3.1 para decidir qué partículas deben permanecer en él. Para resolver la problemática presentada por este mecanismo de selección se calcula la contribución al hipervolumen a cada generación como se puede ver en la Línea 11 del algoritmo 4.

Como operador de turbulencia se utiliza el operador de mutación polinomial [6], el cual es aplicado al 15 % de las partículas.

Para elegir la partícula líder al momento de aplicar la ecuación (4.2), se realizan L torneos binarios ($L = 0.05 \cdot (\text{número de soluciones en el archivo de líderes})$) y los ganadores de cada torneo compiten entre sí para elegir un solo ganador (aquél que tenga mayor valor de contribución al hipervolumen) que será el líder. Por cada torneo binario se toman dos soluciones aleatoriamente del archivo de líderes y la solución que tenga el mayor valor de contribución al hipervolumen es el ganador de ese torneo. Para seleccionar el $pbest_i$, éste siempre es actualizado a la nueva posición de la partícula x_i , a menos que el $pbest_i$ domine a x_i ; en tal caso, $pbest_i$ no cambia. Después de aplicar la ecuación (4.1), $SMPSO_{FHV}$ revisa si la posición resultante está fuera de

los límites de las variables del problema. En ese caso, las posiciones son asignadas a los valores correspondientes de los límites superior o inferior; además, la dirección de la velocidad es invertida multiplicando por -1.

5 — Estudio Experimental

Este capítulo está orientado a mostrar el desempeño de la propuesta $SMPSO_{FHV}$ con respecto a $NSGA-II$, $MOEA/D$ y $SMPSO$ (tres AEMOs del estado del arte). $SMPSO_{FHV}$ es analizado empleando POMs que tienen de 2 a 5 funciones objetivo.

El desempeño de los AEMOs se analiza empleando una metodología basada en estadística no paramétrica. Para cada POM en particular y para cada AEMO se ejecutan 30 corridas independientes y se mide el tiempo de ejecución de cada corrida. Los frentes de Pareto generados son evaluados empleando indicadores de desempeño del estado del arte [11], para así contar con un valor escalar que muestre su calidad. Posteriormente, para realizar comparaciones de desempeño entre los AEMOs, 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. Se ha utilizado la función $ranksum()$ proporcionada por *Matlab*® para este propósito. Siempre es considerado un nivel de confianza de 95% (valor p menor a 0.05) en las pruebas estadísticas. Todos los experimentos han sido llevados a cabo en una computadora que cuenta con un procesador Intel Core i5 con una frecuencia de reloj de 2.6 GHz y memoria RAM de 8 GB, empleando el sistema operativo OS X Yosemite.

En la sección 5.1, se describen los 12 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 AEMOs. En la sección 5.3, se presenta el experimento 1, el cual consiste en evaluar los AEMOs cuando éstos son sometidos al conjunto de problemas ZDT1-4, ZDT6 y DTLZ1-7 (con 3 objetivos). Al final de la sección 5.3 se discuten los resultados obtenidos del experimento 1. En la sección 5.4, se presenta el experimento 2, el cual tiene la finalidad de evaluar la habilidad de nuestro algoritmo

para resolver problemas de optimización de muchos objetivos (más de 3), para lo cual escalamos el problema DTLZ2 hasta cinco objetivos y comparamos su desempeño con respecto a *NSGA – II*, *MOEA/D* y *SMPSO*. Al final de la sección 5.4 se discuten los resultados obtenidos del experimento 2.

5.1. Problemas de prueba

Para comparar las metaheurísticas, se han elegido dos conjuntos de problemas de prueba. El primero es el Zitzler-Deb-Thiele (ZDT) [12], el cual consiste en un conjunto de 6 problemas (ZDT1-6) de 2 funciones objetivo y escalables en el número de variables. Para fines de esta tesis sólo utilizaremos ZDT1-4 y ZDT6. Excluimos el problema ZDT5 ya que utiliza codificación binaria. En segunda instancia, se utiliza el conjunto de problemas Deb-Thiele-Laumanns-Zitzler (DTLZ) [13], el cual consiste de 9 problemas (DTLZ1-9) escalables en el número de funciones objetivo. Para fines de esta tesis sólo utilizamos los problemas DTLZ1-7, ya que los dos restantes utilizan restricciones. Todos los problemas de prueba adoptados son usados para minimización y cuentan con características que pueden causar dificultades para los Algoritmos Evolutivos Multi-objetivo en: 1) Converger al frente óptimo de Pareto y 2) Mantener diversidad dentro de la población. En el apéndice A se definen formalmente todos ellos.

En el conjunto de problemas ZDT, se utilizan 30 variables de decisión. En los problemas DTLZ, el número total de variables de decisión está dado por $n = m + k - 1$, donde m es el número de objetivos. El valor de k es 5 para DTLZ1, 10 para DTLZ2-6 y 20 para DTLZ7.

5.2. Indicadores de evaluación de desempeño

Con el fin de evaluar y comparar los frentes de Pareto aproximados 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}) [14, 15] que mide el grado de convergencia que ostenta una aproximación al frente verdadero de Pareto (\mathcal{PF}^*). Este indicador, evalúa simultáneamente convergencia y dispersión de las soluciones.

Este indicador de desempeño 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. El punto de referencia es sumamente importante al utilizar este indicador por lo que en las tablas 5.1 y 5.2 se muestran los puntos de referencia utilizados en el experimento 1 y 2 respectivamente. Los puntos de referencia del experimento 1 fueron obtenidos de [16] y para el experimento 2 de [17] .

Tabla 5.1: Puntos de referencia para el Experimento 1.

| Experimento 1 | |
|---------------------------|----------------------------|
| 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, 21) |

Tabla 5.2: Puntos de referencia para el Experimento 2.

| Experimento 2 | |
|---------------------------|----------------------------|
| Problema de Prueba | Punto de referencia |
| DTLZ2 | (1.1, 1.1, 1.1, ...) |

Matemáticamente, el hipervolumen (HV) se describe mediante la ecuación (5.1). Cabe destacar que cuanto mayor sea su valor, indica un mejor resultado.

$$HV(\mathbf{A}, \vec{y}_{ref}) = \left\{ \bigcup \text{volumen}(\vec{z}, \vec{y}_{ref}) \mid \vec{z} \in \mathbf{A} \right\} \quad (5.1)$$

Se ha empleado el algoritmo propuesto por While en [18] ya que es generalmente aceptado como el más rápido para calcular el hipervolumen exacto.

El segundo indicador de calidad empleado es denominado *Espaciado* (S) [11, 19]. Este indicador describe numéricamente la propagación de los vectores que constituyen

la aproximación al frente de Pareto (PF_{known}). Por lo tanto, este indicador mide la varianza de la distancia entre vectores vecinos en PF_{known} . Las ecuaciones (5.2) y (5.3) muestran las fórmulas para el cálculo de este indicador.

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

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

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 (existe buena diversidad en las soluciones). Antes de aplicar este indicador se presupone que el algoritmo ha convergido cerca del frente verdadero de Pareto.

El tercer indicador seleccionado, recientemente propuesto, es el denominado *modified Inverted Generational Distance* (IGD+), propuesto por Ishibuchi en [20]. Este indicador trabaja con un frente de referencia tal como lo hace IGD [21], el cual calcula la distancia mínima de cada punto de referencia a algún punto de la aproximación al frente de Pareto que entrega un algoritmo. La gran diferencia es que IGD+ toma en cuenta la relación de dominancia de Pareto cuando estas distancias mínimas son calculadas. Debido a la consideración de la dominancia de Pareto, IGD+ es débilmente Pareto compatible. La definición formal de IGD+, considerando un problema de minimización, es mostrada en las ecuaciones (5.4) y (5.5).

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

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

donde \mathbf{A} es la aproximación al frente de Pareto y Z es el conjunto de referencia. Cuando $IGD+ \rightarrow 0$, entonces las soluciones de \mathbf{A} están muy cercanas a Z . En consecuencia, IGD+ evalúa simultáneamente convergencia y distribución de las soluciones

de una aproximación al frente de Pareto. Los frentes de referencia Z de los problemas ZDT fueron obtenidos de <http://people.ee.ethz.ch/~sop/download/supplementary/test-problems> en su versión de 100 puntos. Los frentes de referencia para los problemas DTLZ fueron obtenidos de <https://www.cs.cinvestav.mx/~emoobook> y reducidos al tamaño requerido por medio del cálculo de la contribución al hipervolumen de las soluciones.

5.3. Experimento 1: Comparación de desempeño en problemas ZDT1-4, ZDT6 y DTLZ1-7 (3 objetivos)

En este experimento se compara el desempeño de $SMPSO_{FHV}$ con respecto a tres AEMOs del estado del arte utilizando los tres indicadores antes mencionados: HV , $Espaciado$ e $IGD+$.

5.3.1. Configuración de parámetros de los algoritmos

En esta sección se presenta una descripción breve de cada algoritmo utilizado, haciendo énfasis en la configuración de parámetros establecida en la experimentación:

1. NSGA-II: El algoritmo *Non-dominating Sorting Genetic Algorithm-II* [22] es un algoritmo genético basado en obtener una nueva población de la población original aplicando los operadores genéticos típicos (selección, cruza, y mutación). Este algoritmo considera los individuos de las dos poblaciones y los ordena de acuerdo a su jerarquía, y las mejores soluciones son elegidas para crear una nueva población. En caso de tener que seleccionar entre individuos con la misma jerarquía, un estimador de densidad basado en la distancia de agrupamiento es utilizado para obtener la solución más prometedora. Este algoritmo se encuentra ya implementado en lenguaje C++ en el framework `jMetalCpp` v1.7 disponible en <https://github.com/jMetal/jMetalCpp>.

2. MOEA/D: el *Multi-Objective Evolutionary Algorithm based on Decomposition* [23] tiene como idea fundamental la transformación de un POM en N subproblemas escalares de optimización. Estos subproblemas son optimizados de forma si-

multánea usando información de los T subproblemas más cercanos. Para la generación de los subproblemas, se crean N vectores de pesos, de tal forma que exista una relación uno-a-uno entre vectores de pesos y subproblemas. *MOEA/D* emplea el método SLD [24] para la generación de los vectores de pesos. En este experimento, *MOEA/D* utiliza como método de descomposición el *enfoque de Tchebycheff* [25]. El tamaño del vecindario T es establecido en 20. El código fuente de *MOEA/D* en lenguaje C++ está disponible en <http://dces.essex.ac.uk/staff/zhang/webofmoead.htm>.

3. SMPSO: el *Speed-constrained Multi-objective PSO* [2] se describió completamente en la sección 4.1. Brevemente, es un AEMO basado en PSO (Particle Swarm Optimization), el cual utiliza un mecanismo para limitar la velocidad una vez que ésta se vuelve muy grande. Esto ayuda a que el algoritmo realice una búsqueda eficiente. Este algoritmo se encuentra ya implementado en lenguaje C++ en el framework *jMetalCpp* v1.7 disponible en <https://github.com/jMetal/jMetalCpp>.

4. SMPSO_{FHV}: El algoritmo *SMPSO* en lenguaje C++ del framework *jMetalCpp* v1.7 fue transformado a lenguaje C. Posteriormente, se modificó para así obtener *SMPSO_{FHV}* (las diferencias pueden ser encontradas en la sección 4.5). Cabe mencionar que para el cálculo del hipervolumen es utilizado el algoritmo propuesto por While en [18]. Se pudo utilizar cualquier otro método de cálculo de hipervolumen exacto y nuestro mecanismo de selección no afectaría su desempeño, pero sí lo haría su tiempo de ejecución. Es por esto que se eligió el algoritmo de While debido a que es considerado como uno de los más rápidos. El algoritmo de While codificado en lenguaje C utilizado por nosotros está disponible en <http://www.wfg.csse.uwa.edu.au/hypervolume>.

MOEA/D y *NSGA – II* a diferencia de *SMPSO* y *SMPSO_{FHV}*, utilizan el operador de cruce simulada binaria (SBX, por sus siglas en inglés) [26] con índice de distribución de cruce (N_c) establecido en 20 y porcentaje de cruce (P_c) establecido en 1.0. Los cuatro algoritmos utilizan el operador de mutación polinomial [6] con índice de distribución de mutación (N_m) establecido en 20 y porcentaje de mutación (P_m) establecido en $1/n$ (donde n representa el número de variables de decisión).

Tabla 5.3: Parámetros comunes entre las metaheurísticas en el experimento 1.

| Conjunto de problemas | h | Tamaño de población | Generaciones | Número de evaluaciones |
|-----------------------|-----|---------------------|--------------|------------------------|
| ZDT | 99 | 100 | 250 | 25000 |
| DTLZ (3 objetivos) | 14 | 120 | 208 | 24960 |

5.3.2. Parámetros de evaluación

El tamaño de población y máximo número de generaciones adoptados en este experimento se muestran en la tabla 5.3. Los valores varían de acuerdo con m ($m =$ número de objetivos). *MOEA/D* utiliza un conjunto de vectores de pesos, en la tabla se muestra el valor del parámetro h que controla el número de vectores de pesos generados. Para ser más precisos, si $\lambda^1, \dots, \lambda^N$ representan los N vectores de pesos, cada peso individual toma un valor de: $\{\frac{0}{h}, \frac{1}{h}, \dots, \frac{h}{h}\}$. Por tanto, el número de vectores de pesos está dado por la siguiente combinatoria:

$$N = C_{h+m-1}^{m-1} \quad (5.6)$$

El número total de evaluaciones de la función objetivo es determinado de tal forma que no exceda 25,000. Para el caso de *SMPSO* y *SMPSO_{FHV}*, el tamaño de la población mostrado en la tabla corresponde con el número de partículas del cúmulo y el tamaño del archivo externo (tamaño de población = tamaño del archivo externo).

Se realizaron 30 ejecuciones independientes por cada uno de los cuatro AEMOs comparados. En cada uno de los problemas de prueba adoptados, se mide el tiempo de cada ejecución y se aplican los indicadores *HV*, *S* e *IGD+* a cada frente generado. Para asegurar la superioridad de desempeño de *SMPSO_{FHV}* en los experimentos, se utiliza la prueba de suma de rango Wilcoxon de una cola, empleando un nivel de confianza del 95 % (valor p menor a 0.05).

5.3.3. Discusión de resultados

En el apéndice B se muestra, empleando tablas, el resumen de resultados numéricos obtenidos en este experimento. Con base a las tablas, las figuras 5.1, 5.2 y 5.3 muestran

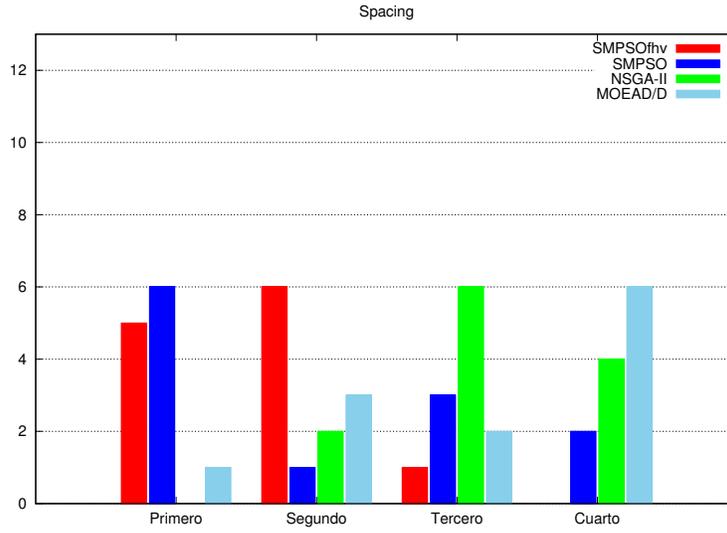


Figura 5.1: Clasificación de desempeño de las metaheurísticas probadas en el conjunto ZDT y DTLZ (3 objetivos) con base en el indicador *Espaciado*. Por cada algoritmo, se muestra el número de POMs en los que obtiene una clasificación dada.

la clasificación de desempeño de los algoritmos por cada indicador de calidad. Los histogramas muestran el número de ocasiones en que un algoritmo se clasificó en primero, segundo, tercero o cuarto lugar de desempeño para cada indicador de calidad en cada uno de los 12 problemas de este experimento.

Cuando se comparan diferentes AEMOs con respecto al indicador *Espaciado*, un menor valor indica mejor desempeño de un AEMO. En la figura 5.1 se puede observar que nuestro algoritmo ($SMPSO_{FHV}$) tiene un desempeño ligeramente mejor que $SMPSO$, ya que clasificó como primer lugar en cinco de doce PMOs y como segundo lugar en seis problemas mientras que $SMPSO$ clasificó como primero en seis PMOs y como segundo sólo en uno. Al comparar nuestro algoritmo contra $MOEA/D$ y $NSGA-II$ se puede ver que nuestro algoritmo supera a ambos ya que $MOEA/D$ sólo fue capaz de clasificar como primer lugar en un PMO y $NSGA-II$ en ninguno. Puesto que el indicador *Espaciado* evalúa la distribución de las soluciones, podemos concluir que nuestro algoritmo es capaz de obtener buena distribución en los 12 problemas de prueba adoptados aunque no haya clasificado como primero en todos ellos.

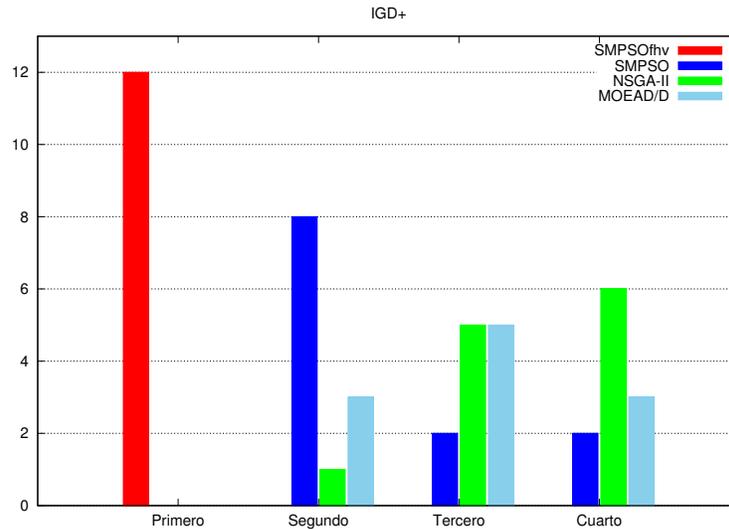


Figura 5.2: Clasificación de desempeño de las metaheurísticas probadas en el conjunto ZDT y DTLZ (3 objetivos) con base en el indicador IGD+. Por cada algoritmo, se muestra el número de POMs en los que obtiene una clasificación dada.

Con respecto al indicador de distancia generacional invertida modificada (IGD+), un menor valor indica mejor desempeño de un AEMO. En la figura 5.2 se puede observar que nuestro algoritmo tiene el mejor desempeño ya que clasificó como primero en doce de doce PMOs. *SMPSO* clasificó como segundo en ocho PMOs, *MOEA/D* en tres PMOs y *NSGA-II* sólo en uno. Este indicador evalúa convergencia y distribución, por lo que podemos concluir que nuestro algoritmo presenta buena convergencia y distribución en sus soluciones.

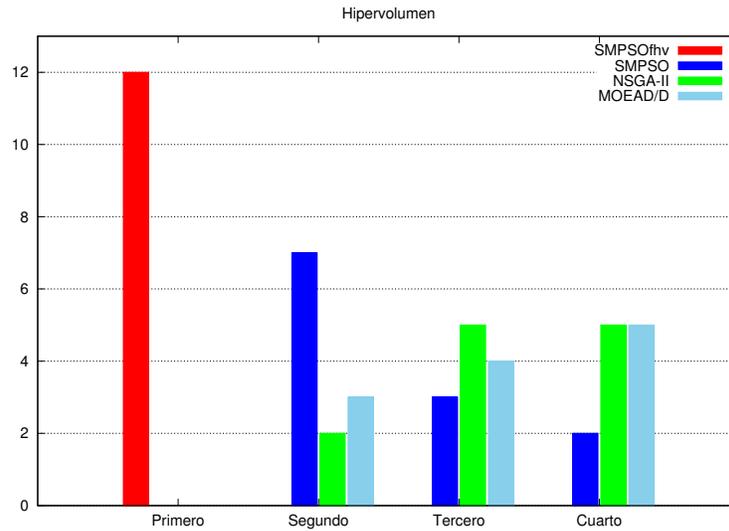


Figura 5.3: Clasificación de desempeño de las metaheurísticas probadas en el conjunto ZDT y DTLZ (3 objetivos) con base en el indicador Hipervolumen. Por cada algoritmo, se muestra el número de POMs en los que obtiene una clasificación dada.

Con respecto al indicador hipervolumen, un mayor valor representa un mejor desempeño. En la figura 5.3 se muestra que nuestro algoritmo fue el mejor ya que clasificó como primero en doce de doce PMOs. *SMPSO* clasificó como segundo en siete PMOs, *MOEA/D* en tres PMOs y *NSGA-II* en sólo dos. Este resultado se ve reafirmado de alguna forma con los dos indicadores anteriores ya que el indicador de hipervolumen evalúa convergencia y distribución de las soluciones. Se puede concluir en base a los resultados obtenidos que nuestro algoritmo es el mejor de los AEMOs utilizados en este estudio en cuanto a convergencia y distribución de las soluciones. Además, la tabla B.4 en el apéndice B muestra los tiempos de ejecución de los AEMOs en los problemas adoptados y puede notarse que nuestro algoritmo es competitivo en este aspecto.

5.4. Experimento 2: Prueba de escalabilidad utilizando el problema DTLZ2 (2-5 objetivos)

Para evaluar la escalabilidad de $SMPSO_{FHV}$ es utilizado el problema de prueba DTLZ2 con $k = 10$ y el número de objetivos m es variado de 2 a 5. Los algoritmos $NSGA - II$, $MOEA/D$ y $SMPSO$ también son puestos a prueba y su desempeño es comparado con $SMPSO_{FHV}$.

5.4.1. Configuración de parámetros de los algoritmos

La configuración de parámetros de los algoritmos se mantiene como se especifica en el experimento 1, sección 5.3.1.

5.4.2. Parámetros de evaluación

Tabla 5.4: Parámetros comunes entre las metaheurísticas en el experimento 2.

| Conjunto de problemas | h | Tamaño de población | Generaciones | Número de evaluaciones |
|-----------------------|----|---------------------|--------------|------------------------|
| DTLZ2 (2 objetivos) | 99 | 100 | 300 | 30000 |
| DTLZ2 (3 objetivos) | 14 | 120 | 250 | 30000 |
| DTLZ2 (4 objetivos) | 7 | 120 | 250 | 30000 |
| DTLZ2 (5 objetivos) | 5 | 126 | 238 | 29988 |

El tamaño de población y máximo número de generaciones adoptados en este experimento se muestran en la tabla 5.4. Los valores varían de acuerdo con m ($m =$ número de objetivos). $MOEA/D$ utiliza un conjunto de vectores de pesos, en la tabla se muestra el valor del parámetro h que controla el número de vectores de pesos generados. El número de vectores de pesos N es calculado con la ecuación (5.6). El número total de evaluaciones de la función objetivo es determinado de tal forma que no exceda 30,000. Para el caso de $SMPSO$ y $SMPSO_{FHV}$, el tamaño de la población mostrado en la tabla corresponde con el número de partículas del cúmulo y el tamaño del archivo externo (tamaño de población = tamaño del archivo externo).

Se realizaron 30 ejecuciones independientes por cada uno de los cuatro AEMOs comparados, para cada número de objetivos m ($m = \{2, 3, 4, 5\}$) en el problema DTLZ2. Adicionalmente, se midió el tiempo de cada ejecución y se aplicó el indicador de HV a cada frente generado. Para validar la superioridad de desempeño de $SMPSO_{FHV}$ en los experimentos, se utiliza la prueba de suma de rango Wilcoxon de una cola, empleando un nivel de confianza del 95 % (valor p menor a 0.05).

5.4.3. Discusión de resultados

En la figura 5.4 se muestra cómo los algoritmos $SMPSO$ y $NSGA - II$ se degradan con respecto al indicador hipervolumen al aumentar el número de objetivos. $MOEA/D$ es competitivo con nuestro algoritmo en el caso de dos objetivos, pero de tres a cinco objetivos se ve superado. En la figura 5.5 se reportan los tiempos de ejecución al aumentar el número de objetivos y se puede notar que nuestro algoritmo es competitivo en dos y tres objetivos con respecto a los algoritmos adoptados en este estudio. A pesar de tener un aumento notable en los tiempos de ejecución en las instancias con cuatro y cinco objetivos, es destacable mencionar que el tiempo de ejecución es menor a dos minutos (el uso del cálculo del hipervolumen exacto habría requerido tiempos de ejecución considerablemente mayores).

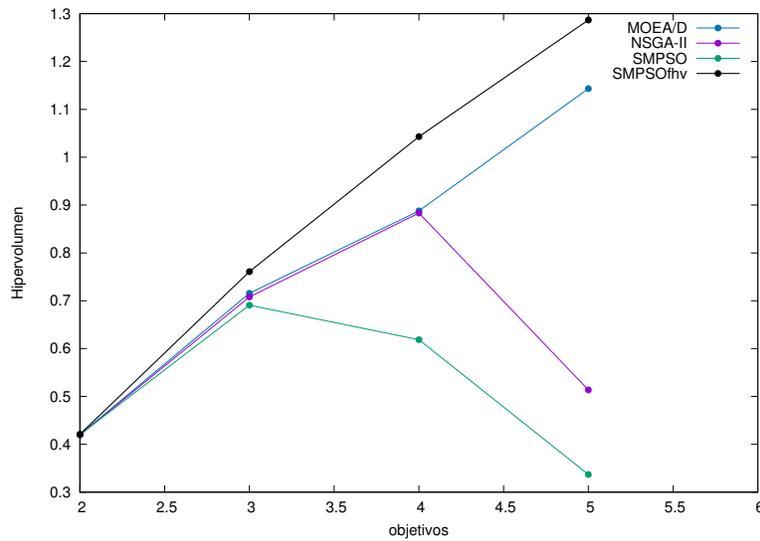


Figura 5.4: Valores del indicador Hipervolumen para el problema DTLZ2 cuando el número de objetivos es variado de 2 a 5.

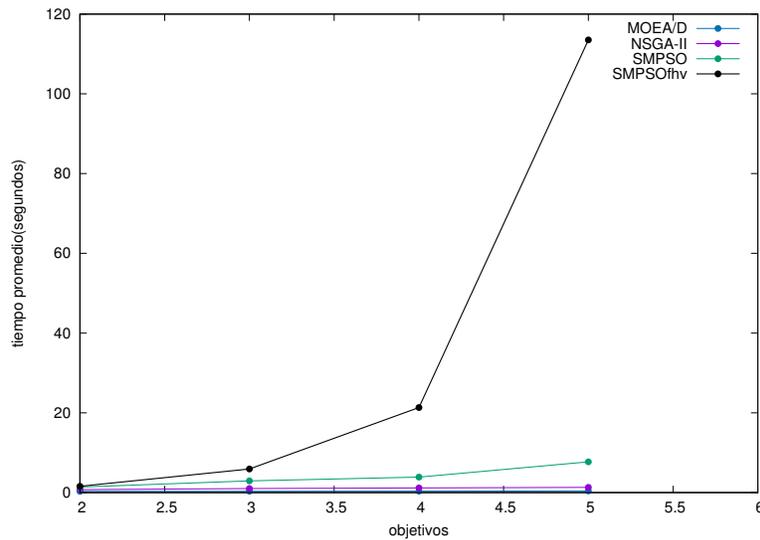


Figura 5.5: Tiempo de ejecución para el problema DTLZ2 cuando el número de objetivos es variado de 2 a 5.

6 — Resultados y trabajo futuro

En este trabajo se ha introducido un nuevo mecanismo de selección basado en el hipervolumen que calcula un valor aproximado de la contribución al hipervolumen con lo que reduce su costo computacional. El mecanismo de selección propuesto fue añadido al algoritmo *SMPSO* (uno de los mejores MOPSOs de la literatura especializada), dando como resultado un nuevo algoritmo llamado *SMPSO_{FHV}*. Este algoritmo, a pesar de calcular una aproximación a la contribución al hipervolumen ha mostrado mantener un buen desempeño en cuanto a distribución y convergencia de sus soluciones. Los resultados muestran que nuestra propuesta superó a *SMPSO*, *NSGA – II* y *MOEA/D* al resolver los conjuntos de prueba ZDT y DTLZ (3 objetivos) bajo los indicadores de calidad *HV*, *IGD+* y *Espaciado*. Además, la prueba de Wilcoxon (con un nivel de confianza del 95%) confirmó que es mejor significativamente en la mayoría de los casos. Al comparar los tiempos de ejecución por cada algoritmo y en cada problema de prueba, se puede notar que nuestro algoritmo es realmente competitivo en este aspecto. En la prueba de escabilidad nuestro algoritmo también superó a *SMPSO*, *NSGA – II* y *MOEA/D* bajo el indicador de hipervolumen y a pesar de que fue el algoritmo con mayor tiempo de ejecución requerido, su tiempo de ejecución fue menor a dos minutos. Se puede concluir que nuestro algoritmo puede ser utilizado con problemas de dos a cinco funciones objetivo sin mostrar deterioro en las soluciones entregadas por el hecho de que se calcula una aproximación de la contribución al hipervolumen. Como trabajo futuro, se propone realizar un estudio detallado del funcionamiento del método de selección propuesto y averiguar sus virtudes y debilidades más a fondo. Además, este mecanismo de selección puede ser utilizado en otros AEMOs por lo que sería interesante probar su comportamiento en otro AEMO del estado del arte, por ejemplo *NSGA – III*. Puesto que nuestro mecanismo de selección encuentra aquellas soluciones necesarias para el cálculo de la

contribución de una solución, con esto se podría definir una nueva topología (forma en la cual las partículas transmiten la información) para un MOPSO.

A — Problemas de Prueba

Con el fin de obtener un mejor entendimiento acerca del funcionamiento de los Algoritmos Evolutivos Multi-objetivo, existe una gran cantidad de conjuntos de problemas de prueba artificiales [11]. Estos problemas prueban la capacidad de los AEMOs para lidiar con diversas dificultades en los frentes óptimos de Pareto. En este apéndice, se revisan el conjunto de problemas Zitzler-Deb-Thiele (ZDT) [12] y el conjunto Deb-Thiele-Laumanns-Zitzler (DTLZ) [13], los cuales se han utilizado ampliamente en el área de optimización evolutiva multi-objetivo. El conjunto de problemas DTLZ tiene la peculiaridad de poder ser empleado en optimización con muchas funciones objetivo (dos o más). 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

Deb [27] identificó varias características que pueden causar dificultades a los Algoritmos Evolutivos Multi-objetivo para: 1) Convergencia al frente óptimo de Pareto y 2) Mantener diversidad dentro de la población. Con respecto al primer aspecto, se tienen características tales como multifrontalidad, falsos atractores y óptimos aislados. Con respecto a la segunda, se tienen características tales como convexo o no convexo, desconectado, y no uniformidad. Para cada una de las 6 características se define una función de prueba, es decir, un Problema de Optimización Multi-objetivo (ZDT1-6). Se hace notar que, ZDT5 utiliza codificación binaria por lo que es descartado en esta sección, mientras que los restantes, son POMs para espacios de diseño continuos. Es importante notar que todos los problemas ZDT sólo utilizan 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.

Los frentes óptimos de Pareto mostrados en las figuras A.1-A.5 fueron obtenidos de [11]. A continuación se ofrece una definición formal de cada uno de los problemas ZDT considerados en este trabajo.

ZDT1

Este problema presenta un frente óptimo de Pareto convexo (figura A.1) y se define de la siguiente forma:

$$\begin{aligned} \text{Dado} \quad & \vec{x} = (x_1, \dots, x_n)^T \\ \text{Minimizar} \quad & f_1(\vec{x}) = x_1 \\ & f_2(\vec{x}) = g(\vec{x}) \cdot (1 - \sqrt{f_1/g(\vec{x})}) \end{aligned} \quad (\text{A.1})$$

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

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

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

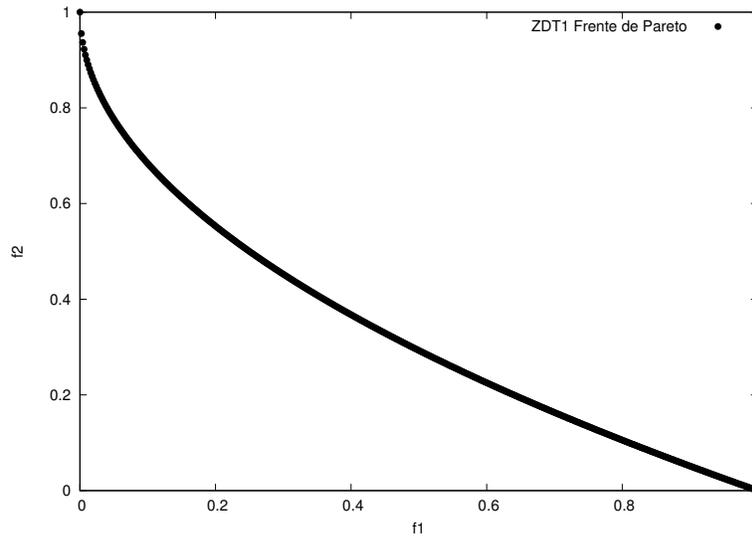


Figura A.1: Frente óptimo de Pareto para el problema ZDT1.

ZDT2

Este problema presenta un frente de Pareto no convexo (figura A.2) y se define de la siguiente forma:

$$\begin{aligned} \text{Dado} \quad & \vec{x} = (x_1, \dots, x_n)^T \\ \text{Minimizar} \quad & f_1(\vec{x}) = x_1 \\ & f_2(\vec{x}) = g(\vec{x}) \cdot (1 - (f_1/g(\vec{x}))^2) \end{aligned} \quad (\text{A.2})$$

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

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

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

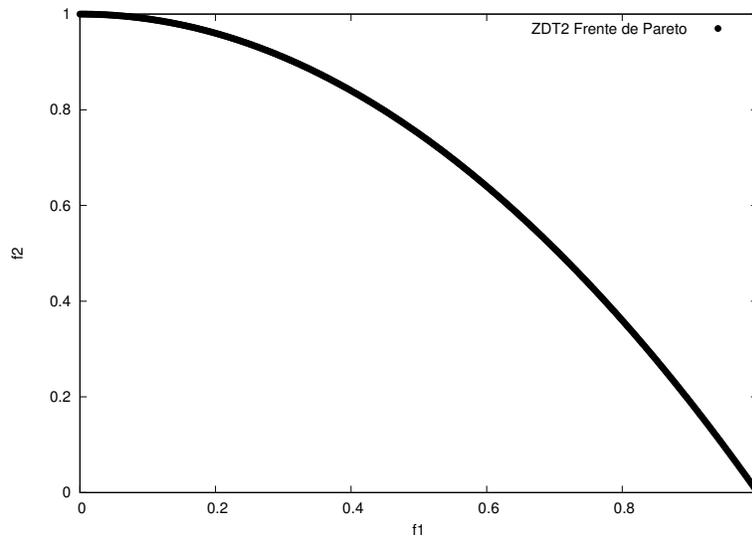


Figura A.2: Frente óptimo de Pareto para el problema ZDT2.

ZDT3

Este problema presenta un frente desconectado (figura A.3). El frente óptimo de Pareto consta de varios fragmentos convexos no contiguos. El problema se define de la siguiente forma:

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

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

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

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

sujeito a $\forall i \in \{1, \dots, n\} 0 \leq x_i \leq 1$

donde $n = 30$. El frente óptimo de Pareto se forma cuando $g(\vec{x}) = 1$. La introducción de la función senoidal en f_2 causa discontinuidad en el frente óptimo de Pareto. Sin embargo, no hay discontinuidad en el espacio de las variables de decisión.

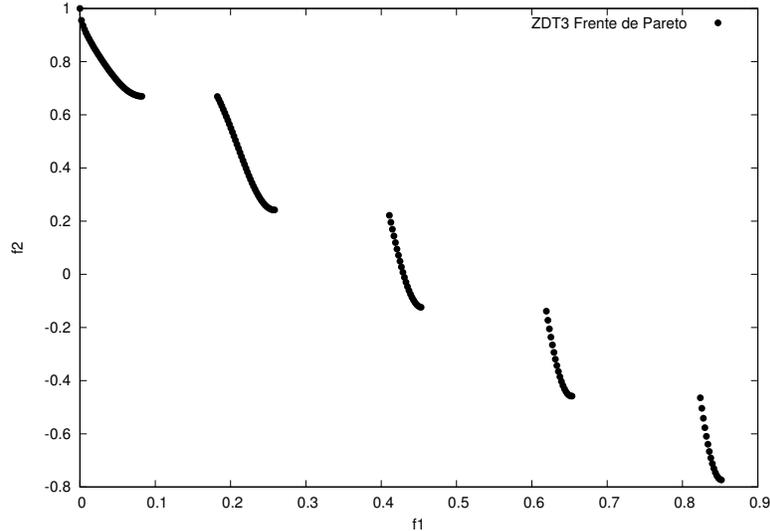


Figura A.3: Frente óptimo de Pareto para el problema ZDT3.

ZDT4

Este problema es multifrontal. Contiene 21^9 frentes óptimos de Pareto locales y, por lo tanto, prueba la habilidad de un AEMO para tratar con esta propiedad. El problema se define de la siguiente forma:

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

donde $n = 10$. El frente óptimo de Pareto de este problema (figura A.4) se forma con $g(\vec{x}) = 1$ y el mejor frente local con $g(\vec{x}) = 1.25$.

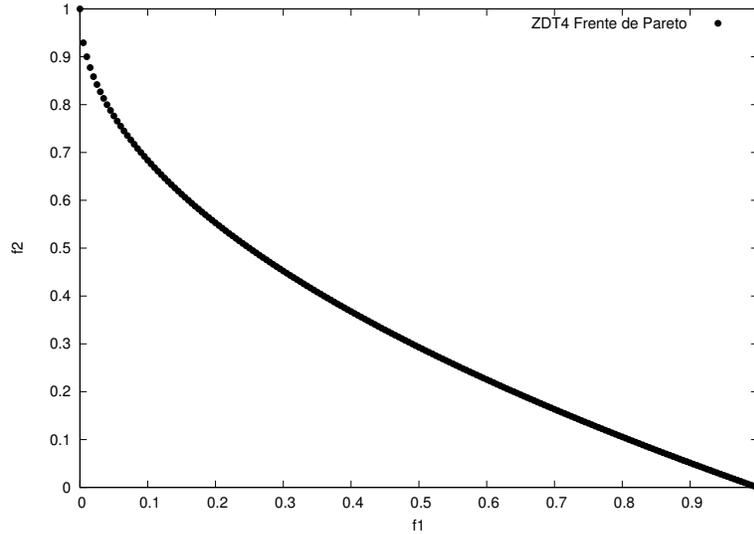


Figura A.4: Frente óptimo de Pareto para el problema ZDT4.

ZDT6

Este problema presenta un frente óptimo de Pareto no conexo (figura A.5) e incluye dos dificultades causadas por la no uniformidad del espacio de búsqueda :

- 1) El conjunto de óptimos de Pareto no está uniformemente distribuido a lo largo del frente óptimo de Pareto (el frente está sesgado para soluciones en la cuales $f_1(\vec{x})$ es

cercano a uno).

2) La densidad de las soluciones es más baja cerca del frente óptimo de Pareto y más alta al estar lejos del frente.

El problema se define de la siguiente forma:

$$\begin{aligned} \text{Dado} \quad & \vec{x} = (x_1, \dots, x_n)^T \\ \text{Minimizar} \quad & f_1(\vec{x}) = 1 - \exp(-4x_1) \sin^6(6\pi x_1) \\ & f_2(\vec{x}) = g(\vec{x}) \cdot (1 - (f_1/g(\vec{x}))^2) \end{aligned} \quad (\text{A.5})$$

$$\text{donde} \quad g(\vec{x}) = 1 + 9 \cdot \left[\frac{\sum_{i=2}^n x_i}{n-1} \right]^{0.25}$$

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

donde $n = 10$. El frente óptimo de Pareto se forma con $g(\vec{x}) = 1$.

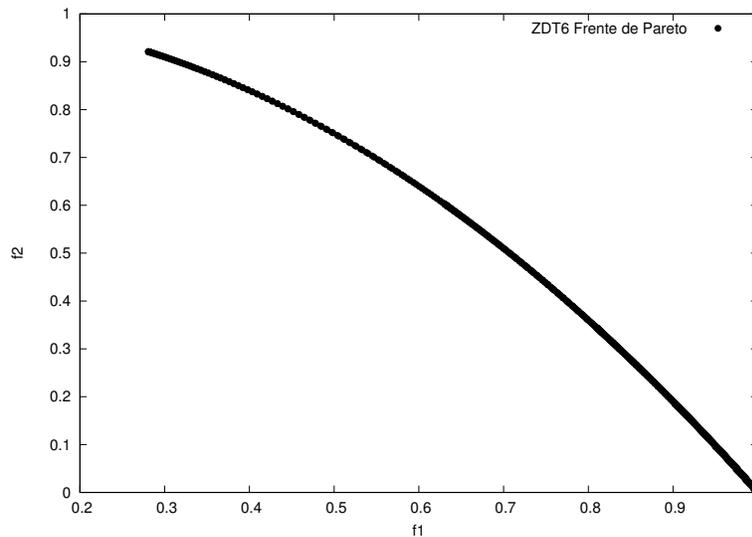


Figura A.5: Frente óptimo de Pareto para el problema ZDT6.

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

El conjunto de POMs DTLZ contiene 9 problemas [13], donde siete de ellos no tienen restricciones y dos sí, pero para nuestro trabajo sólo son de interés aquellos que no tienen restricciones (DTLZ1-7). Estos problemas presentan un gran avance con respecto a los antes mencionados (ZDT), dado que permiten investigar las propiedades de los AEMOs en problemas con muchas funciones objetivo de una manera controlada. 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.

Los frentes óptimos de Pareto mostrados en las figuras A.6-A.12 fueron obtenidos de [11]. A continuación se ofrece una definición formal de cada uno de los problemas DTLZ considerados en este trabajo.

DTLZ1

Este problema es separable y multifrontal, teniendo un frente óptimo de Pareto lineal. El problema se define de la siguiente forma:

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

El conjunto de óptimos de Pareto corresponde a $\vec{y} = (0.5, 0.5, \dots)^T$ y todos los valores del frente óptimo de Pareto se encuentran en el hiperplano lineal: $\sum_{i=1}^m f_i = 0.5$. 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 Algoritmo Evolutivo Multi-objetivo antes de alcanzar el frente óptimo de Pareto global. En la figura A.6 se muestra el frente óptimo de Pareto correspondiente a DTLZ1.

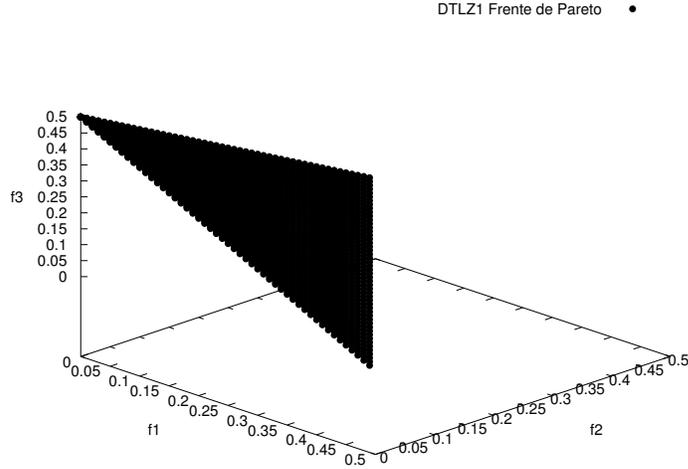


Figura A.6: Frente óptimo de Pareto para el problema DTLZ1 con tres objetivos.

DTLZ2

Este problema es unifrontal, separable y su geometría es cóncava (figura A.7). El problema se define de la siguiente forma:

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

El conjunto de óptimos de Pareto corresponde con $\vec{y} = (0.5, 0.5, \dots)^T$ y los valores de las funciones objetivo deben satisfacer: $\sum_{i=1}^m f_i^2 = 1$. Se recomienda $k = 10$. Esta función es muy útil para investigar la habilidad de un Algoritmo Evolutivo

Multi-objetivo para escalar su funcionamiento a altas dimensiones en el espacio de los objetivos.

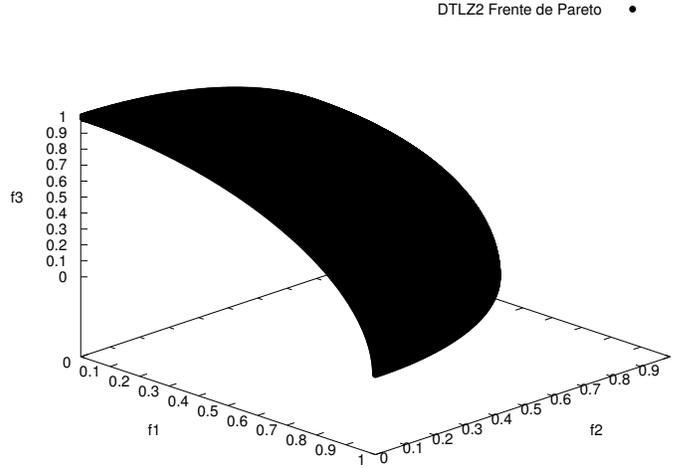


Figura A.7: Frente óptimo de Pareto para el problema DTLZ2 con tres objetivos.

DTLZ3

Este problema simplemente agrega multifrontalidad a DTLZ2 mediante la definición de una nueva función g . El problema se define de la siguiente forma:

$$\begin{aligned}
 \text{Dado} \quad & \vec{x} = (x_1, \dots, x_{m-1}, x_m, \dots, x_n)^T \\
 \text{Minimizar} \quad & f_1(\vec{x}) = (1 + g(\vec{y})) \prod_{i=1}^{m-1} \cos(x_i \pi / 2) \\
 & f_{j=2:m-1}(\vec{x}) = (1 + g(\vec{y})) (\prod_{i=1}^{m-j} \cos(x_i \pi / 2)) \text{sen}(x_{m-j+1} \pi / 2) \\
 & f_m(\vec{x}) = (1 + g(\vec{y})) \text{sen}(x_1 \pi / 2) \tag{A.8} \\
 \text{donde} \quad & y_{i=1:k} = \{x_m, x_{m+1}, \dots, x_n\} \\
 & g(\vec{y}) = 100[k + \sum_{i=1}^k (y_i - 0.5)^2 - \cos(20\pi(y_i - 0.5))] \\
 \text{sujeto a} \quad & \forall i \in \{1, \dots, n\} \quad 0 \leq x_i \leq 1
 \end{aligned}$$

Se sugiere $k = 10$. 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 Algoritmo Evolutivo Multi-objetivo puede estancarse en alguno de ellos antes de lograr convergencia al frente verdadero

en $g = 0$. El conjunto de óptimos de Pareto del frente global corresponde con $\vec{y} = (0.5, 0.5, \dots)^T$. El siguiente frente óptimo de Pareto local se encuentra en $g = 1$. En la figura A.8 se muestra el frente óptimo de Pareto del problema DTLZ3.

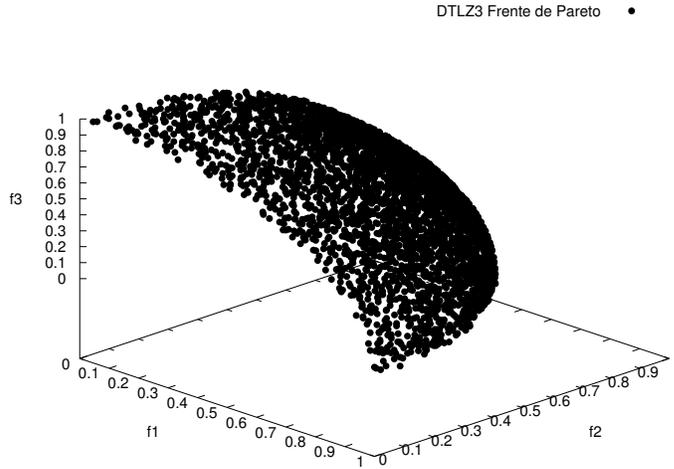


Figura A.8: Frente óptimo de Pareto para el problema DTLZ3 con tres objetivos.

DTLZ4

Este problema es cóncavo, separable y unifrontal (figura A.9). Su utilidad principal recae en probar la habilidad de un Algoritmo Evolutivo Multi-objetivo para mantener una buena distribución de soluciones. El problema se define de la siguiente forma:

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

Se sugiere emplear $\alpha = 100$ y $k = 10$. Las modificaciones presentadas por este proble-

ma 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.

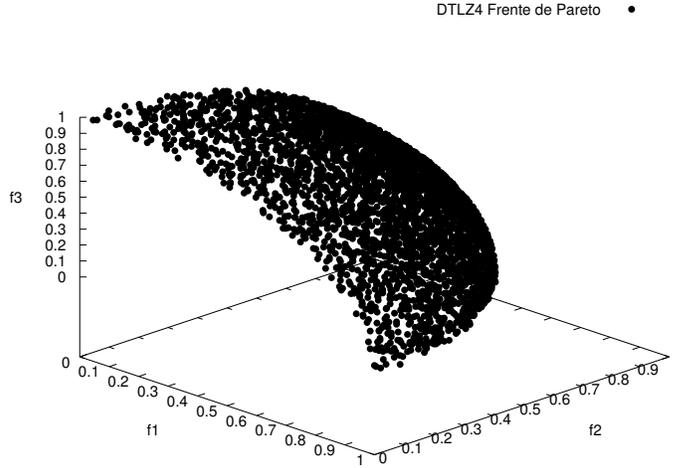


Figura A.9: Frente óptimo de Pareto para el problema DTLZ4 con tres objetivos.

DTLZ5

Nuevamente, DTLZ2 es modificado con el objeto de producir un problema unifrontal y degenerado (figura A.10). El problema se define de la siguiente forma:

$$\begin{aligned} \text{Dado} \quad & \vec{x} = (x_1, \dots, x_{m-1}, x_m, \dots, x_n)^T \\ \text{Minimizar} \quad & f_1(\vec{x}) = (1 + g(\vec{y})) \prod_{i=1}^{m-1} \cos(\theta_i \pi / 2) \\ & f_{j=2:m-1}(\vec{x}) = (1 + g(\vec{y})) (\prod_{i=1}^{m-j} \cos(\theta_i \pi / 2)) \text{sen}(\theta_{m-j+1} \pi / 2) \\ & f_m(\vec{x}) = (1 + g(\vec{y})) \text{sen}(\theta_1 \pi / 2) \end{aligned} \quad (\text{A.10})$$

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

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

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

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

Este problema prueba la habilidad de un Algoritmo Evolutivo Multi-objetivo para converger hacia una curva y permitirá demostrar visualmente (sólo graficando f_m con otra función objetivo) el desempeño del algoritmo. Dado que existe un sesgo natural para las soluciones cercanas a la curva del óptimo de Pareto, 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 utilizar $k = 10$.

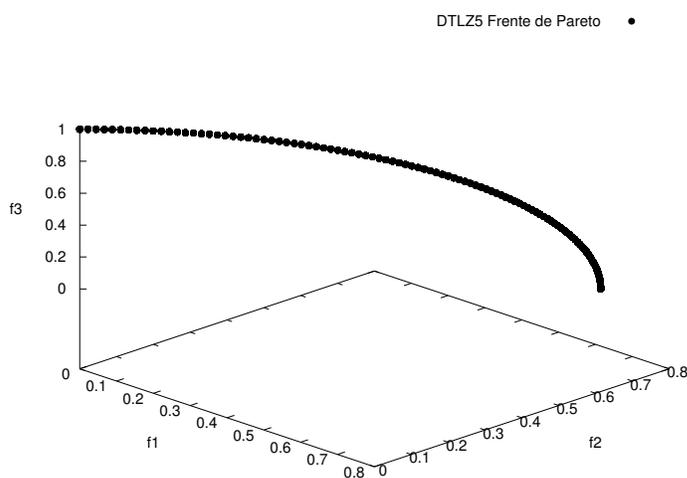


Figura A.10: Frente óptimo de Pareto para el problema DTLZ5 con tres objetivos.

DTLZ6

Modificando DTLZ5, un problema más complejo surge como consecuencia de cambiar la función g . El problema resultante es unifrontal, degenerado, presenta sesgo y es una transformación muchos-a-uno. El problema se define de la siguiente forma:

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

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

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

$$f_m(\vec{x}) = (1 + g(\vec{y})) \operatorname{sen}(\theta_1 \pi / 2) \tag{A.11}$$

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

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

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

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

El frente óptimo de Pareto corresponde con $\vec{y} = (0, 0, \dots)^T$ y se muestra en la figura [A.11](#). La modificación impuesta, vuelve más difícil el problema para un Algoritmo Evolutivo Multi-objetivo. La falta de convergencia al frente de Pareto verdadera causa que los algoritmos busquen una superficie no dominada como frente resultante, mientras que el frente óptimo de 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. Se sugiere utilizar $k = 10$.

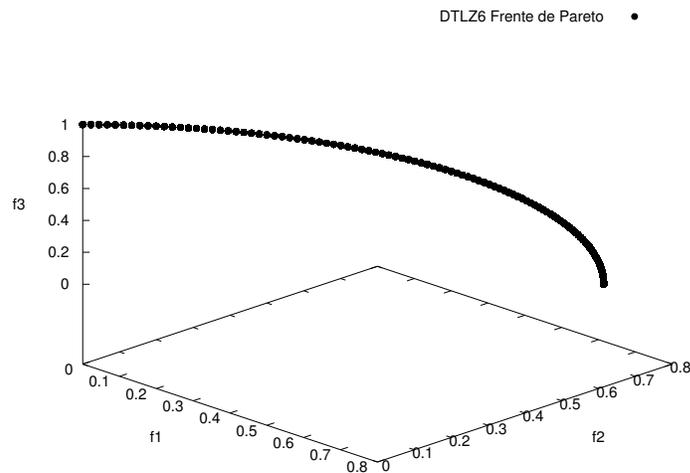


Figura A.11: Frente óptimo de Pareto para el problema DTLZ6 con tres objetivos.

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. El problema se define de la siguiente forma:

$$\begin{aligned}
 \text{Dado} \quad & \vec{x} = (x_1, \dots, x_{m-1}, x_m, \dots, x_n)^T \\
 \text{Minimizar} \quad & f_{j=1:m-1}(\vec{x}) = x_j \\
 & f_m(\vec{x}) = (1+g(\vec{y}))\left(m - \sum_{i=1}^{m-1} \left[\frac{f_i}{1+g(\vec{y})}(1+\text{sen}(3\pi f_i))\right]\right) \\
 \text{donde} \quad & y_{i=1:k} = \{x_m, x_{m+1}, \dots, x_n\} \quad (\text{A.12}) \\
 & g(\vec{y}) = 1 + \frac{9}{k} \sum_{i=1}^k y_i \\
 \text{sujeto a} \quad & \forall i \in \{1, \dots, n\} \quad 0 \leq x_i \leq 1
 \end{aligned}$$

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

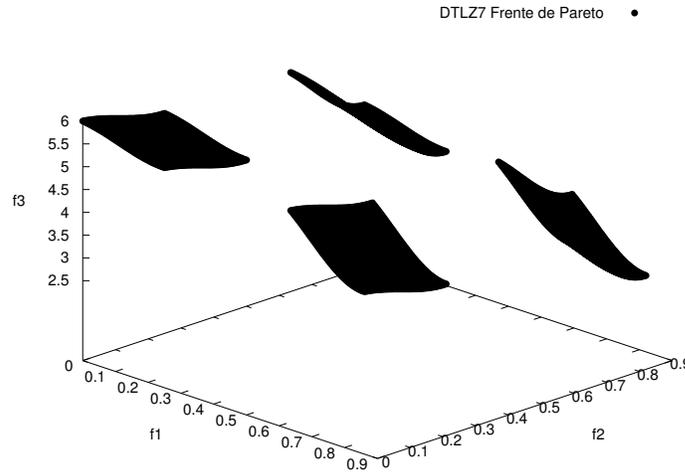


Figura A.12: Frente óptimo de Pareto para el problema DTLZ7 con tres objetivos.

B — Resultados Numéricos

Este apéndice presenta los resultados numéricos obtenidos en los experimentos hechos en el capítulo 5. Los datos están organizados en tablas (sección B.1 y B.2) y figuras (sección B.3).

B.1. Tablas de comparación de los AEMOs (experimento 1)

Los datos del experimento 1 son organizados por tablas donde cada una de éstas evalúa el conjunto de problemas bajo un indicador (HV , $IGD+$ y $Espaciado$). Cada tabla contiene la información de la media \bar{x} y desviación estándar σ (tal valor se muestra entre paréntesis) del valor de los indicadores. Además, existe una tabla que representa la media del tiempo de ejecución necesario por cada AEMO para resolver un cierto PMO. El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de $SMPSO_{FHV}$ en la prueba de Wilcoxon con un nivel de confianza del 95% (valor p menor a 0.05). En todas las tablas, los dos mejores valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor.

Tabla B.1: Comparación de resultados obtenidos del indicador HV por $MOEA/D$, $NSGA-II$, $SMPSO$ y nuestro algoritmo $SMPSO_{FHV}$ para el conjunto de problemas de prueba ZDT y DTLZ (3 objetivos).

| Hipervolumen | | | | |
|--------------|-----------------------------|--------------------------------|----------------------------|---|
| Problema | MOEA/D $\bar{x}(\sigma)$ | NSGA – II $\bar{x}(\sigma)$ | SMPSO $\bar{x}(\sigma)$ | SMPSO _{FHV} $\bar{x}(\sigma)$ |
| ZDT1 | 120.603653(0.087198)# | 120.652981(0.002169)# | 120.661787(0.000087) # | 120.662109(0.000024) |
| ZDT2 | 120.123072(0.798128)# | 120.311799(0.002981)# | 120.328554(0.000070) # | 120.328859(0.000007) |
| ZDT3 | 127.989416(1.640689)# | 128.767400(0.001334) # | 128.745843(0.049335)# | 128.773843(0.008046) |
| ZDT4 | 111.639202(3.133896)# | 81.269836(10.865684)# | 120.658012(0.002679) # | 120.662061(0.000048) |
| ZDT6 | 116.038918(0.942948)# | 112.983947(0.521464)# | 117.514683(0.000134) # | 117.514912(0.000029) |
| DTLZ1(3) | 0.968571(0.000754) # | 0.929405(0.080623)# | 0.966068(0.014135)# | 0.974187(0.000176) |
| DTLZ2(3) | 7.385004(0.0.001259) # | 7.361126(0.020570)# | 7.356077(0.006553)# | 7.429236(0.000558) |
| DTLZ3(3) | 336.269138(23.177404)# | 67.527773(98.185116)# | 338.813232(8.773980) # | 340.620552(6.830202) |
| DTLZ4(3) | 7.389452(0.003054)# | 7.378667(0.014630)# | 7.364996(0.019411)# | 7.425374(0.012514) |
| DTLZ5(3) | 59.841970(0.001034)# | 59.865831(0.001102)# | 59.871441(0.000757)# | 59.875164(0.000087) |
| DTLZ6(3) | 1316.960405(0.547851)# | 1299.509472(1.673649)# | 1319.066274(0.000601) # | 1319.069737(0.000251) |
| DTLZ7(3) | 16.048249(0.129315)# | 16.285166(0.014768)# | 16.269578(0.015387)# | 16.392244(0.001243) |

Tabla B.2: Comparación de resultados obtenidos del indicador $IGD+$ por $MOEA/D$, $NSGA-II$, $SMPSO$ y nuestro algoritmo $SMPSO_{FHV}$ para el conjunto de problemas de prueba ZDT y DTLZ (3 objetivos).

| IGD+ | | | | |
|----------|-----------------------------|--------------------------------|----------------------------|---|
| Problema | MOEA/D $\bar{x}(\sigma)$ | NSGA – II $\bar{x}(\sigma)$ | SMPSO $\bar{x}(\sigma)$ | SMPSO _{FHV} $\bar{x}(\sigma)$ |
| ZDT1 | 0.003147(0.000594)# | 0.003539(0.000186)# | 0.002455(0.000269) # | 0.002267(0.000322) |
| ZDT2 | 0.005943(0.014091)# | 0.003533(0.000214) # | 0.002499(0.000299)# | 0.002104(0.000299) |
| ZDT3 | 0.013078(0.022182)# | 0.001843(0.000101)# | 0.001670(0.000429)# | 0.001018(0.000135) |
| ZDT4 | 0.554100(0.238603)# | 3.193289(0.981366)# | 0.002750(0.000297)# | 0.002154(0.000283) |
| ZDT6 | 0.060985(0.021326)# | 0.233658(0.029182)# | 0.002095(0.000229) # | 0.001938(0.000206) |
| DTLZ1(3) | 0.020571(0.001918)# | 0.074923(0.099867)# | 0.026057(0.009441)# | 0.012675(0.000729) |
| DTLZ2(3) | 0.045668(0.001228)# | 0.046538(0.002778)# | 0.057181(0.002979)# | 0.017342(0.000812) |
| DTLZ3(3) | 0.525234(0.889361)# | 7.671927(3.839428)# | 0.109376(0.081902)# | 0.060648(0.076405) |
| DTLZ4(3) | 0.046044(0.001508)# | 0.048080(0.002141)# | 0.057455(0.008535)# | 0.029181(0.006384) |
| DTLZ5(3) | 0.005125(0.000013)# | 0.002634(0.000145)# | 0.001937(0.000160)# | 0.001569(0.000226) |
| DTLZ6(3) | 0.123883(0.038659)# | 1.162660(0.082196)# | 0.001749(0.000114)# | 0.001535(0.000160) |
| DTLZ7(3) | 0.208311(0.263047)# | 0.041564(0.003188)# | 0.048539(0.005023)# | 0.015786(0.000731) |

Tabla B.3: Comparación de resultados obtenidos del indicador *Espaciado* por *MOEA/D*, *NSGA – II*, *SMPSO* y nuestro algoritmo *SMPSO_{FHV}* para el conjunto de problemas de prueba ZDT y DTLZ (3 objetivos).

| Espaciado | | | | |
|-----------|------------------------------|---------------------------------|-----------------------------|--|
| Problema | MOEA/D $\bar{x} (\sigma)$ | NSGA – II $\bar{x} (\sigma)$ | SMPSO $\bar{x} (\sigma)$ | SMPSO _{FHV} $\bar{x} (\sigma)$ |
| ZDT1 | 0.009806(0.000902)# | 0.006674(0.000921) # | 0.001474(0.000223) # | 0.002742(0.000391) |
| ZDT2 | 0.005783(0.001322)# | 0.006888(0.000631)# | 0.001336(0.000165) # | 0.004250(0.000413) |
| ZDT3 | 0.021962(0.001868)# | 0.007508(0.000466)# | 0.003997(0.001033) # | 0.005857(0.000508) |
| ZDT4 | 0.109452(0.134248)# | 0.072776(0.042845)# | 0.002114(0.000359) # | 0.003091(0.000483) |
| ZDT6 | 0.007807(0.004692)# | 0.009475(0.002033)# | 0.161956(0.211907)# | 0.001447(0.000203) |
| DTLZ1(3) | 0.027799(0.000908) # | 0.843736(3.365517) # | 0.038139(0.091910)# | 0.009582(0.001050) |
| DTLZ2(3) | 0.078221(0.001264)# | 0.051411(0.004314)# | 0.051278(0.004844) # | 0.037354(0.002181) |
| DTLZ3(3) | 0.120775(0.089109) # | 2.639188(3.558492)# | 3.320615(9.110912)# | 2.314299(7.592437) |
| DTLZ4(3) | 0.080520(0.001864)# | 0.051473(0.004178)# | 0.059437(0.014143)# | 0.041095(0.009643) |
| DTLZ5(3) | 0.007112(0.000100)# | 0.007687(0.000775) | 0.002939(0.000587) # | 0.007559(0.000527) |
| DTLZ6(3) | 0.081209(0.076573)# | 0.118630(0.017131)# | 0.002346(0.000445) # | 0.007457(0.000460) |
| DTLZ7(3) | 0.075637(0.027562)# | 0.066537(0.008775)# | 0.069129(0.009970)# | 0.048161(0.003441) |

Tabla B.4: Media del tiempo de ejecución necesario por *MOEA/D*, *NSGA – II*, *SMPSO* y nuestro algoritmo *SMPSO_{FHV}* para resolver cada problema del conjunto ZDT y DTLZ (3 objetivos).

| Tiempo por cada ejecución(segundos) | | | | |
|-------------------------------------|---------------------|------------------------|--------------------|-----------------------------------|
| Problema | MOEA/D \bar{x} | NSGA – II \bar{x} | SMPSO \bar{x} | SMPSO _{FHV} \bar{x} |
| ZDT1 | 0.392372s | 0.998503s | 1.348095s | 1.163147s |
| ZDT2 | 0.355814s | 0.992668s | 1.446957s | 1.168401s |
| ZDT3 | 0.382340s | 0.993331s | 0.469604s | 0.552431s |
| ZDT4 | 0.311473s | 1.009011s | 0.680899s | 0.800961s |
| ZDT6 | 0.316212s | 0.993589s | 0.791282s | 0.709016s |
| DTLZ1(3) | 0.218305s | 0.662691s | 0.930408s | 2.196107s |
| DTLZ2(3) | 0.264887s | 0.743724s | 0.425149s | 1.245310s |
| DTLZ3(3) | 0.264887s | 0.743724s | 0.425150s | 1.245310s |
| DTLZ4(3) | 0.277927s | 0.836847 | 1.412292s | 3.271978s |
| DTLZ5(3) | 0.273356s | 0.762603s | 1.894200s | 1.919124s |
| DTLZ6(3) | 0.243517s | 0.777613s | 1.306176s | 0.995256s |
| DTLZ7(3) | 0.350805s | 1.013050s | 1.057886s | 2.189276s |

B.2. Tablas de comparación de los AEMOs (experimento 2)

Los datos del experimento 2 son organizados en una tabla donde se evalúa, con respecto al indicador HV , el problema DTLZ2 cuando el número de funciones objetivo se incrementa ($m = \{2, 3, 4, 5\}$). La tabla contiene la información de la media \bar{x} , desviación estándar σ (tal valor se muestra entre paréntesis). Además, se añade la media del tiempo de ejecución necesario (en segundos) por cada AEMO para resolver el problema DTLZ2 para un cierto valor de m . El símbolo # se coloca cuando hay una diferencia estadísticamente significativa con respecto al resultado de $SMPSO_{FHV}$ en la prueba de Wilcoxon con un nivel de confianza del 95% (valor p menor a 0.05). En la tabla, los dos mejores valores se resaltan en tonos de gris, donde el más oscuro corresponde con el mejor valor.

Tabla B.5: Comparación de resultados obtenidos del indicador HV por $MOEA/D$, $NSGA - II$, $SMPSO$ y nuestro algoritmo $SMPSO_{FHV}$ para el problema DTLZ2 usando de 2 a 5 objetivos. El tiempo de ejecución promedio también es reportado.

| Número de Objetivos (m) | Tiempo por cada ejecución(segundos) | | | |
|-----------------------------------|---|--|--|--|
| | MOEA/D \bar{x} (σ) tiempo (s) | NSGA – II \bar{x} (σ) tiempo (s) | SMPSO \bar{x} (σ) tiempo (s) | SMPSO _{FHV} \bar{x} (σ) tiempo (s) |
| 2 | 0.420117# (0.000005) 0.296408s | 0.419369# (0.000210) 0.740713s | 0.420443# (0.000068) 1.392658s | 0.421008 (0.000007) 1.573000s |
| 3 | 0.7155335932# (0.0010985063) 0.333499s | 0.7077757481# (0.0055753405) 1.001962s | 0.6907789644# (0.0074940598) 2.940502s | 0.7607997199 (0.0004268652) 5.912880s |
| 4 | 0.8877273827# (0.0012193132) 0.353928s | 0.8832622355# (0.0157349149) 1.154944s | 0.6186617958# (0.0550533902) 3.870611s | 1.0428842150 (0.0019745785) 21.318708s |
| 5 | 1.1433409158# (0.0009806104) 0.400735s | 0.5135937492# (0.0660124795) 1.310926s | 0.3369037205# (0.0618680550) 7.707680s | 1.2866743506 (0.0042731162) 113.524727s |

B.3. Gráficas de aproximación de los AEMOs (2D y 3D)

En esta sección se presentan las aproximaciones a los frentes de Pareto obtenidos por $SMPSO_{FHV}$, $SMPSO$, $NSGA-II$ y $MOEA/D$ en el conjunto de problemas ZDT y DTLZ (3 objetivos). Las gráficas corresponden a la mediana del valor del hipervolumen de 30 ejecuciones independientes.

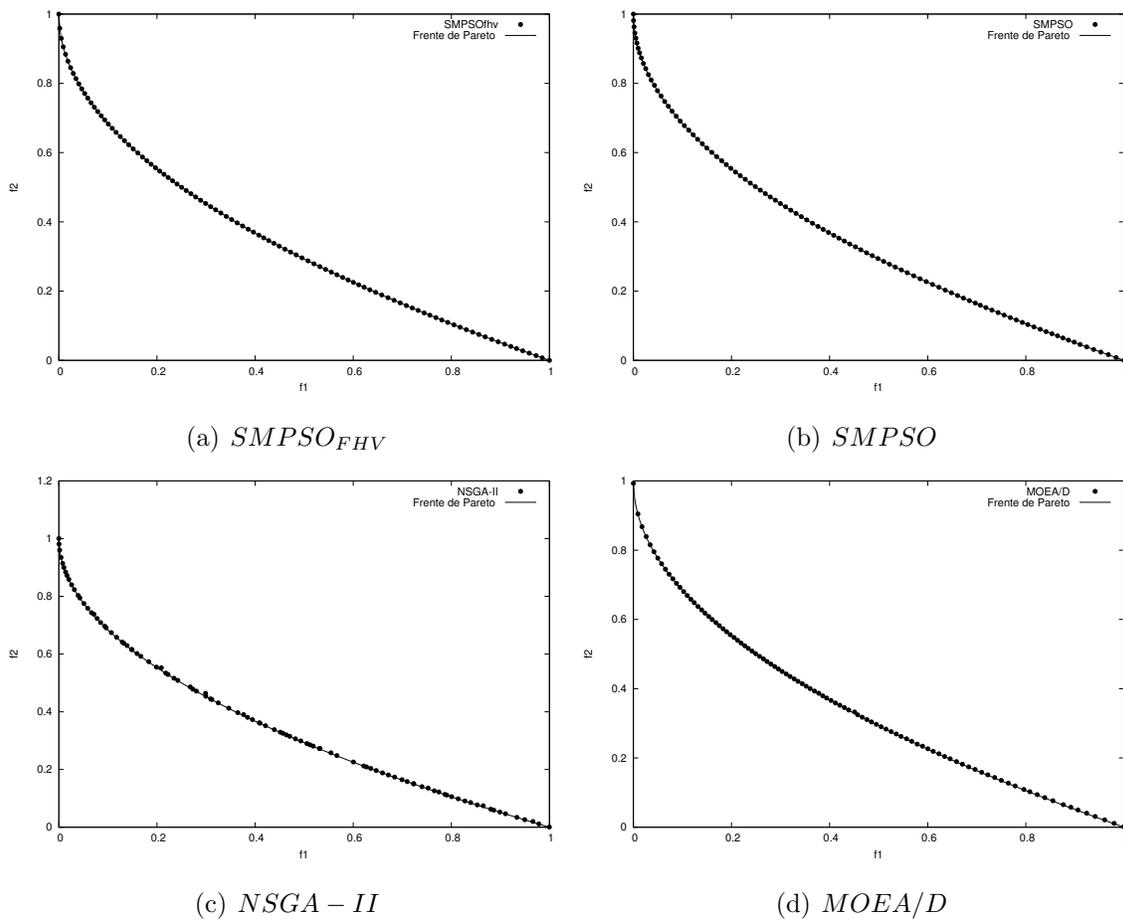


Figura B.1: Problema de prueba ZDT1.

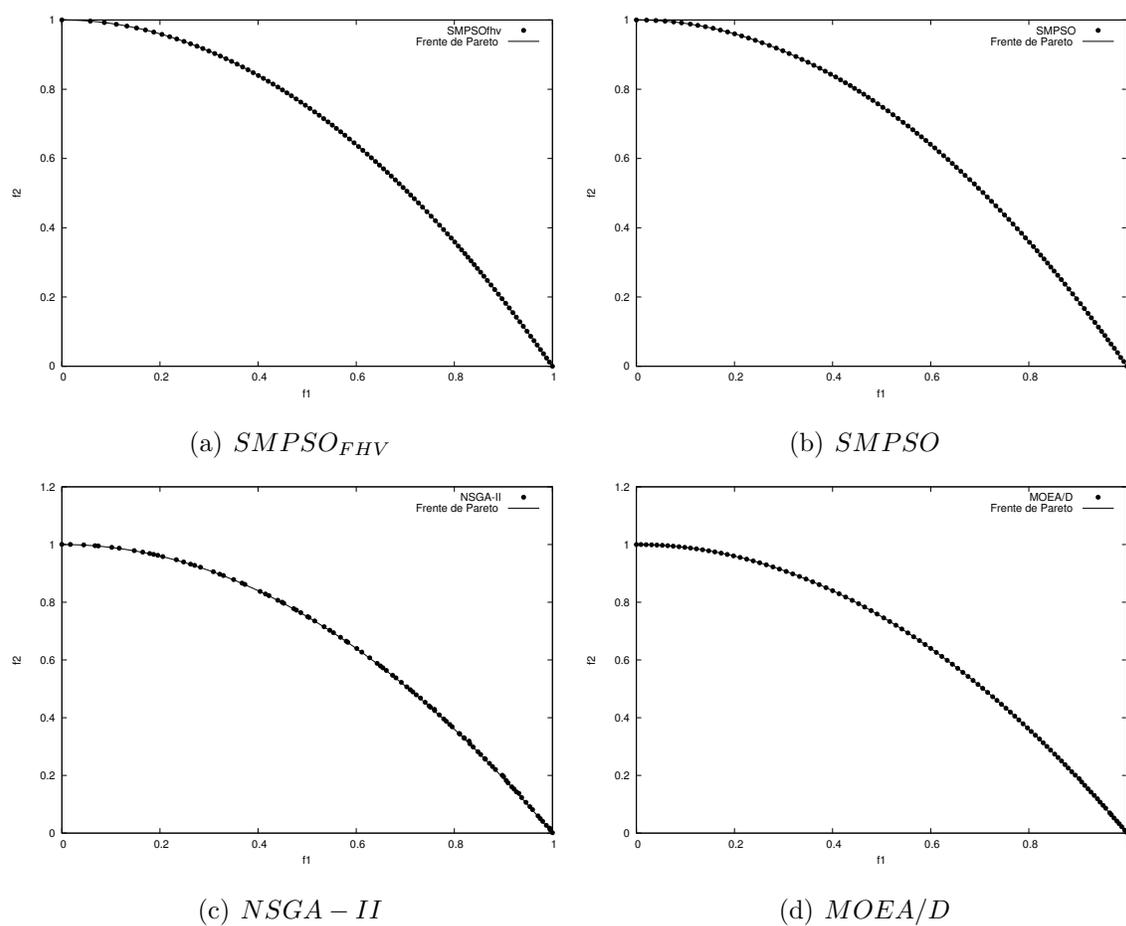


Figura B.2: Problema de prueba ZDT2.

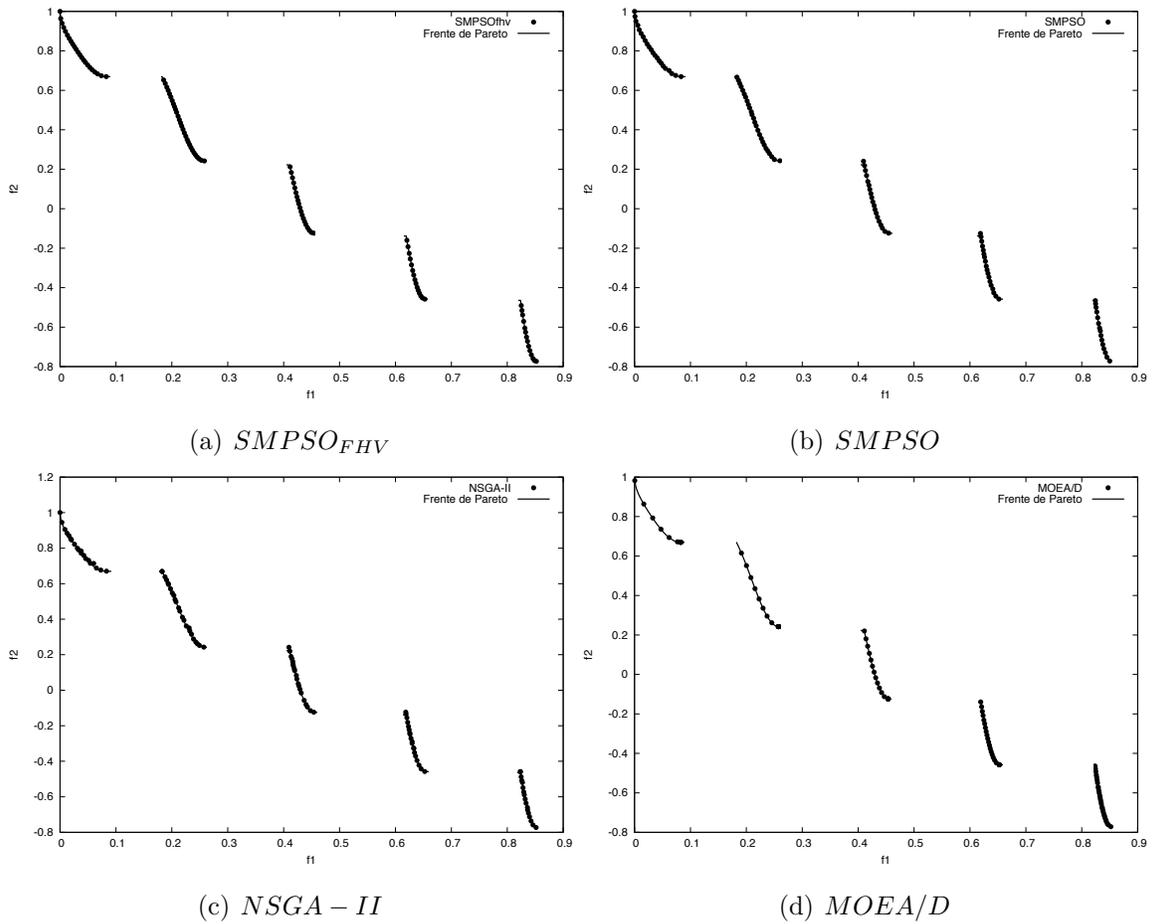
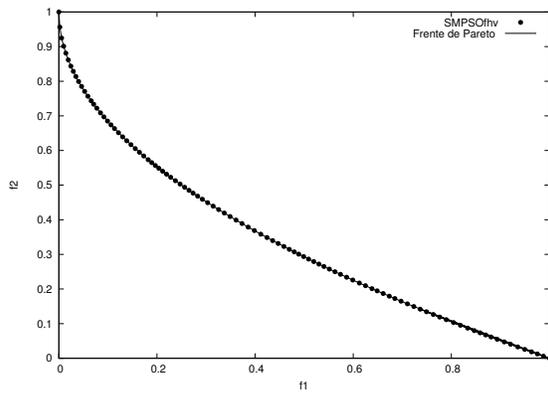
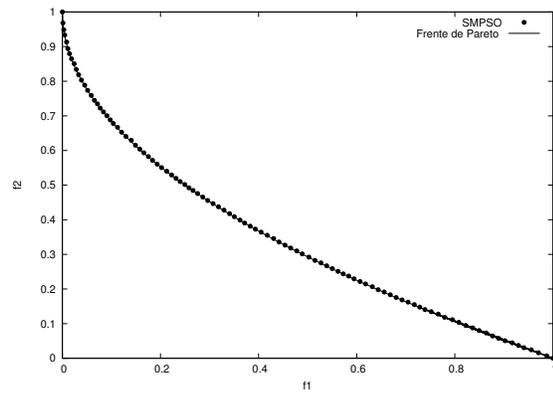


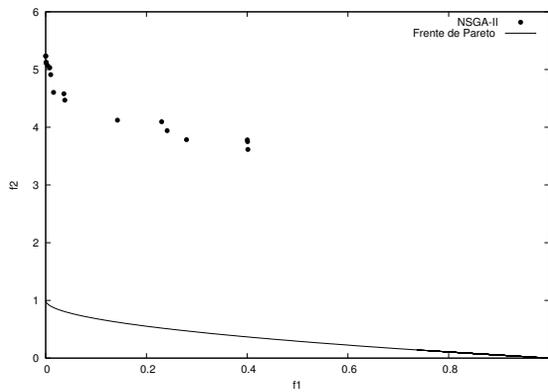
Figura B.3: Problema de prueba ZDT3.



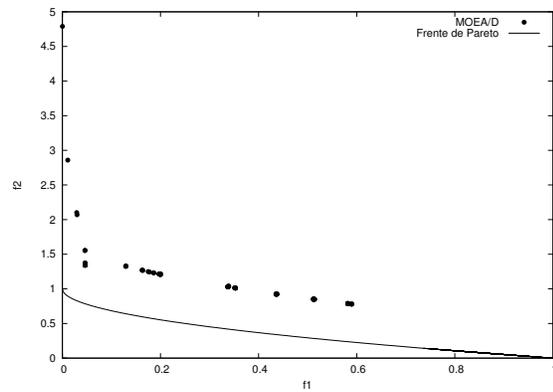
(a) $SMPSO_{FHV}$



(b) $SMPSO$

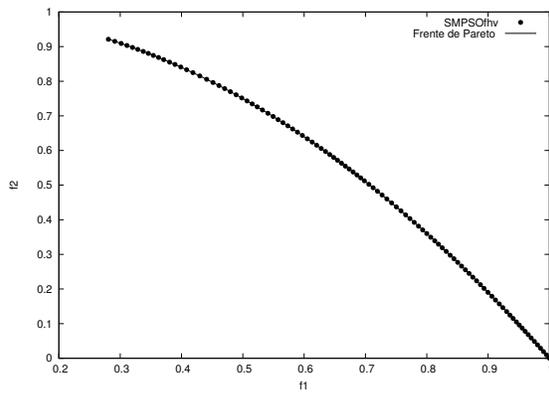


(c) $NSGA-II$

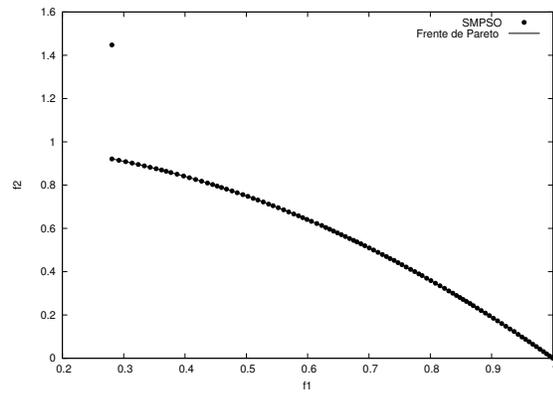


(d) $MOEA/D$

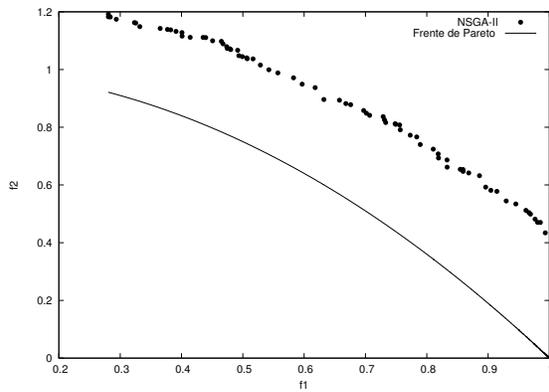
Figura B.4: Problema de prueba ZDT4.



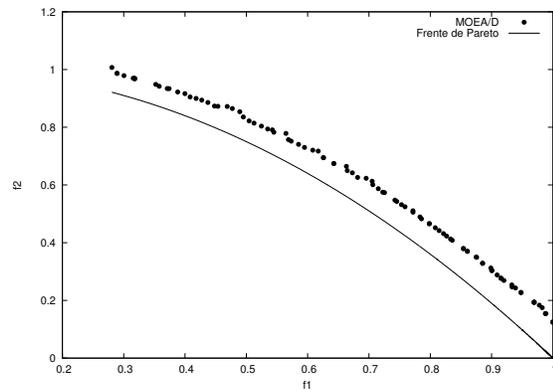
(a) $SMPSO_{FHV}$



(b) $SMPSO$



(c) $NSGA-II$



(d) $MOEA/D$

Figura B.5: Problema de prueba ZDT6.

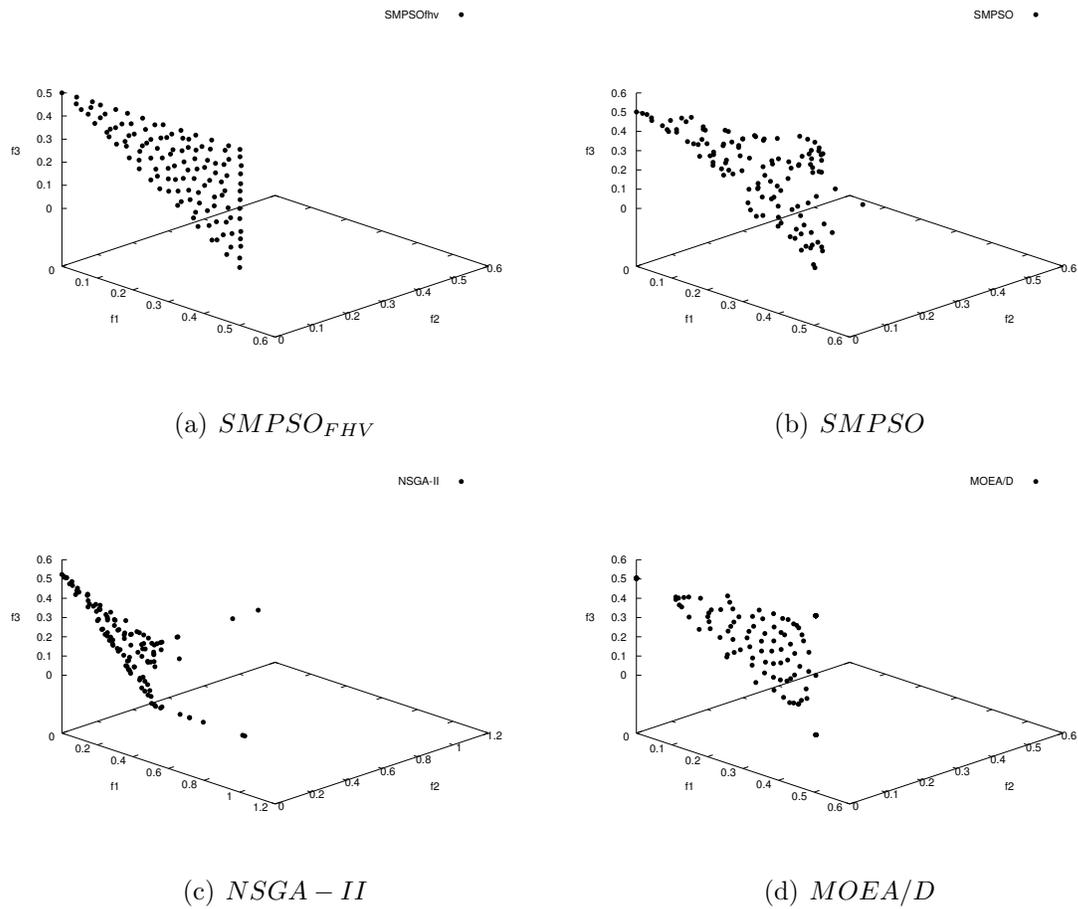


Figura B.6: Problema de prueba DTLZ1 con tres objetivos.

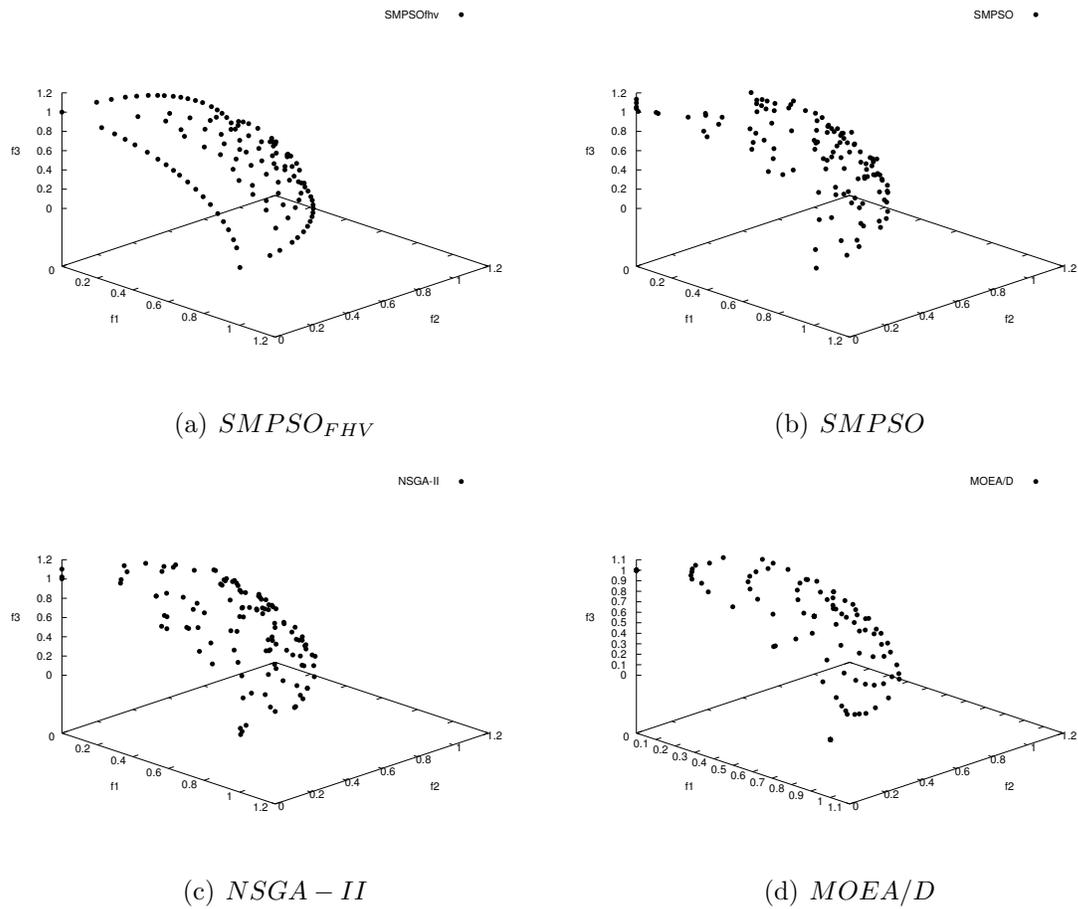
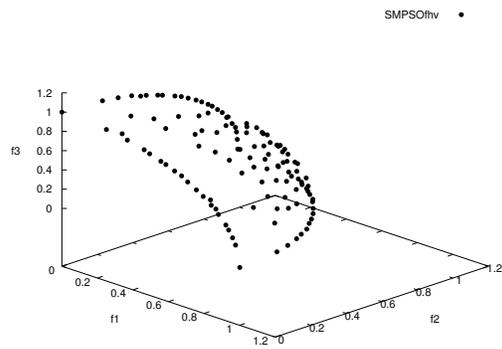
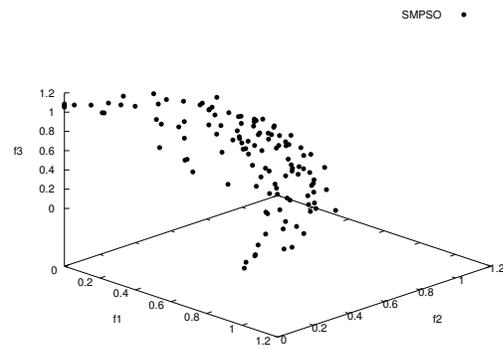


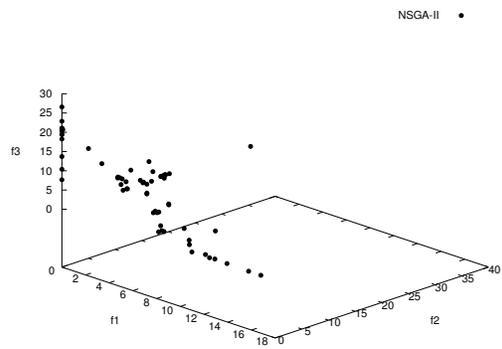
Figura B.7: Problema de prueba DTLZ2 con tres objetivos.



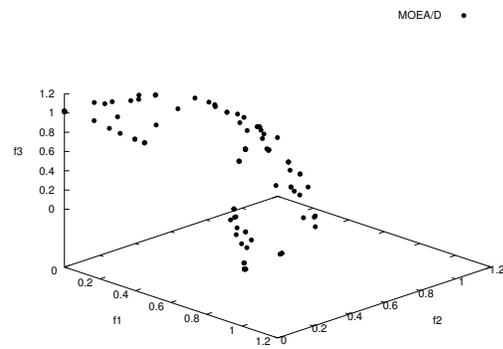
(a) $SMPSO_{FHV}$



(b) $SMPSO$

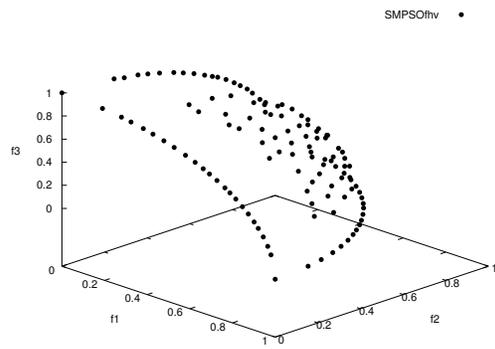


(c) $NSGA-II$

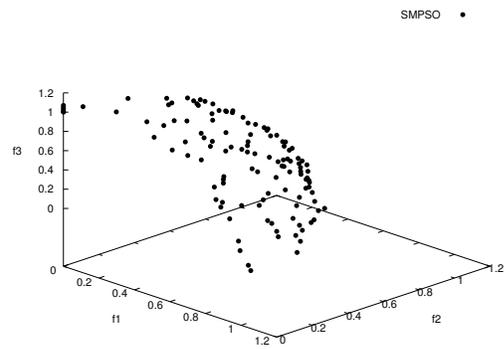


(d) $MOEA/D$

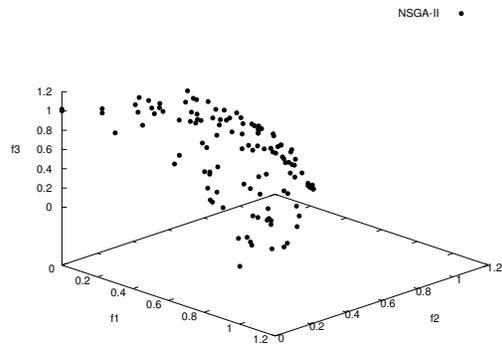
Figura B.8: Problema de prueba DTLZ3 con tres objetivos.



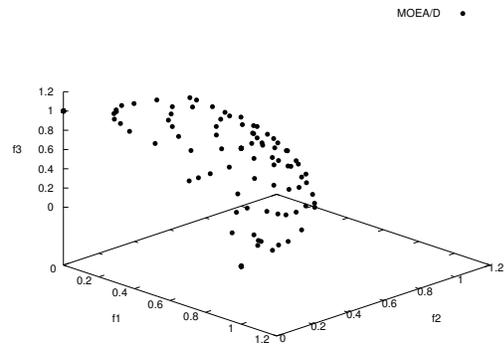
(a) $SMPSO_{FHV}$



(b) $SMPSO$

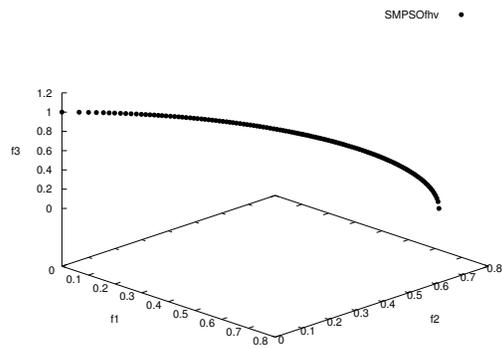


(c) $NSGA-II$

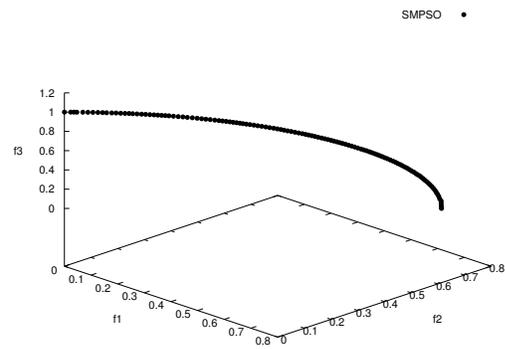


(d) $MOEA/D$

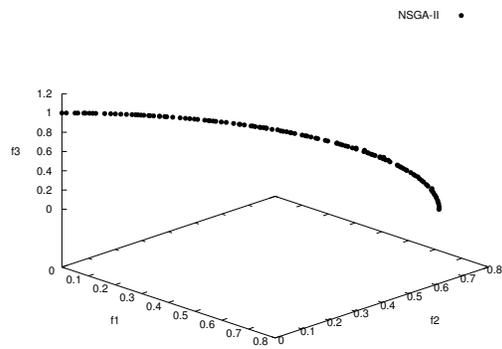
Figura B.9: Problema de prueba DTLZ4 con tres objetivos.



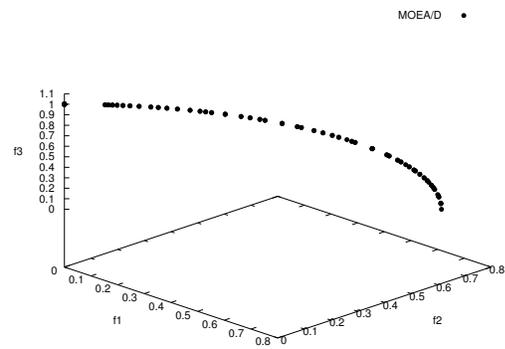
(a) $SMPSO_{FHV}$



(b) $SMPSO$

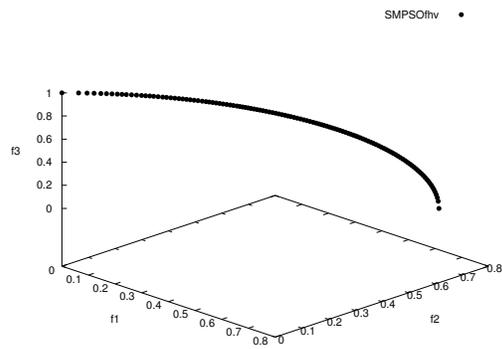


(c) $NSGA-II$

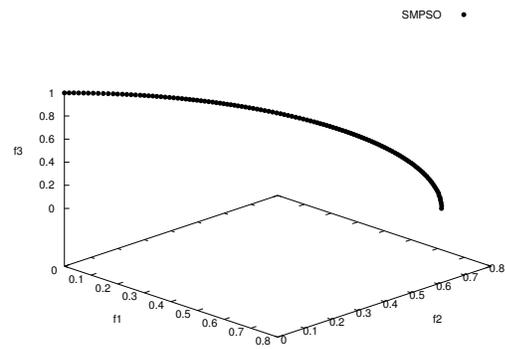


(d) $MOEA/D$

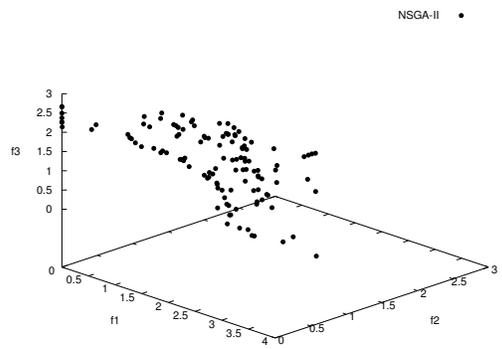
Figura B.10: Problema de prueba DTLZ5 con tres objetivos.



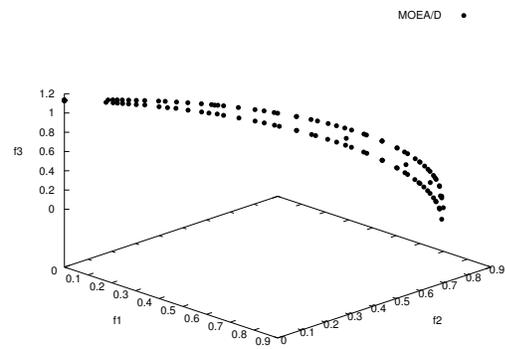
(a) $SMPSO_{FHV}$



(b) $SMPSO$

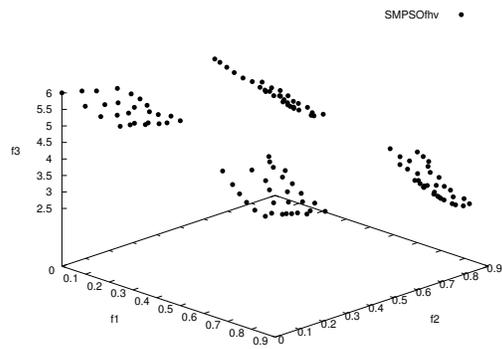


(c) $NSGA-II$

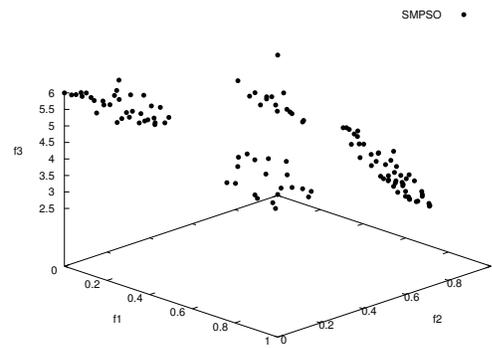


(d) $MOEA/D$

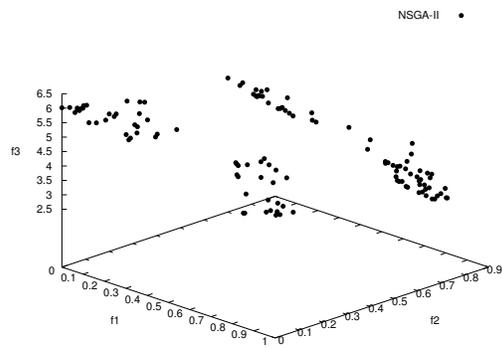
Figura B.11: Problema de prueba DTLZ6 con tres objetivos.



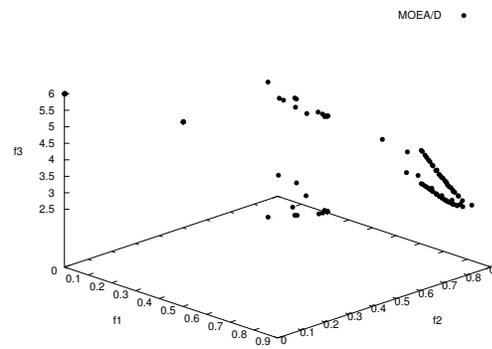
(a) $SMPSO_{FHV}$



(b) $SMPSO$



(c) $NSGA-II$



(d) $MOEA/D$

Figura B.12: Problema de prueba DTLZ7 con tres objetivos.

Bibliografía

- [1] J. Kennedy and R. Eberhart. Particle swarm optimization. In *1995 IEEE Press, IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4, Nov 1995.
- [2] Antonio J. Nebro, Juan J. Durillo, Jose Garcia-Nieto, Carlos A. Coello Coello, Francisco Luna, and E. Alba. SMPSO: A New PSO-based Metaheuristic for Multi-objective Optimization. In *2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM'2009)*, pages 66–73, Nashville, TN, USA, March 30 - April 2 2009. IEEE Press. ISBN 978-1-4244-2764-2.
- [3] Antonio J. Nebro, Juan J. Durillo, and Carlos A. Coello Coello. Analysis of Leader Selection Strategies in a Multi-Objective Particle Swarm Optimizer. In *2013 IEEE Congress on Evolutionary Computation (CEC'2013)*, pages 3153–3160, Cancún, México, 20-23 June 2013. IEEE Press. ISBN 978-1-4799-0454-9.
- [4] Karl Bringmann and Tobias Friedrich. Approximating the volume of unions and intersections of high-dimensional geometric objects. *Computational Geometry-Theory and Applications*, 43(6-7):601–610, August 2010.
- [5] M. Clerc and J. Kennedy. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73, Feb 2002.
- [6] Kalyanmoy Deb and Deb Kalyanmoy. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 2001.

- [7] Joshua Knowles and David Corne. Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors. *IEEE Transactions on Evolutionary Computation*, 7(2):100–116, April 2003.
- [8] Adriana Menchaca-Mendez and Carlos A. Coello Coello. A New Selection Mechanism Based on Hypervolume and its Locality Property. In *2013 IEEE Congress on Evolutionary Computation (CEC'2013)*, pages 924–931, Cancún, México, 20–23 June 2013. IEEE Press. ISBN 978-1-4799-0454-9.
- [9] Anne Auger, Johannes Bader, Dimo Brockhoff, and Eckart Zitzler. Theory of the Hypervolume Indicator: Optimal μ -Distributions and the Choice of the Reference Point. In *FOGA '09: Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms*, pages 87–102, Orlando, Florida, USA, January 2009. ACM.
- [10] Anne Auger, Johannes Bader, and Dimo Brockhoff. Theoretically Investigating Optimal μ -Distributions for the Hypervolume Indicator: First Results for Three Objectives. In Robert Schaefer, Carlos Cotta, Joanna Kołodziej, and Günter Rudolph, editors, *Parallel Problem Solving from Nature—PPSN XI, 11th International Conference, Proceedings, Part I*, pages 586–596. Springer, Lecture Notes in Computer Science Vol. 6238, Kraków, Poland, September 2010.
- [11] 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.
- [12] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.
- [13] 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.

- [14] Eckart Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, November 1999.
- [15] 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.
- [16] 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, Peter R. Lewis, Manuel López-Ibáñez, Gabriela Ochoa, and Ben Paechter, editors, *Parallel Problem Solving from Nature – PPSN XIV, 14th International Conference*, pages 389–398. Springer. Lecture Notes in Computer Science Vol. 9921, Edinburgh, UK, September 17-21 2016. ISBN 978-3-319-45822-9.
- [17] Ivan Chaman García, Carlos A. Coello Coello, and Alfredo Arias-Montaño. MOPSO_{hv}: A New Hypervolume-based Multi-Objective Particle Swarm Optimizer. In *2014 IEEE Congress on Evolutionary Computation (CEC'2014)*, pages 266–273, Beijing, China, 6-11 July 2014. IEEE Press. ISBN 978-1-4799-1488-3.
- [18] 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.
- [19] 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.
- [20] 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 Interna-*

- tional Conference, EMO 2015*, pages 110–125. Springer. Lecture Notes in Computer Science Vol. 9019, Guimarães, Portugal, March 29 - April 1 2015.
- [21] 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.
- [22] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.
- [23] 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.
- [24] Scott D. Chasalow and Richard J. Brand. Algorithm as 299: Generation of simplex lattice points. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 44(4):534–545, 1995.
- [25] A. Jaskiewicz. On the performance of multiple-objective genetic local search on the 0/1 knapsack problem - a comparative experiment. *IEEE Transactions on Evolutionary Computation*, 6(4):402–412, Aug 2002.
- [26] Ram Bhushan Agrawal, K Deb, and RB Agrawal. Simulated binary crossover for continuous search space. *Complex systems*, 9(2):115–148, 1995.
- [27] Kalyanmoy Deb. Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. *Evolutionary Computation*, 7(3):205–230, Fall 1999.