



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS  
DEL INSTITUTO POLITÉCNICO NACIONAL

Unidad Zacatenco  
Departamento de Computación

**Optimización Evolutiva Multiobjetivo basada en el  
Algoritmo de Kuhn-Munkres**

Tesis que presenta

**José Antonio Molinet Berenguer**

para obtener el Grado de

**Maestro en Ciencias en Computación**

Director de la Tesis

**Dr. Carlos Artemio Coello Coello**

México, Distrito Federal

Octubre, 2014



# Resumen

Un gran número de problemas presentes en diversas áreas del conocimiento requieren la optimización simultánea de varios objetivos en conflicto. Estos problemas, denominados Problemas de Optimización Multiobjetivo (POMs), no poseen una solución única, sino un conjunto de soluciones que representan los distintos compromisos entre los objetivos. Los Algoritmos Evolutivos Multiobjetivo (AEMOs) se han convertido en una de las técnicas más utilizadas para lidiar con POMs, pues han mostrado gran efectividad al solucionar problemas con dos y tres objetivos. Sin embargo, a medida que aumenta el número de objetivos en conflicto, el desempeño de la mayoría de los AEMOs se deteriora drásticamente. Esto ha provocado un creciente interés en el desarrollo de AEMOs que puedan solucionar POMs con más de tres objetivos, debido a la frecuencia con que surge este tipo de problemas en la ciencia y la ingeniería.

En esta tesis se propone un nuevo algoritmo evolutivo multiobjetivo para resolver problemas con más de tres objetivos. Nuestra propuesta transforma el proceso de selección de un AEMO en un problema de asignación lineal utilizando un conjunto de vectores de pesos uniformemente distribuidos y una función de costo. El problema de asignación obtenido se soluciona mediante el algoritmo de Kuhn-Munkres (conocido también como *método húngaro*). Para construir el conjunto de vectores de pesos proponemos un algoritmo basado en el *diseño uniforme* que evita las deficiencias del método *simplex-lattice*, el cual es comúnmente utilizado en los AEMOs basados en descomposición. En nuestra propuesta se generan las soluciones utilizando los operadores de recombinación de la evolución diferencial, debido a lo cual denominamos al AEMO propuesto *evolución diferencial húngara* (EDH).

EDH se comparó con respecto a tres de los AEMOs con mejor desempeño registrado en la literatura especializada, dos de ellos basados en descomposición (MOEA/D y MOEA/D-DE) y uno basado en el hipervolumen (SMS-EMOA). Los resultados obtenidos en 16 problemas de prueba con 144 instancias de entre 2 y 10 objetivos, indican que nuestra propuesta supera a MOEA/D y MOEA/D-DE tanto en convergencia como en diversidad de las soluciones. Además, EDH obtiene resultados competitivos con respecto a SMS-EMOA y en varias instancias supera a éste en cuanto a diversidad de las soluciones. Asimismo, el costo computacional de EDH no depende del número de funciones objetivo, mientras que el tiempo de ejecución de SMS-EMOA crece exponencialmente con el número de objetivos.



# Abstract

A large number of problems in a wide variety of domains involve simultaneous optimization of several conflicting objectives. These problems, called Multi-objective Optimization Problems (MOPs), do not have a single optimal solution, but rather a set of solution that represent the best trade-offs among all the objectives. Among the different techniques available to solve MOPs, Multi-objective Evolutionary Algorithms (MOEAs) have become very popular, mainly because of their effectiveness in problems with two and three objectives. However, as the number of conflicting objectives increases, the performance of most MOEAs severely deteriorates. This has motivated a growing interest for developing MOEAs for handling four or more objectives, due to the frequency with such problems arise in science and engineering.

In this work, we propose a new multi-objective evolutionary algorithm to solve optimization problems having four or more objectives. Our proposal transforms the selection process of a MOEA into a linear assignment problem using a set of weight vectors uniformly scattered and a cost function. The Kuhn-Munkres or Hungarian algorithm is used to solve the resulting assignment problem. We propose an algorithm based on *uniform design* to obtain the set of weights and avoid the shortcomings of the *simplex-lattice* design method, which is the approach most commonly adopted in decomposition-based MOEAs. Differential evolution is used as our search engine giving rise to the so-called *hungarian differential evolution* (HDE) algorithm.

Our proposed approach is compared with respect to three well-known MOEAs, two of them based on decomposition (MOEA/D and MOEA/D-DE) and one based on hypervolume (SMS-EMOA). The results obtained in 16 test problems with 144 instances having 2 to 10 objectives, indicate that HDE outperforms MOEA/D and MOEA/D-DE both in terms of convergence and diversity of the solutions obtained. Furthermore, our approach obtains competitive results with respect to SMS-EMOA and, in several instances, outperforms it in terms of the diversity of solutions. Additionally, the computational cost of EDH does not depend on the number of objective functions, whereas SMS-EMOA has an execution time which grows exponentially on the number of objectives.



# Dedicatoria

A mis padres y a mi hermano, por ser el faro que descifra mi destino

A mi novia Carolina, eterna briza encantadora que logra lo mejor de mí

A Luis Manuel, esta también es tu tesis de maestría, seguiremos

A la Dra. Aurora Pons Porrata, ejemplo de mujer investigadora y proa de muchas generaciones.





# Agradecimientos

Esta tesis y el programa de maestría que con ella culmino han sido realizados con el apoyo de una gran familia, que reúne personas excepcionales de Cuba y de México.

A mis padres y a mi hermano agradezco enormemente la confianza y el apoyo que me han brindado durante estos dos años. A pesar de la gran distancia que nos separa, sus palabras y ejemplo iluminaron mis decisiones. Ha sido muy difícil estar tanto tiempo alejado de seres tan especiales en mi vida, espero que el resultado esté a la altura de la nostalgia que han sentido mis padres.

A mi novia Carolina, su hermana y sus padres les estaré eternamente agradecido por considerarme parte de su familia, por dejarme conocer un maravilloso hogar mexicano, por cuidarme como un hijo y protegerme en todo momento. El cariño que me han brindado me unió mucho más con México, su cultura y su gente.

Este trabajo de tesis se derivó del proyecto CONACyT titulado Nuevos Paradigmas Algorítmicos en Optimización Evolutiva Multi-Objetivo (Ref. 221551), cuyo responsable es el Dr. Carlos A. Coello Coello y a quien deseo agradecer la oportunidad de trabajar bajo su supervisión. Su guía, sus comentarios y su apoyo en los momentos más difíciles reflejan su grandeza como investigador y como ser humano.

Agradezco la beca terminal de maestría de un mes otorgada por el Dr. Oliver Schütze, quien además de ser un excelente profesor e investigador, es un ejemplo de valentía y empeño.

Quiero darle las gracias a un gran número de amigos, todos atentos a mis progresos en los estudios y en la salud. A Erick y Day, por mantenerse a mi lado y al de mi familia. A Adanay y Yiny por compartir tantos desvelos en la maestría y tanta alegría de conocer a México. A Napoleón por brindarme su conocimiento y su confianza. A la Sra. Franca por recibirme en su casa y hacerme sentir como en la mía. A Vicente, Esthela, Edgar, Luis, Petlachi e Imanti por darme la bienvenida y conservar la curiosidad. A Mireya, Hugo, Tania, Hegel, Alejandro, Mónica y Ana, quienes junto a todos los demás amigos mencionados, permanecieron a mi lado durante mi cirugía y posterior recuperación. Gracias a todos por ser mi familia en México, por su amistad incondicional y su sonrisa inmensa.

Agradezco a CONACyT la beca de maestría que me otorgó durante dos años para cursar estudios en el CINVESTAV-IPN. Ambas instituciones hacen realidad los sueños de muchos jóvenes mexicanos y extranjeros. El apoyo y consideración del personal que trabaja en el departamento de computación es fundamental para nuestro desempeño. Gracias Sofi por siempre ayudarnos a todos. Gracias a Felipa, a Santiago y todos.



# Índice general

Índice de figuras	XI
Índice de tablas	XIII
<b>1. Introducción</b>	<b>1</b>
1.1. Antecedentes . . . . .	2
1.2. Descripción del problema . . . . .	3
1.3. Objetivos . . . . .	3
1.4. Estructura del documento . . . . .	4
<b>2. Fundamentos Teóricos</b>	<b>5</b>
2.1. Optimización multiobjetivo . . . . .	5
2.1.1. Problemas de prueba . . . . .	7
2.1.2. Indicadores de desempeño . . . . .	9
2.2. Computación evolutiva . . . . .	11
2.2.1. Programación evolutiva . . . . .	13
2.2.2. Estrategias evolutivas . . . . .	13
2.2.3. Algoritmos genéticos . . . . .	14
2.2.4. Ventajas de los algoritmos evolutivos . . . . .	15
2.3. Metaheurísticas Alternativas . . . . .	15
2.3.1. Evolución diferencial . . . . .	17
<b>3. Optimización Evolutiva de Muchos Objetivos</b>	<b>19</b>
3.1. Dificultades en el manejo de muchos objetivos . . . . .	20
3.1.1. Deterioro de la capacidad de búsqueda . . . . .	20
3.1.2. Representación del frente de Pareto . . . . .	22
3.1.3. Visualización del frente de Pareto . . . . .	22
3.2. Algoritmos Evolutivos para la optimización de muchos objetivos . . . . .	23
3.2.1. Relaciones de preferencia alternativas . . . . .	24
3.2.2. Reducción de la dimensión del espacio objetivo . . . . .	26
3.2.3. Basados en información de preferencia . . . . .	28
3.2.4. Basados en indicadores . . . . .	32
3.2.5. Basados en descomposición . . . . .	36

<b>4. Algoritmo Evolutivo Multiobjetivo Basado en el Algoritmo de Kuhn-Munkres</b>	<b>39</b>
4.1. Evolución diferencial húngara . . . . .	39
4.2. Algoritmo de Kuhn-Munkres . . . . .	45
4.3. Generación de vectores de pesos . . . . .	48
<b>5. Estudio Experimental</b>	<b>55</b>
5.1. Diseño experimental . . . . .	55
5.2. Estudio de los componentes del algoritmo propuesto . . . . .	58
5.3. Comparación de la propuesta con otros algoritmos . . . . .	66
<b>6. Conclusiones</b>	<b>79</b>
6.1. Trabajo futuro . . . . .	80
<b>Bibliografía</b>	<b>82</b>

# Índice de figuras

2.1. Características del conjunto de soluciones que aproximan a un frente de Pareto. . . . .	10
4.1. Deficiencia del algoritmo MOEA/D en su proceso de actualización de las soluciones vecinas. . . . .	41
4.2. Proceso de actualización de las soluciones de una vecindad en los algoritmos basados en descomposición. . . . .	42
4.3. Solución de un problema de asignación lineal (PAL). . . . .	46
4.4. Ejemplo de la ejecución del algoritmo de Kuhn-Munkres. . . . .	48
4.5. Deficiencias del método <i>simplex-lattice</i> . . . . .	50
4.6. Vectores de pesos obtenidos en los símplex de 3 y 4 dimensiones por los métodos <i>simplex-lattice</i> , diseño uniforme y Monte Carlo. . . . .	52
5.1. Soluciones obtenidas por las distintas variantes de EDH en el problema DTLZ2 con 6 objetivos. . . . .	58
5.2. Soluciones obtenidas por las distintas variantes de EDH en problemas con 3 objetivos de los conjuntos DTLZ y WFG. . . . .	61
5.3. Soluciones obtenidas por EDH, MOEA/D, MOEA/D-DE y SMS-EMOA en los problemas DTLZ1 a 4 con 3 objetivos. . . . .	68
5.4. Soluciones obtenidas por EDH, MOEA/D, MOEA/D-DE y SMS-EMOA en los problemas DTLZ5 a 7 con 3 objetivos. . . . .	70
5.5. Soluciones obtenidas por EDH, MOEA/D, MOEA/D-DE y SMS-EMOA en los problemas WFG1 a 5 con 3 objetivos. . . . .	74
5.6. Soluciones obtenidas por EDH, MOEA/D, MOEA/D-DE y SMS-EMOA en los problemas WFG6 a 9 con 3 objetivos. . . . .	75
5.7. Tiempo de ejecución promedio considerando 30 corridas independientes de EDH, MOEA/D, MOEA/D-DE y SMS-EMOA en los problemas DTLZ y WFG. . . . .	78



# Índice de tablas

2.1. Propiedades de los problemas de prueba DTLZ y WFG. . . . .	8
4.1. Uniformidad de los conjuntos de vectores de pesos generados con el método <i>simplex-lattice</i> , diseño uniforme y Monte Carlo. . . . .	53
5.1. Puntos de referencia seleccionados para calcular el hipervolumen de las soluciones de cada problema con $m$ objetivos. . . . .	56
5.2. Valores asignados a los parámetros de los AEMOs. . . . .	56
5.3. Tamaño de la población y número máximo de generaciones utilizados en la ejecución de cada algoritmo. . . . .	57
5.4. Hipervolumen obtenido por las variantes de EDH en los problemas DTLZ. Se muestra la media y la desviación estándar de 30 ejecuciones independientes. . . . .	59
5.5. Distancia Generacional Invertida de las variantes de EDH en los problemas DTLZ. Se muestra la media y la desviación estándar de 30 ejecuciones independientes. . . . .	60
5.6. Hipervolumen obtenido por las variantes de EDH en los problemas WFG. Se muestra la media y la desviación estándar de 30 ejecuciones independientes. . . . .	62
5.7. Distancia Generacional Invertida de cada variante de EDH en los problemas WFG. Se muestra la media y la desviación estándar de 30 ejecuciones independientes. . . . .	64
5.8. Hipervolumen obtenido por cada AEMO en los problemas DTLZ. Se muestra la media y la desviación estándar de 30 ejecuciones independientes. . . . .	67
5.9. Distancia Generacional Invertida obtenida por cada AEMO en los problemas DTLZ. Se muestra la media y la desviación estándar de 30 ejecuciones independientes. . . . .	69
5.10. Hipervolumen obtenido por cada AEMO en los problemas WFG. Se muestra la media y la desviación estándar de 30 ejecuciones independientes. . . . .	72
5.11. Distancia Generacional Invertida obtenida por cada AEMO en los problemas WFG. Se muestra la media y la desviación estándar de 30 ejecuciones independientes. . . . .	76





# Capítulo 1

## Introducción

La optimización simultánea de varios objetivos en conflicto es uno de los problemas más frecuentes en la ciencia y la ingeniería [1], al cual se le conoce como Problema de Optimización Multiobjetivo (POM). A diferencia de los problemas donde se optimiza un solo objetivo, los POMs no poseen una solución única, sino un conjunto de soluciones que representan los diferentes compromisos entre los objetivos. La noción de óptimo más comúnmente empleada en este tipo de problemas es la optimalidad de Pareto, la cual considera como soluciones óptimas a las no dominadas. Se dice que una solución domina a otra si no es peor en ninguno de los objetivos y es mejor en al menos uno. El conjunto de soluciones no dominadas se conoce como *conjunto de óptimos de Pareto* y su proyección en el espacio de los objetivos se denomina *frente de Pareto* (FP).

Los Algoritmos Evolutivos Multiobjetivo (AEMOs) se han convertido en una de las técnicas más populares para solucionar POMs, debido fundamentalmente a su flexibilidad y fácil uso [1, 2]. Los AEMOs simulan los principios básicos de la evolución natural planteados en el Neo-Darwinismo. Estos algoritmos aplican sobre una población de individuos (soluciones) operadores de selección, asignación de aptitud, mutación, reproducción y elitismo, con el fin de lograr la mejor aproximación posible del frente de Pareto. Para determinar la calidad de una aproximación del frente de Pareto se consideran dos criterios básicos [3]: el primero es la cercanía entre las soluciones encontradas y el frente de Pareto verdadero; el segundo es la diversidad de las soluciones, que comprende tanto la uniformidad de su distribución como su extensión por todo el frente de Pareto.

Durante varios años, el mecanismo de selección de los AEMOs estuvo mayormente basado en la dominancia de Pareto, pues permite diferenciar soluciones en problemas con dos o tres objetivos. Sin embargo, en años recientes, diversos estudios han comprobado que al incrementarse el número de funciones objetivo a optimizar, resulta ineficaz el uso de la dominancia de Pareto para lograr convergencia y buena distribución de las soluciones [4, 5, 6]. Esto ocurre porque a medida que el número de objetivos aumenta, la proporción de individuos no dominados en la población crece, lo cual deteriora la capacidad de la dominancia de Pareto para discriminar entre soluciones [7].

## 1.1. Antecedentes

El buen desempeño de los algoritmos evolutivos al solucionar problemas de optimización con dos y tres objetivos ha provocado un creciente interés en su aplicación en problemas con cuatro o más objetivos. Para extender el uso de los AEMOs a este tipo de problema ha sido necesario el desarrollo de criterios de selección distintos a la dominancia de Pareto, que puedan ser capaces de diferenciar soluciones con un gran número de objetivos [6, 8].

Dentro de estas nuevas propuestas, los métodos de escalarización se han hecho populares debido al surgimiento del algoritmo evolutivo multiobjetivo basado en descomposición (MOEA/D) [9]. Este algoritmo utiliza una función de escalarización para transformar un POM en varios subproblemas de un solo objetivo, los cuales se optimizan simultáneamente mediante la evolución de una población de soluciones. El buen desempeño de MOEA/D ha generado cada vez más atención y se han propuesto varios AEMOs inspirados en su funcionamiento [10]. Algunas de las propuestas se han enfocado en mejorar la uniformidad de los vectores de pesos utilizados por MOEA/D [11, 12, 13, 14]. Otras emplean funciones de escalarización alternativas para construir los subproblemas [14, 15]. Recientemente, se ha empleado la descomposición en un esquema basado en dominancia de Pareto para diferenciar soluciones no dominadas [16] y en un mecanismo de selección basados en puntos de referencia para guiar la búsqueda en regiones específicas del frente de Pareto [17].

Pero, sin duda, son los AEMOs basados en indicadores los que más han llamado la atención de los investigadores del área [18, 19]. Estos algoritmos utilizan un indicador de calidad para asignar la aptitud de cada individuo en la población, transformando así el problema original en el de maximizar el valor del indicador. Entre los indicadores existentes, el hipervolumen ha sido el más empleado [19], debido principalmente a que es compatible con la dominancia de Pareto y a que se ha podido demostrar que su maximización es equivalente a encontrar el frente de Pareto óptimo [20, 21]. La desventaja de emplear el hipervolumen radica en su alto costo computacional, el cual crece exponencialmente con el número de objetivos [18, 22]. Algunas propuestas han tratado de reducir el costo computacional del cálculo del hipervolumen [22, 23, 24]. Sin embargo, al aumentar el número de objetivos, la calidad de las soluciones obtenidas con este tipo de métodos se ve afectada en gran medida.

En los últimos años, los indicadores de desempeño  $\Delta_p$  [25] y  $R2$  [26] se han utilizado dentro de esquemas de selección de AEMOs como alternativas al hipervolumen [27, 28, 29, 30], debido principalmente a sus bajos costos computacionales.  $\Delta_p$  puede ser visto como la distancia Hausdorff promedio entre un conjunto de soluciones y el frente de Pareto. Este indicador está constituido por la modificación de otros dos indicadores de desempeño: la distancia generacional y la distancia generacional invertida. Por su parte, el indicador  $R2$  fue propuesto inicialmente para comparar aproximaciones del frente de Pareto con base en un conjunto de funciones de utilidad [31]. Su característica más relevante es ser débilmente monótono, además de poseer cierta correlación con el hipervolumen [31].

## 1.2. Descripción del problema

En la actualidad, existe un gran número de problemas con varios objetivos en conflicto que deben ser optimizados. En muchos de ellos, son más de tres las funciones objetivo que deben optimizarse simultáneamente. Sin embargo, la dominancia de Pareto ha demostrado ser insuficiente para diferenciar los individuos en espacios de solución con alta dimensionalidad.

Varios mecanismos de selección han sido desarrollados para enfrentar esta dificultad. Se han propuesto formas relajadas de la dominancia de Pareto, reducir el número de objetivos, enfocarse en áreas específicas del espacio de los objetivos o utilizar funciones de escalarización. Sin embargo, son los AEMOs basados en el hipervolumen los que han mostrado el mejor desempeño. No obstante, los algoritmos basados en el hipervolumen o bien poseen un costo computacional muy elevado, o la calidad de las soluciones que producen se degrada rápidamente según aumenta el número de objetivos. Otros indicadores poseen menor complejidad computacional, pero la calidad de sus soluciones no iguala la obtenida con el hipervolumen. Debido a esto, persiste la necesidad de buscar esquemas de selección escalables con el número de objetivos. Esto es, se requieren mecanismos de selección tales que al aumentar el número de objetivos, tengan un costo computacional bajo y sean capaces de obtener una buena aproximación del frente de Pareto.

## 1.3. Objetivos

En este trabajo se pretende desarrollar un algoritmo evolutivo multiobjetivo capaz de solucionar problemas con más de tres objetivos. Nos enfocaremos principalmente en el mecanismo de selección, pues se desea que este escale correctamente con el número de objetivos. La idea principal radica en transformar el problema de seleccionar los individuos más aptos en un problema de asignación lineal, el cual pueda ser resuelto mediante el algoritmo de Kuhn-Munkres.

### General

Proponer un algoritmo evolutivo multiobjetivo basado en un nuevo esquema de selección que mejore el desempeño de los algoritmos representativos del estado del arte en la optimización de muchas funciones objetivo.

### Particulares

1. Estudio de los algoritmos evolutivos para la optimización de muchos objetivos.
2. Adaptar el problema de la selección de individuos en una población para ser resuelto con el algoritmo de asignación lineal de Kuhn-Munkres.

3. Implementar un método de diseño uniforme para generar conjuntos de vectores de pesos uniformemente espaciados.
4. Implementar un algoritmo evolutivo multiobjetivo con un esquema de selección basado en el algoritmo de Kuhn-Munkres.
5. Comparar el algoritmo propuesto con otros representativos del estado del arte, utilizando diversos problemas de prueba.

### 1.4. Estructura del documento

El presente documento consta de seis capítulos que describen el trabajo realizado y los resultados experimentales obtenidos. A continuación, se describe de forma breve el contenido de los siguientes capítulos.

En el **Capítulo 2**, se especifican formalmente los problemas de optimización multiobjetivo sobre los que trabajaremos, se describen las características principales de la computación evolutiva, así como los distintos paradigmas que la componen. Además, se presentan otras metaheurísticas que han sido empleadas con éxito en la optimización multiobjetivo.

El **Capítulo 3** contiene una revisión de las principales técnicas inspiradas en los algoritmos evolutivos y otras metaheurísticas para solucionar problemas con más de tres funciones objetivo. Estas técnicas se dividen en distintas categorías según las características de su mecanismo de selección.

Nuestra propuesta se introduce en el **Capítulo 4**, donde se describen cada uno de los componentes de nuestro algoritmo, así como los conceptos fundamentales en los que se basa su funcionamiento.

En el **Capítulo 5** se explican los problemas utilizados para evaluar el desempeño de nuestra propuesta con respecto a otros algoritmos evolutivos del estado del arte y se presentan las medidas de desempeño utilizadas en la comparación. Además, se muestran los resultados experimentales obtenidos y se realiza un análisis de sensibilidad de nuestro algoritmo con respecto a cada una de sus componentes.

Finalmente, en el **Capítulo 6**, se discuten los principales resultados obtenidos y su relevancia, así como los trabajos futuros a desarrollar como consecuencia de la investigación presente.

# Capítulo 2

## Fundamentos Teóricos

En este capítulo se presentan los conceptos básicos necesarios para comprender el resto del documento. La sección 2.1 contiene una serie de definiciones matemáticas relacionadas con la optimización multiobjetivo. En esta sección también se describen las principales características de los problemas de prueba utilizados para evaluar los algoritmos de optimización multiobjetivo. Además, se explican las propiedades deseables en el conjunto de soluciones de un problema de optimización multiobjetivo y se especifican algunos de los indicadores más utilizados para medir estas propiedades. En la sección 2.2 se realiza una breve descripción de las características fundamentales de la computación evolutiva y sus paradigmas más representativos. Finalmente, en la sección 2.3 se comentan diversas metaheurísticas recientes que han sido aplicadas en la optimización multiobjetivo.

### 2.1. Optimización multiobjetivo

Un problema de optimización consiste en seleccionar el mejor elemento de un conjunto con base en algún criterio, i.e., encontrar un vector de variables de decisión que satisfaga un conjunto de restricciones y maximice (o minimice) una o más funciones objetivo. Los problemas de optimización que poseen una única función objetivo se denominan problemas de optimización mono-objetivo y aquellos con dos o más funciones objetivo se conocen como problemas de optimización multiobjetivo (POMs). En la mayoría de las ocasiones, las funciones objetivo de un POM se expresan en unidades diferentes y se encuentran en conflicto entre sí. A continuación se presenta una definición formal de los POMs.

**Definición 1 (Problema de optimización multiobjetivo)** Sea  $\Omega$  el espacio de definición de las variables de decisión, encontrar un vector  $\vec{x}^* = [x_1^*, x_2^*, \dots, x_n^*]^T \in \Omega$  que satisfaga las  $m$  restricciones de desigualdad  $g_i(\vec{x}) \leq 0, i = 1, 2, \dots, m$ , las  $p$  restricciones de igualdad  $h_j(\vec{x}) = 0, j = 1, 2, \dots, p$  y maximice (o minimice) el vector de funciones objetivo  $\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x})]^T$ , donde  $\vec{x} = [x_1, x_2, \dots, x_n]^T \in \Omega$ .

**Definición 2 (Región factible)** Es el conjunto  $\mathcal{F}$  de todos los vectores de variables de decisión  $\vec{x} = [x_1, x_2, \dots, x_n]^T \in \Omega$  que satisfacen las  $m$  restricciones de desigualdad  $g_i(\vec{x}) \leq 0, i = 1, 2, \dots, m$  y las  $p$  restricciones de igualdad  $h_j(\vec{x}) = 0, j = 1, 2, \dots, p$ .

En los problemas de optimización mono-objetivo el espacio de definición de la función objetivo posee un orden total, permitiendo la comparación de todo par de valores. Debido a lo anterior, este tipo de problemas tiene generalmente una única solución. Sin embargo, en los POMs los objetivos pueden estar en conflicto, haciendo incomparables a las soluciones. En este caso, la noción de óptimo más aceptada es la propuesta originalmente por Francis Ysidro Edgeworth en 1881 y generalizada posteriormente por Vilfredo Pareto en 1896, más conocida por el nombre de *óptimo de Pareto*. Según este criterio, la solución de un POM no es única, sino un conjunto de soluciones que representa los mejores compromisos posibles entre las funciones objetivo. Este conjunto de soluciones se denomina *conjunto de óptimos de Pareto* y su imagen en el espacio de los objetivos se nombra *frente de Pareto*.

A continuación, se presentarán varias definiciones relacionadas con el concepto de *óptimo de Pareto*, en las cuales se asume (sin perder generalidad) solo problemas de minimización y que el vector de funciones objetivo es  $\vec{f}: \mathbb{R}^n \rightarrow \mathbb{R}^k$ .

**Definición 3 (Dominancia de Pareto)** Dados dos vectores  $\vec{x}, \vec{y} \in \mathbb{R}^k$ , se dice que  $\vec{x}$  **domina a**  $\vec{y}$  (denotado por  $\vec{x} \prec \vec{y}$ ) si y solo si  $x_i \leq y_i, i = 1, \dots, k$  y  $\vec{x} \neq \vec{y}$ .

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

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

**Definición 6 (Conjunto de óptimos de Pareto)** Sea  $\mathcal{F}$  la región factible de un problema de optimización multiobjetivo, el conjunto de óptimos de Pareto  $\mathcal{P}^*$  se define como:

$$\mathcal{P}^* = \{\vec{x} \in \mathcal{F} \mid \vec{x} \text{ es un óptimo de Pareto}\}$$

**Definición 7 (Frente de Pareto)** Sea  $\mathcal{P}^*$  el conjunto de óptimos de Pareto de un problema de optimización multiobjetivo, el frente de Pareto  $\mathcal{PF}^*$  se define como:

$$\mathcal{PF}^* = \{\vec{f}(\vec{x}) \in \mathbb{R}^k \mid \vec{x} \in \mathcal{P}^*\}$$

De acuerdo con las condiciones de Karush-Kuhn-Tucker [32, 33], en los problemas de optimización continuos con  $k$  objetivos, el frente de Pareto es una variedad continua a trozos de dimensión  $k - 1$ . Por ejemplo, en el caso más simple de dos objetivos, el frente de Pareto será una curva (o un conjunto de segmentos de curva). En general, no es fácil encontrar una expresión analítica de la curva, superficie o variedad de mayor

dimensión que corresponda al frente de Pareto de un POM y, de hecho, en la mayoría de los casos resulta imposible. El procedimiento usual para generar el frente de Pareto es encontrar un gran número de soluciones no dominadas y evaluarlas en las funciones objetivo [1]. El frente de Pareto así obtenido estará conformado por un número finito de soluciones computacionalmente representables y a las que denominaremos en esta tesis como *frente de Pareto aproximado*.

### 2.1.1. Problemas de prueba

En la literatura especializada se han propuesto varios problemas que permiten evaluar el desempeño de los algoritmos de optimización multiobjetivo. Entre los más representativos y de mayor uso se encuentran los conjuntos de prueba *Deb-Thiele-Laumanns-Zitzler* (DTLZ) [128] y *Walking-Fish-Group* (WFG) [129]. Estos conjuntos contienen 16 problemas artificiales que son escalables con respecto al número de variables de decisión y al número de objetivos. Además, a estos problemas se les conoce el conjunto de soluciones óptimas de Pareto y la geometría del frente de Pareto. Tanto los problemas DTLZ como los WFG permiten analizar la capacidad que tiene un algoritmo de converger hacia el frente de Pareto y mantener una buena diversidad de las soluciones, al mismo tiempo que enfrenta una serie de dificultades. Estas dificultades están relacionadas con las propiedades de cada problema, tales como la geometría del frente de Pareto, la presencia de varios óptimos locales, la variación de la distribución de las soluciones en el espacio objetivo y la existencia de vectores de variables que corresponden a un mismo vector solución. En la tabla 2.1 se presenta un resumen de estas características para cada problema.

Los problemas clasificados como separables se pueden optimizar considerando una sola variable a la vez, de manera independiente al resto de las variables. Los problemas DTLZ1 a 4 no son estrictamente separables, pues no se puede garantizar que al optimizarlos de esta manera se obtengan todas las soluciones óptimas. En cuanto a los problemas que poseen varios óptimos locales, éstos representan un verdadero reto para los algoritmos de optimización. Por ejemplo, DTLZ1 tiene  $11^k - 1$  frentes de Pareto locales ( $k$  es un parámetro), en los cuales puede quedar atrapado un algoritmo de optimización antes de alcanzar el verdadero frente de Pareto. En el caso de DTLZ3 el número de frentes de Pareto locales es  $3^k - 1$ . Un caso especial dentro de los problemas con múltiples óptimos locales son los problemas deceptivos. Estos problemas están compuestos por una o más funciones objetivo que tienen al menos dos óptimos, el óptimo verdadero y un óptimo falso, siendo este último el más fácil de localizar en el espacio de búsqueda.

En la tabla 2.1 se puede observar la diversidad que presentan los conjuntos de prueba DTLZ y WFG en cuanto a la geometría del frente de Pareto. Algunos de estos frentes de Pareto se consideran mixtos porque tienen secciones de distintas geometrías. En problemas como DTLZ7 y WFG2 el frente de Pareto no es una variedad continua, sino que está constituido por secciones desconectadas entre sí. En el caso de DTLZ7, el número de regiones desconectadas es  $2^{m-1}$ , donde  $m$  es el número de objetivos.

Tabla 2.1: Propiedades de los problemas de prueba DTLZ y WFG.

Problema	Separable	Múltiples óptimos locales	Geometría del frente de Pareto	Distribución de las soluciones
DTLZ1	si*	si	lineal	uniforme
DTLZ2	si*	no	cóncavo	uniforme
DTLZ3	si*	si	cóncavo	uniforme
DTLZ4	si*	no	cóncavo	polinomial
DTLZ5	desconocido	no	degenerado, arco	uniforme
DTLZ6	desconocido	no	degenerado, arco	dependiente de parámetro
DTLZ7	$f_m$ si	$f_{1:m-1}$ no, $f_m$ si	desconectado, mixto	uniforme
WFG1	si	no	$f_{1:m-1}$ convexo, $f_m$ mixto	polinomial, regiones planas
WFG2	no	$f_{1:m-1}$ no, $f_m$ si	desconectado, convexo	uniforme
WFG3	no	no	degenerado, lineal	uniforme
WFG4	si	si	cóncavo	uniforme
WFG5	si	deceptivo	cóncavo	uniforme
WFG6	no	no	cóncavo	uniforme
WFG7	si	no	cóncavo	dependiente de parámetro
WFG8	no	no	cóncavo	dependiente de parámetro
WFG9	no	si, deceptivo	cóncavo	dependiente de parámetro

En DTLZ5, DTLZ6 y WFG3, el frente de Pareto es degenerado, debido a que su dimensión es inferior a  $m - 1$  en un espacio objetivo de dimensión  $m$ .

Una de las propiedades que influye directamente en el proceso de búsqueda de un algoritmo de optimización multiobjetivo es la distribución de las soluciones en el espacio objetivo. Este impacto es más negativo mientras mayor sea el grado de variación de la densidad de las soluciones. En varios de los problemas de la tabla 2.1, las soluciones uniformemente dispersas en el espacio de las variables no se distribuyen de manera uniforme en el espacio objetivo. Un caso especial se presenta en problemas



que poseen un conjunto de vectores de variables a los cuales les corresponde la misma solución en el espacio objetivo; esto se conoce como región plana.

En los problemas DTLZ la relación entre el número de variables ( $n$ ) y el número de objetivos ( $m$ ) está dada por la expresión  $n = m + k - 1$ , donde  $k$  es un parámetro definido por el usuario y determina el grado de dificultad del problema. El dominio de las variables en estos problemas está acotado entre 0 y 1 ( $x_i \in [0, 1], \forall i \in \{1 \dots n\}$ ). El valor de los objetivos en las soluciones del frente de Pareto se encuentra en  $[0, 0.5]$  para DTLZ1 y en  $[0, 1]$  para DTLZ2 a 7. Dado que el valor de todos los objetivos se encuentra en un mismo rango, estos problemas se consideran normalizados. Por el contrario, los problemas WFG tienen un rango distinto para el valor de cada objetivo ( $f_j(\vec{x}) \in [0, 2j], \forall j \in \{1, \dots, m\}$ ) y para el dominio de cada variable ( $x_i \in [0, 2i], \forall i \in \{1 \dots n\}$ ). Además, el número de variables lo determinan los parámetros  $k$  y  $l$  ( $n = k + l$ ), que representan el grado de dificultad del problema en cuanto a convergencia y distribución de las soluciones, respectivamente.

### 2.1.2. Indicadores de desempeño

Cuando se utiliza un algoritmo evolutivo multiobjetivo (AEMO) para solucionar un problema de optimización multiobjetivo se obtiene un conjunto de vectores no dominados entre sí, los cuales se espera que aproximen lo mejor posible el frente de Pareto. Generalmente, para evaluar la calidad de esta aproximación se consideran tres aspectos [3, 130]:

1. Convergencia de las soluciones, de tal forma que la distancia entre el conjunto obtenido y el verdadero frente de Pareto sea mínima.
2. Buena distribución de las soluciones en el espacio objetivo, es decir, una distribución de las soluciones lo más uniforme posible.
3. La extensión del conjunto de soluciones en el espacio objetivo debe cubrir la mayor región posible.

En la figura 2.1.2 (a) se presenta un conjunto de soluciones que cumplen con los criterios 1 y 3, pero no poseen una distribución uniforme. El ejemplo mostrado en (b) posee buena extensión y distribución, pero no una buena convergencia. En (c), los criterios 1 y 2 se satisfacen, siendo la extensión la característica deficiente de este conjunto. La aproximación deseada del frente de Pareto es la mostrada en (d), pues satisface los tres criterios simultáneamente. Para referirse tanto a la extensión como a la buena distribución de un conjunto de soluciones es común utilizar el término diversidad. De esta manera, las características deseadas en una aproximación del frente de Pareto se reducen a dos: la convergencia y la diversidad de las soluciones.

A continuación, se describen dos indicadores de desempeño utilizados frecuentemente para evaluar la calidad de las soluciones obtenidas con los AEMOs y que cuentan con una amplia aceptación en la literatura especializada [3, 19, 25]. Estos indicadores son el Hipervolumen ( $I_{HV}$ ) y la Distancia Generacional Invertida

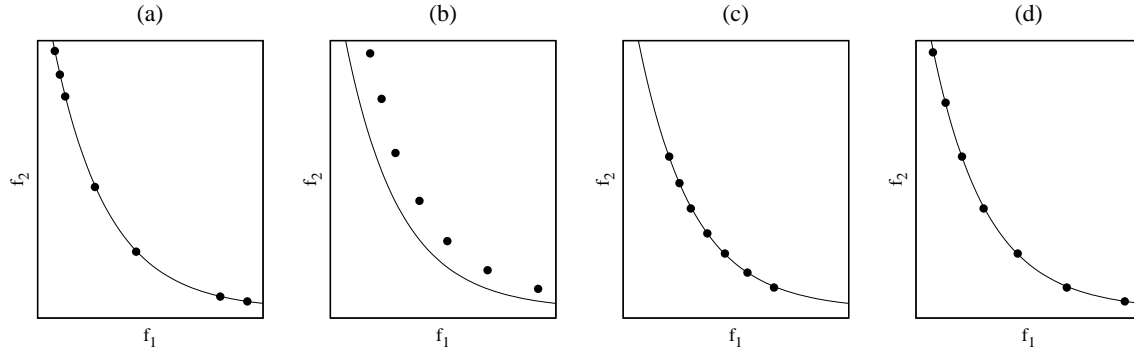


Figura 2.1: Características del conjunto de soluciones que aproximan a un frente de Pareto. En (a) las soluciones no poseen una distribución uniforme, en (b) las soluciones no convergen al frente de Pareto verdadero y en (c) las soluciones no se extienden por todo el frente. Las soluciones presentadas en (d) si poseen al mismo tiempo las tres propiedades deseables en una aproximación del frente de Pareto: distribución uniforme, convergencia y extensión.

( $I_{IGD}$ ). Cada una de estas medidas refleja en un único valor la convergencia y la diversidad de un conjunto de soluciones no dominadas. El hipervolumen es el volumen del espacio objetivo dominado por las soluciones que aproximan el frente de Pareto  $P' = \{\vec{f}(\vec{x}_1^*), \dots, \vec{f}(\vec{x}_N^*)\}$  y acotado por un punto de referencia  $\vec{r}$ , tal que  $\forall i \in [1, N], \vec{f}(\vec{x}_i^*) \prec \vec{r}$ . Por tanto, se puede expresar matemáticamente el hipervolumen como

$$I_{HV}(P', \vec{r}) = \text{volumen} \left( \bigcup_{i=1}^N v_i \right), \quad (2.1)$$

donde  $v_i$  representa el hiper-rectángulo que se forma entre el punto de referencia  $\vec{r}$  y la solución no dominada  $\vec{f}(\vec{x}_i^*)$ . Entre dos conjuntos que aproximen un frente de Pareto, se considera mejor aquel con mayor hipervolumen. Por su parte, el indicador  $I_{IGD}$  es una distancia promedio entre las soluciones aproximadas  $P'$  y el verdadero frente de Pareto, siendo mejor el conjunto de soluciones que posea el menor valor de  $I_{IGD}$ . Sin embargo, para la mayoría de los problemas de prueba no es práctico calcular el frente de Pareto en su totalidad. Debido a esto, el valor de  $I_{IGD}$  se obtiene utilizando un conjunto  $\tilde{P}$  representativo del verdadero frente de Pareto. Es importante resaltar que  $\tilde{P}$  debe cumplir con los tres criterios mencionados al inicio de esta sección. La expresión matemática que define a  $I_{IGD}$  es

$$I_{IGD}(P', \tilde{P}) = \frac{\sum_{\nu \in \tilde{P}} d(\nu, P')}{|\tilde{P}|}, \quad (2.2)$$

donde  $d(\nu, P')$  es la mínima distancia euclidiana entre  $\nu \in \tilde{P}$  y las soluciones en  $P'$ . La precisión con que  $I_{IGD}$  puede medir la uniformidad y la convergencia de las soluciones en  $P'$  depende del conjunto  $\tilde{P}$ . Por tanto,  $\tilde{P}$  debe ser lo suficientemente grande para representar el verdadero frente de Pareto en toda su extensión y con distribución uniforme.

## 2.2. Computación evolutiva

La Computación Evolutiva es la rama de la Inteligencia Artificial que reúne las técnicas inspiradas en los mecanismos de la evolución y la selección natural. A estas técnicas se les conoce como *Algoritmos Evolutivos* (AEs) y simulan los procesos evolutivos enunciados en el Neo-Darwinismo. El término Neo-Darwinismo representa la unión de tres teorías principales: la teoría de la selección natural de Charles Darwin, la genética de Gregor Mendel y la teoría del plasma germinal de August Weismann.

La teoría de la evolución por selección natural planteada por Darwin considera que toda población se compone de individuos ligeramente distintos entre sí y, debido a estas pequeñas variaciones, cada uno tiene distintas capacidades para adaptarse a su medio ambiente. Los individuos con mayor aptitud para adaptarse son seleccionados con mayor probabilidad para reproducirse y transmitir sus rasgos a la próxima generación. De esta forma, las características necesarias para sobrevivir se vuelven más comunes, haciendo que la población, en su conjunto, evolucione.

La genética de Mendel consiste en tres leyes que explican cómo se heredan y expresan en los descendientes las características de sus progenitores. La *Ley de la Segregación* establece que durante la formación de los gametos se separan los dos alelos que codifican cada característica y se combina un alelo materno con uno paterno para formar el par descendiente, asegurando la variación. La *Ley de la Independencia* plantea que diferentes rasgos son heredados independientemente unos de otros y no existe relación entre ellos. Por lo tanto, el patrón de herencia de un rasgo no afectará al patrón de herencia de otro. Por último, la *Ley de la Uniformidad* manifiesta que al cruzar dos individuos con características puras y distintas entre sí, los descendientes de la primera generación serán todos iguales entre sí, e iguales a uno de los progenitores.

La teoría del plasma germinal de August Weismann expresa que los organismos pluricelulares están constituidos por células germinales, encargadas de transmitir la información hereditaria a la descendencia, y células somáticas, responsables de las demás funciones en el organismo. Las células germinales producen células somáticas, pero estas últimas no pueden transmitir información a las células germinales, lo cual se conoce como *la barrera de Weismann*. Por tanto, las habilidades aprendidas por el individuo durante su vida no pueden ser heredadas a la siguiente generación.

Las tres teorías anteriores forman las bases del Neo-Darwinismo, el cual establece a manera de síntesis que la diversidad de la vida en el planeta se puede explicar mediante cuatro procesos estocásticos:

- **Reproducción:** es el proceso mediante el cual se producen nuevos individuos, asegurando la permanencia del material genético de la especie en las futuras generaciones y el surgimiento de nuevas características, debido a la combinación de los individuos progenitores.
- **Mutación:** se produce durante la reproducción, debido a un error en la copia del material genético, alterando así la información transmitida al individuo. Las nuevas características surgidas por la mutación pueden ser heredadas a

la descendencia y en algunas ocasiones, estas características incrementan la aptitud del individuo para adaptarse a su medio ambiente.

- **Competencia:** es la rivalidad entre los individuos que comparten un hábitat, debido a la limitada disponibilidad de los recursos necesarios para sobrevivir.
- **Selección:** es el proceso que otorga mayor oportunidad de reproducirse a los individuos más aptos para sobrevivir, transmitiendo su información genética a una mayor proporción de la población.

La idea de aplicar los procesos de la evolución natural en la computación tiene uno de sus primeros antecedentes en el trabajo de Alan M. Turing de 1950 [34]. Pero no es sino hasta finales de los 1950s cuando se publican una serie de estudios que sirven de inspiración para el posterior desarrollo de lo que hoy se conoce como la Computación Evolutiva. Entre ellos se destacan los del biólogo Alexander S. Fraser [35]. Una de las primeras aplicaciones en la solución de un problema práctico fue el enfoque evolutivo propuesto por George E. P. Box [36] para la optimización de la producción industrial. Desde entonces, la optimización ha sido uno de los objetivos principales de los Algoritmos Evolutivos.

La simulación en una computadora de los procesos evolutivos planteados por el Neo-Darwinismo requiere de cuatro aspectos básicos:

- Codificar las estructuras que se replicarán. Esto se logra utilizando estructuras de datos para representar las posibles soluciones del problema, también conocidas como *individuos*.
- Operaciones que afecten a los individuos. Se deben crear operadores de variación basados en los principios de la reproducción y la mutación, los cuales puedan generar nuevos individuos.
- Una función de aptitud. Esta función se encarga de evaluar la calidad de la solución y, a la vez, permite establecer la competencia entre individuos.
- Un mecanismo de selección. Se basa en la función de aptitud para decidir los individuos con más probabilidad de transmitir sus características.

En sus inicios, desde los años 1960s, los algoritmos inspirados en el Neo-Darwinismo se desarrollaron bajo tres paradigmas principales, que se originaron de manera independiente y con motivaciones distintas. Estos son: *Programación Evolutiva*, *Estrategias Evolutivas* y *Algoritmos Genéticos*. Posteriormente, se agruparon bajo el nombre de Algoritmos Evolutivos y, en la actualidad, los algoritmos propuestos usan indistintamente características de uno y otro paradigma. Los tres paradigmas poseen elementos comunes en su funcionamiento, los cuales se presentan en el Algoritmo 1. En las secciones siguientes se destacan las características específicas de cada tipo de algoritmo evolutivo.

---

**Algoritmo 1** Esquema general de un algoritmo evolutivo

---

- 1: Generar una población inicial con soluciones candidatas aleatorias
  - 2: Calcular la aptitud de cada solución candidata
  - 3: **while** no se satisfaga la condición de terminación **do**
  - 4:   Seleccionar los individuos padres
  - 5:   Aplicar operadores de variación a los padres y obtener los hijos
  - 6:   Calcular la aptitud de cada hijo
  - 7:   Determinar los individuos que pasan a la próxima generación
  - 8: **end while**
- 

### 2.2.1. Programación evolutiva

La Programación Evolutiva (PE) fue propuesta por Lawrence J. Fogel en los años 1960s [37], quien entendía la inteligencia como un comportamiento adaptativo. Según Fogel, el comportamiento inteligente requiere de dos habilidades: 1) la habilidad de un organismo para hacer predicciones correctas dentro de su ambiente y 2) la capacidad de traducir estas predicciones en una respuesta adecuada para una meta dada. Los autómatas de estado finito fueron la representación ideal para modelar este comportamiento.

La PE enfatiza los nexos de comportamiento entre padres e hijos y se inspira en la evolución al nivel de las especies, por lo que no utiliza un operador de recombinación (diferentes especies no se pueden cruzar entre sí). El único operador de variación que incorpora es la mutación. La selección en la PE se realiza normalmente mediante un torneo estocástico, que consiste en comparar un número de individuos entre sí y seleccionar, con determinada probabilidad, el de mayor aptitud.

### 2.2.2. Estrategias evolutivas

Las Estrategias Evolutivas (EEs) tienen su origen en 1964, con los trabajos de los alemanes Peter Bienert, Ingo Rechenberg y Hans-Paul Schwefel [38]. Este modelo fue inicialmente concebido para resolver problemas hidrodinámicos de alto grado de complejidad. La representación de cada individuo fue un vector de valores reales, el cual era manipulado, principalmente, por operadores de mutación que perturbaban los valores reales.

La versión original, denotada por (1+1)-EE, usa un solo padre y produce en cada iteración un solo hijo, mediante un operador de mutación. Si el hijo obtenido tiene mayor aptitud que el padre (i.e., representa una mejor solución) es seleccionado para ser el padre de la próxima generación. En caso contrario, es eliminado y se conserva el mismo padre. Este tipo de selección se llama *extintiva* porque los peores individuos no tienen oportunidad de ser seleccionados.

En la (1+1)-EE, a partir de un padre  $\vec{x}^{(t)} = (x_1^{(t)}, x_2^{(t)}, \dots, x_n^{(t)})$ , se genera al hijo mediante la expresión:

$$x_i^{(t+1)} = x_i^{(t)} + N_i(0, \sigma_i)$$

donde  $t$  es la generación a la que pertenece el individuo padre y corresponde al número de iteraciones realizadas por el algoritmo. El valor de  $N_i(0, \sigma_i)$  es un número aleatorio, generado con una distribución normal de media cero y desviación estándar  $\sigma_i$

En trabajos posteriores, se desarrollaron distintas variantes de la (1+1)-EE. Rechenberg propuso en [39] la estrategia evolutiva  $(\mu+1)$ -EE, en donde los padres se seleccionan de entre  $\mu$  individuos para generar un solo hijo. De esta forma se introduce el concepto de población en las EEs, pues en cada generación se tiene un conjunto de  $\mu$  individuos. En la  $(\mu+1)$ -EE, el hijo reemplaza al peor individuo de la población, lo cual coincide con la selección extintiva de la (1+1)-EE. Sin embargo, es en los trabajos de Schwefel [40] que aparece el uso de múltiples hijos, con las estrategias evolutivas  $(\mu, \lambda)$ -EE y  $(\mu+\lambda)$ -EE. En la primera estrategia se producen  $\lambda$  hijos, de los cuales los  $\mu$  mejores pasan a la siguiente generación como únicos sobrevivientes. En cambio, la estrategia  $(\mu+\lambda)$ -EE selecciona como sobrevivientes a los  $\mu$  mejores individuos de la combinación entre padres e hijos.

Una característica relevante de las estrategias evolutivas, conocida con el nombre de *autoadaptación*, es que evoluciona no solo a las variables del problema, sino también a los parámetros mismos de la técnica (las desviaciones estándar). La adaptación automática de la desviación estándar fue propuesta Schwefel [40] y consiste en representar cada individuo con la tupla  $(\vec{x}^{(t)}, \vec{\sigma}^{(t)})$ .

### 2.2.3. Algoritmos genéticos

Los Algoritmos Genéticos (AGs) fueron propuestos por John H. Holland a principios de los 1960s [41], motivado por su interés en resolver problemas de aprendizaje de máquina, y se han convertido en la técnica evolutiva más popular. El desarrollo de los AGs estuvo influenciado por el estudio formal de los procesos de adaptación natural y el traslado de estos mecanismos a la computación. Por lo cual, en los AGs se utilizan conceptos genéticos que no aparecen en los otros dos paradigmas, tales como: *genotipo* y *fenotipo*. En la evolución natural, el genotipo representa la información genética de un individuo, que ha sido heredada de sus antepasados y puede transmitir a sus descendientes. En cuanto al fenotipo, son las características que se manifiestan en un individuo por la expresión de su genotipo.

Una de las particularidades de los AGs es la codificación del fenotipo (soluciones potenciales del problema) en el genotipo (generalmente una cadena binaria). Por tanto, los individuos se representan con una secuencia de ceros y unos, a la cual se le llama *cromosoma*. A cada posición del cromosoma se le denomina *gene* y al valor dentro de esta posición se le llama *alelo*.

La representación empleada en los AGs hace que los operadores de variación actúen directamente sobre el genotipo, a diferencia de lo que ocurre en las EEs y la PE, donde estos operadores actúan sobre el fenotipo. Además, los algoritmos genéticos reconocen la importancia de la cruce sexual (la cual es su operador principal) sobre la mutación. La cruce sexual utilizada en los AGs tiene su inspiración en las Leyes de Mendel. Este operador de reproducción se aplica sobre dos padres y genera dos

hijos, los cuales poseen genes heredados de ambos padres. Por su parte, el operador de mutación puede modificar uno o más genes de cada cromosoma hijo.

El mecanismo de selección utilizado en los AGs permite elegir a los individuos de acuerdo a su aptitud. Generalmente la selección es proporcional al valor de la aptitud y se aplica de manera estocástica (otorga a los menos aptos cierta probabilidad de reproducirse). Otro mecanismo importante dentro de los AGs es el *elitismo*. Se ha demostrado [42] que los AGs requieren de elitismo, i.e., retener intacto al mejor individuo de cada generación para poder converger a la solución óptima del problema.

#### 2.2.4. Ventajas de los algoritmos evolutivos

El uso de las algoritmos evolutivos para resolver problemas de búsqueda y optimización presenta diversas ventajas, lo que ha motivado su aplicación en diversas áreas [1]. Estas técnicas han mostrado, en varios tipos de problemas, mejores desempeños que las técnicas tradicionales [43]. A continuación, se presentan algunas de estas ventajas.

- Simplicidad conceptual y amplia aplicabilidad.
- Usan una población de soluciones potenciales en vez de un solo individuo, lo cual las hace menos susceptibles a quedar atrapadas en óptimos locales.
- No necesitan conocimientos específicos sobre el problema que intentan resolver.
- Pueden explotar fácilmente las arquitecturas en paralelo.
- Son robustas a los cambios dinámicos y generalmente pueden auto-adaptar sus parámetros.
- Tienen el potencial para incorporar conocimiento sobre el dominio del problema y formar híbridos con otras técnicas de optimización.

### 2.3. Metaheurísticas Alternativas

Los Algoritmos Evolutivos se han utilizado con éxito para resolver un gran número de problemas de búsqueda y optimización, pero no han sido las únicas técnicas desarrolladas para enfrentar esos problemas. Otras metaheurísticas propuestas, en su mayoría en años posteriores al surgimiento de los AEs, se han inspirado en diversos fenómenos que ocurren en la naturaleza. Entre las que han generado mayor atención se encuentran: Recocido Simulado, Búsqueda Tabú, Optimización por Enjambre de Partículas y Evolución Diferencial.

El *Recocido Simulado* (RS) es un algoritmo de búsqueda estocástica basado en la evolución de un sólido desde una temperatura inicial hasta alcanzar su equilibrio térmico. Fue propuesto por Kirkpatrick et al. (1983) [44] y Černý (1985) [45], quienes

de forma independiente observaron la relación entre el proceso de recocido y la optimización combinatoria. En el RS, el estado del sólido es análogo a una solución del problema. En cada iteración se genera una nueva solución mediante cambios locales y es aceptada en función de su calidad y de la temperatura, la cual disminuye al acercarse al equilibrio térmico (equivalente al óptimo). El RS se ha adaptado para problemas de optimización en espacios continuos y con varios objetivos [1, 46].

La *Búsqueda Tabú* (BT) fue propuesta por Fred Glover [47] a mediados de los 1980s y se basa en la información histórica del proceso de búsqueda. La BT posee tres componentes principales que guían la búsqueda. El primero es una memoria de corto plazo que contiene las soluciones exploradas recientemente y las declara tabú para no volver a visitarlas (evita los ciclos). El segundo componente es una memoria que almacena las mejores soluciones encontradas y las utiliza como semillas para intensificar la búsqueda. El último componente es una memoria de largo plazo que almacena las regiones exploradas y se usa para diversificar la búsqueda. La BT se concibió inicialmente para problemas de optimización combinatoria, pero se han propuesto variantes para espacios continuos y problemas multiobjetivo [1, 48].

La *Optimización por Enjambre de Partículas* fue propuesta en 1995 por James Kennedy y Russell Eberhart [49], inspirados en el comportamiento social de las aves durante el vuelo. En esta técnica, la búsqueda se realiza utilizando un enjambre de partículas, cada una de las cuales representa una posible solución del problema. El movimiento a realizar por cada partícula es determinado por su velocidad, una componente social y una componente cognitiva. La velocidad de la partícula indica la dirección actual de su desplazamiento. La componente social se refiere a la mejor solución alcanzada entre todo el enjambre (o parte de él) y la cognitiva a la mejor solución explorada por la partícula. Esta heurística fue desarrollada para la búsqueda en espacios continuos, pero también se ha utilizado con éxito en espacios discretos y problemas multiobjetivo [1, 50].

La *Evolución Diferencial* (ED) fue propuesta a mediados de 1990s por Rainer Storn y Kenneth Price [51] y su origen estuvo motivado por la optimización en espacios continuos. Esta metaheurística evoluciona una población de individuos, sobre los que se aplican operadores de variación. En ese sentido, posee una serie de similitudes con los Algoritmos Evolutivos tradicionales, pero la representación de los individuos y los operadores que los modifican, la distinguen de los demás paradigmas. En la ED, cada una de las posibles soluciones del problema se representa mediante un vector de números reales; de esta forma, el algoritmo opera a nivel de fenotipo. La exploración de nuevas soluciones se realiza perturbando cada vector con la diferencia de otros vectores. Debido a que la evolución diferencial es la heurística utilizada en el algoritmo propuesto en este trabajo, sus características serán explicadas con mayor detalle en la siguiente sección.



### 2.3.1. Evolución diferencial

Los operadores de la evolución diferencial pueden adoptar diferentes estrategias, basándose en la selección del vector que será perturbado, el número de padres a utilizar y el tipo de cruce. Las combinaciones de estos elementos generan distintas variantes de la ED, debido a lo cual, se ha desarrollado una nomenclatura para identificar a cada variante. La más popular se denomina *DE/rand/1/bin*, donde *DE* significa evolución diferencial, *rand* indica que los individuos a utilizar en la mutación son seleccionados de forma aleatoria, *1* es el número de pares de soluciones a emplear en la mutación y *bin* representa el uso de la recombinación binaria. En el Algoritmo 2 se muestra el pseudocódigo de *DE/rand/1/bin*.

---

#### Algoritmo 2 Evolución Diferencial (DE/rand/1/bin)

---

```

1: Generar una población inicial aleatoria:  $P^{(1)} = (\vec{x}_1^{(1)}, \vec{x}_2^{(1)}, \dots, \vec{x}_N^{(1)})$ 
2: Evaluar cada individuo de la población:  $f(\vec{x}_i)$ ,  $i = 1, 2, \dots, N$ 
3: for  $g = 1$  to  $g_{max}$  do
4:   for  $i = 1$  to  $N$  do
5:     Seleccionar de forma aleatoria  $r_1, r_2, r_3 \in [1, 2, \dots, N]$ ,  $r_1 \neq r_2 \neq r_3$ 
6:     Seleccionar de forma aleatoria  $j_{rand} \in [1, 2, \dots, n]$ 
7:     for  $j = 1$  to  $n$  do
8:       if  $rand[0, 1) < CR$  or  $j = j_{rand}$  then
9:          $u_{i,j}^{(g+1)} = x_{r_3,j}^{(g)} + F(x_{r_1,j}^{(g)} - x_{r_2,j}^{(g)})$ 
10:      else
11:         $u_{i,j}^{(g+1)} = x_{i,j}^{(g)}$ 
12:      end if
13:    end for
14:    if  $f(\vec{u}_i^{(g+1)}) \leq f(\vec{x}_i^{(g)})$  then
15:       $\vec{x}_i^{(g+1)} = \vec{u}_i^{(g+1)}$ 
16:    else
17:       $\vec{x}_i^{(g+1)} = \vec{x}_i^{(g)}$ 
18:    end if
19:  end for
20: end for

```

---

En la variante *DE/rand/1/bin*, la población inicial se genera de manera aleatoria, aunque se usan reglas de reparación para asegurar que cada variable pertenezca al dominio. Posteriormente, se selecciona un individuo para reemplazo y tres individuos aleatorios como padres (uno de los cuales es el *padre principal*). Los individuos seleccionados se utilizarán para generar un hijo. Con una probabilidad *CR*, se cambia cada variable del padre principal, de tal forma que al menos una de sus variables sea modificada. El cambio se efectúa agregando al valor de la variable una razón *F* de la diferencia entre los valores de esta variable en los otros dos padres. De esta forma, el hijo tendrá valores del vector perturbado del padre principal y valores del individuo para reemplazo. El hijo sustituye al individuo elegido para reemplazo solo

si representa una mejor solución. De lo contrario, se retiene el vector elegido para reemplazo.

El parámetro  $CR$  controla la influencia de los padres en la descendencia, los valores más altos significan mayor influencia de los padres. Por su parte, el parámetro  $F$  escala la influencia del conjunto de pares de soluciones seleccionados para el cálculo del valor de mutación. Un aumento del tamaño de la población o del número de pares de soluciones involucrados en la mutación puede incrementar la diversidad de las nuevas soluciones y permitir una mayor exploración del espacio. Sin embargo, esto disminuye la probabilidad de encontrar una buena dirección de búsqueda. Por tanto, el balance entre el tamaño de la población y el número de diferencias utilizadas en la mutación determinan la eficiencia del algoritmo.

## Capítulo 3

# Optimización Evolutiva de Muchos Objetivos

Los algoritmos evolutivos se han utilizado con éxito en una amplia variedad de problemas de optimización multiobjetivo. En 1984, David Schaffer [52] propuso el *Vector Evaluated Genetic Algorithm* (VEGA), que es considerado como el primer algoritmo evolutivo multiobjetivo (AEMO) capaz de aproximar varias soluciones del conjunto de óptimos de Pareto en una sola ejecución. Sin embargo, este algoritmo es incapaz de obtener soluciones óptimas en regiones no convexas del frente de Pareto [53]. Con el fin de evitar los problemas presentados por VEGA, Goldberg [43] propuso en 1989 utilizar la dominancia de Pareto para asignar la aptitud de los individuos de una población.

En 1993, Fonseca y Fleming [54] presentaron una variación de la idea propuesta por Goldberg, a la que llamaron *Multiobjective Genetic Algorithm* (MOGA). Éste fue uno de los primeros algoritmos con un esquema de selección basado en la dominancia de Pareto. Posteriormente, se desarrollaron varios AEMOs con selección basada en la dominancia de Pareto, entre los que destacan el *Niched Pareto Genetic Algorithm* (NPGA) [55] y el *Nondominated Sorting Genetic Algorithm* (NSGA) [56]. Estos algoritmos incorporaron en su mecanismo de selección un segundo criterio, el cual estaba generalmente relacionado con el mantenimiento de la diversidad en la población. De esta forma, trataban de evitar la convergencia prematura y, a la vez, generar la mayor porción posible del frente de Pareto.

A finales de los años 1990s se introduce el concepto de elitismo en los AEMOs y se hace popular el uso de poblaciones secundarias. Entre las propuestas más representativas de esos años se encuentran el *Strength Pareto Evolutionary Algorithm* (SPEA) [57], la *Pareto Archived Evolution Strategy* (PAES) [58], el *Pareto Envelope based Selection Algorithm* (PESA) [59] y el *Micro Genetic Algorithm* ( $\mu$ GA) para la optimización multiobjetivo [60]. Posteriormente, se desarrollaron versiones mejoradas de algunos de los AEMOs propuestos hasta ese momento, dando origen a una nueva generación. Entre estos AEMOs se destacaron el NSGA-II [61], el SPEA2 [62] y el  $\mu$ GA2 [63].

Los AEMOs mencionados han mostrado un buen desempeño en problemas de dos y tres objetivos [1, 2]. Sin embargo, estudios recientes han mostrado que la dominancia de Pareto no es efectiva como criterio de selección en problemas de cuatro o más objetivos [4, 6]. Esto ha motivado un gran número de investigaciones para mejorar el rendimiento de los AEMOs en este tipo de problemas. Las principales propuestas se han enfocado en el desarrollo de mecanismos de selección alternativos, la reducción del número de objetivos y la exploración de sólo algunas regiones del espacio de soluciones.

En la primera sección de este capítulo se presentarán los principales desafíos que debe enfrentar un AEMO para solucionar problemas de optimización con más de tres objetivos. A continuación, se describirán las técnicas más representativas que se han propuesto para la optimización de este tipo de problemas, haciendo énfasis en sus ventajas y limitaciones.

### 3.1. Dificultades en el manejo de muchos objetivos

Los problemas de optimización con cuatro o más funciones objetivo presentan una serie de retos que hacen difícil su solución. Algunas de estas dificultades son propias de los mecanismos de selección que poseen los AEMOs, las cuales deterioran su capacidad de búsqueda. Otras son inherentes al problema de optimización, tales como la aproximación del frente de Pareto en su totalidad o la visualización misma de las soluciones.

#### 3.1.1. Deterioro de la capacidad de búsqueda

Desde principios de los 1990s, la dominancia de Pareto se hizo popular como criterio de selección en los algoritmos evolutivos multiobjetivo. Estudios realizados en años posteriores han mostrado que la proporción de soluciones no dominadas en una población aumenta de manera exponencial con el número de objetivos. Esto limita la capacidad de la dominancia de Pareto para discriminar entre soluciones en problemas con muchos objetivos.

En 1978, Bentley et al. [64] determinaron que en un conjunto de  $N$  vectores  $k$ -dimensionales, generados de manera aleatoria, el número esperado de vectores no dominados es de orden  $O(\log^{k-1} N)$ . Sin embargo, no es sino hasta con los trabajos de Deb [2] y Farina et al. [7] que este tipo de análisis se extiende al área de la optimización multiobjetivo. Deb [2] presentó un estudio experimental donde calculaba el número de soluciones no dominadas que existían en poblaciones aleatorias con distintos tamaños y número de objetivos. Los resultados obtenidos mostraron que el aumento del número de objetivos causa un mayor número de soluciones no dominadas en la población. Según [2], esto provoca que la presión de selección disminuya considerablemente, lo cual reduce la posibilidad de aplicar el elitismo y deteriora la capacidad de búsqueda de los AEMOs basados en la dominancia de Pareto.

Por su parte, Farina y Amato [7] proponen la expresión 3.1 para determinar, en un espacio de dimensión  $k$ , la proporción  $e$  de vectores no comparables (según la dominancia de Pareto) con respecto a un vector dado. De acuerdo a 3.1, a medida que aumenta el número de objetivos, la proporción de soluciones no comparables con respecto a una solución determinada crece de manera exponencial. De esta forma, Farina y Amato explican cómo la dominancia de Pareto puede resultar inadecuada para discriminar entre soluciones en espacios de búsqueda con muchos objetivos.

$$e = \frac{2^k - 2}{2^k} \quad (3.1)$$

La expresión 3.1 también representa la forma en que se reduce la proporción de soluciones que pueden dominar a una solución determinada. Debido a esta reducción, la probabilidad de generar descendientes que dominen a sus ancestros disminuye de manera exponencial. Este problema tiene una relación directa con el fenómeno conocido como *resistencia a la dominancia* [65, 66, 67]. Este fenómeno se manifiesta en aquellas soluciones que poseen un valor deficiente en al menos uno de los objetivos, pero en las cuales, en el resto de los objetivos tienen valores cercanos al óptimo. Este tipo de soluciones se encuentran distantes del frente de Pareto, pero generalmente son no dominadas y resulta difícil generar soluciones que las dominen [4].

Cuando la dominancia de Pareto es insuficiente para identificar a los sobrevivientes en una población, el operador de diversidad se convierte en el mecanismo principal dentro del esquema de selección [4]. Esto puede provocar que se deteriore la capacidad de búsqueda de los AEMOs y, en algunos casos, que las soluciones diverjan, alejándose del verdadero frente de Pareto [68]. Varios algoritmos representativos del estado del arte presentan este problema. Por ejemplo, los operadores de diversidad en NSGA-II [61] y SPEA2 [62] otorgan mayor preferencia a las soluciones no dominadas que sean aisladas y fronterizas. Debido a esto, los puntos con mayor diversidad resultan ser las soluciones con peor convergencia [69].

Diversos estudios experimentales [5, 6, 70] han comprobado el pobre desempeño de los AEMOs basados en la dominancia de Pareto cuando optimizan problemas con cuatro o más objetivos. Estos resultados han sido respaldados por el trabajo teórico de Teytau [71], según el cual, comparar las soluciones utilizando sólo la dominancia de Pareto provoca que la velocidad de convergencia de los AEMOs no sea mucho mejor que la de una búsqueda aleatoria. Friedrich et al. [72] también demuestran la deficiencia de la dominancia de Pareto para guiar la búsqueda de los AEMOs en cierto tipo de problemas, aunque sean relativamente sencillos y con pocos objetivos.

Sin embargo, en años recientes, algunos autores han manifestado que aumentar el número de objetivos de un problema no necesariamente lo hace más difícil. Brockhoff et al. [73] realizaron un análisis teórico del tiempo de ejecución de un AEMO en un problema específico y mostraron cómo, al aumentar el número de objetivos, el problema puede volverse más o menos complejo. En tanto, Schütze et al. [74] llevaron a cabo un estudio teórico sobre la influencia del número de objetivos en la convergencia de un AEMO, concluyendo que agregar más objetivos a un problema puede hacerlo más difícil, pero no en una medida significativa.

### 3.1.2. Representación del frente de Pareto

En los problemas de optimización continuos con  $k$  objetivos en conflicto, el frente de Pareto es una variedad de dimensión  $k - 1$  [33]. Debido a esto, el número de soluciones necesarias para aproximar el frente de Pareto aumenta de manera exponencial con el número de objetivos. Formalmente, el número de soluciones requeridas es de orden  $O(kr^{k-1})$ , donde  $r$  es la resolución por cada dimensión [75]. Por ejemplo, cuando se aproxima el FP de un problema de 2 objetivos es común utilizar 100 soluciones ( $r = 50$ ). Sin embargo, para mantener la misma precisión al aproximar el FP de un problema de 5 objetivos serían necesarias más de  $3 \cdot 10^7$  soluciones.

Este crecimiento exponencial del número de soluciones necesarias para aproximar el FP provoca dos dificultades principales. La primera se produce por el aumento del número de evaluaciones de las funciones objetivo, ya que en varios problemas reales estas evaluaciones tienen un elevado costo computacional [1]. En segundo lugar, aunque se pueda obtener una aproximación precisa de todo el FP, el gran número de soluciones haría difícil la selección de un subconjunto por parte del tomador de decisiones. Esto ha motivado el desarrollo de AEMOs basados en preferencias [76], los cuales puedan aproximar regiones específicas del frente de Pareto que sean de interés para el tomador de decisiones.

### 3.1.3. Visualización del frente de Pareto

Los algoritmos evolutivos multiobjetivos por lo general producen un conjunto de soluciones como aproximación del frente de Pareto. Este conjunto es analizado por un tomador de decisiones para seleccionar una solución particular entre las distintas alternativas. La visualización del FP es fundamental para el proceso de toma de decisiones. Sin embargo, a medida que el número de objetivos aumenta, la visualización de las soluciones no dominadas se hace más difícil.

Actualmente, existen diversos métodos para visualizar el conjunto de soluciones de un problema de cuatro o más objetivos, los cuales se pueden agrupar en tres clases principales. En la primera clase, los objetivos se muestran en grupos de dos y tres a la vez. Un ejemplo de estos métodos es la matriz de diagramas de dispersión [2], que visualiza los compromisos entre los distintos pares de objetivos. Sin embargo, mostrar todos los pares de objetivos no resulta práctico cuando el número de objetivos es elevado. Además, puede limitar el análisis de las relaciones existentes entre el conjunto total de objetivos.

En la segunda clase se encuentran los métodos que visualizan todos los objetivos al mismo tiempo. Algunos de los más representativos en esta categoría son: los diagramas de coordenadas paralelas, las gráficas de barras, los diagramas de pétalos y la representación pentagonal [77]. Entre estos métodos, los diagramas de coordenadas paralelas han sido los más populares [8]. Esta técnica de visualización produce un gráfico de dos dimensiones que contiene el nombre de cada objetivo en el eje horizontal y los valores normalizados de cada objetivo en el eje vertical. Recientemente, se ha propuesto utilizar mapas de calor para visualizar el conjunto de soluciones consi-

derando todos los objetivos simultáneamente [78]. Un mapa de calor es una matriz donde cada fila representa una solución, cada columna representa un objetivo y el color de una celda corresponde al valor en ella. Pryke et al. [78] proponen utilizar técnicas de agrupamiento jerárquico para mantener en filas vecinas las soluciones con valores similares en las funciones objetivo. Por su parte, Walker et al. [79] presentaron un método para ordenar las filas utilizando una medida de similitud que se basa en un análisis espectral.

Por último, existen varias técnicas de visualización que reducen el número de objetivos con base en las características del conjunto de soluciones. En [80] se propone un método que consta de dos etapas; primero se utiliza un modelo de red neuronal artificial para proyectar las soluciones en un espacio de menor dimensión y en la segunda etapa se aplica un algoritmo de agrupamiento jerárquico para facilitar el análisis. Una idea distinta fue presentada por Köppen y Yoshida [81], quienes proponen transformar el conjunto de soluciones en un conjunto de Pareto de dos dimensiones, tratando de conservar las relaciones de dominancia.

## 3.2. Algoritmos Evolutivos para la optimización de muchos objetivos

Desde mediados de los años 1990s, los AEMOs basados en la dominancia de Pareto han sido una de las técnicas más utilizadas en la optimización de problemas con dos y tres objetivos [1]. Sin embargo, se ha comprobado que el desempeño de estos algoritmos se deteriora a medida que aumenta el número de objetivos en conflicto. Esto ha motivado el desarrollo de nuevos AEMOs que sean capaces de lidiar con problemas de más de tres objetivos, pues estos problemas son muy frecuentes en la ciencia y la ingeniería [82]. En esta sección se estudiarán algunas de las metodologías más populares que han sido propuestas para mejorar el desempeño de los AEMOs. Por cada metodología (serán cinco en total), se analizarán varios de sus algoritmos más representativos, así como sus ventajas y deficiencias.

En primer lugar, se presentarán los AEMOs basados en relaciones de preferencia, los cuales comparan las soluciones considerando alguna información adicional. Por ejemplo, el número de objetivos en los cuales una solución es mejor que otra [7, 83], la magnitud de la mejora [7, 84] o el número de subespacios en los que una solución permanece no dominada [85]. Este grupo también incluye los algoritmos que usan funciones de pertenencia difusa para representar las preferencias [7]. En segundo lugar, se analizarán las propuestas que reducen la dimensión del espacio objetivo y así evitan las deficiencias de la dominancia de Pareto al comparar las soluciones [86, 87, 88]. A continuación, se estudiarán los AEMOs que utilizan la información sobre el interés del tomador de decisiones por determinadas regiones del frente de Pareto [89, 90, 16]. Posteriormente, se analizarán los AEMOs que evalúan las soluciones utilizando un indicador de desempeño, transformando así el problema original en la optimización de un solo objetivo (el indicador) [91, 18, 30]. Finalmente, se presentarán los AEMOs

basados en descomposición, los cuales dividen el problema original en una colección de funciones escalares que deben ser optimizadas simultáneamente [9, 10, 13].

### 3.2.1. Relaciones de preferencia alternativas

La selección de los individuos que pasarán a la siguiente generación es una etapa fundamental en los algoritmos evolutivos multiobjetivo, pues permite conservar las mejores soluciones encontradas y guiar la búsqueda hacia el frente de Pareto. Un gran número de AEMOs han utilizado la dominancia de Pareto para comparar las soluciones y poder efectuar la selección. Sin embargo, la dominancia de Pareto solo define un orden parcial en el espacio  $k$ -dimensional de los objetivos y a medida que aumenta el valor de  $k$ , disminuye su capacidad de diferenciar entre las distintas soluciones. Debido a esto, se han propuesto modificaciones de la dominancia de Pareto y otras relaciones de preferencia que permiten discriminar entre los individuos de manera más efectiva.

La dominancia- $\alpha$  [65] es una de las primeras variantes de la dominancia de Pareto que fue propuesta para aumentar la presión de selección en un AEMO. La idea principal de la dominancia- $\alpha$  es fijar un límite superior ( $\alpha_{ji}$ ) y un límite inferior ( $1/\alpha_{ij}$ ) para la razón del compromiso entre cada par de objetivos  $f_i$  y  $f_j$ . Según este criterio, sean  $\Delta f_i$  y  $\Delta f_j$  variaciones equivalentes de  $f_i$  y  $f_j$ , se considera que

$$\alpha_{ji} \leq \frac{\Delta f_i}{\Delta f_j} \leq \frac{1}{\alpha_{ij}}, \quad \alpha_{ij}, \alpha_{ji} \geq 0, \quad i, j = 1 \dots k, \quad (3.2)$$

donde  $k$  es el número de funciones objetivo en el problema. La dominancia- $\alpha$  considera la solución  $\vec{x}$  mejor que la solución  $\vec{y}$  (denotado por  $\vec{x} \overset{\alpha}{\prec} \vec{y}$ ) si y sólo si  $g_i(\vec{x}, \vec{y}) \leq 0, i = 1 \dots k$  y  $\vec{g}(\vec{x}, \vec{y}) \neq 0$ , donde

$$g_i(\vec{x}, \vec{y}) = \sum_{j=1}^k \alpha_{ij} (f_j(\vec{x}) - f_j(\vec{y})), \quad \alpha_{ii} = 1, i = 1 \dots k. \quad (3.3)$$

Esta relación de preferencia aumenta la región de dominancia de cada solución, facilitando la convergencia del algoritmo hacia el frente de Pareto. Sin embargo, las soluciones óptimas según la dominancia- $\alpha$  son un subconjunto del conjunto de óptimos de Pareto. Debido a ello, esta relación de preferencia no permite descubrir ciertas regiones del frente de Pareto [65].

Sato et al. [92] mostraron que al utilizar en NSGA-II una modificación de la dominancia de Pareto, en lugar de la dominancia de Pareto estándar, se mejoraba su desempeño en problemas con más de tres objetivos. La relación de preferencia propuesta en [92] es una modificación de la dominancia de Pareto que permite variar el área de dominancia de cada solución. Para controlar el grado de expansión o contracción del área de dominancia, se transforma el valor de cada función objetivo de la



siguiente manera:

$$f'_i(\vec{x}) = \frac{\|\vec{f}(\vec{x})\| \cdot \sin(w_i + s_i \cdot \pi)}{\sin(s_i \cdot \pi)}, \quad i = 1 \dots k, \quad (3.4)$$

donde  $s_i \in [0.25, 0.75]$ ,  $i = 1 \dots k$ , son parámetros definidos por los usuarios y  $w_i$  es el ángulo entre el vector de funciones objetivo  $\vec{f}(\vec{x})$  y el eje de la función objetivo  $f_i$ . Un valor de  $s_i < 0.5$  ( $\forall i = 1 \dots k$ ) aumenta el área de dominancia de las soluciones, lo cual produce mayor presión de selección en la población y posibilita la convergencia del AEMO hacia el frente de Pareto. Esta relación de preferencia no sólo ha mostrado su efectividad al ser utilizada en algoritmos evolutivos [92, 93], sino también en otras metaheurísticas como los algoritmos de optimización por enjambre de partículas [94]. Sin embargo, esta relación de preferencia, al igual que la  $\alpha$ -dominancia, no permite descubrir el frente de Pareto en su totalidad [92, 93].

Farina y Amato [7] proponen la dominancia- $(k - 1)$  como alternativa de la dominancia de Pareto. Esta relación se basa en el número de objetivos para los cuales una solución  $\vec{x}$  es mejor, igual o peor que otra solución  $\vec{y}$ ; estos valores se representan como  $n_b(\vec{x}, \vec{y})$ ,  $n_e(\vec{x}, \vec{y})$  y  $n_w(\vec{x}, \vec{y})$ , respectivamente. Se dice entonces que una solución  $\vec{x}$  es mejor que otra solución  $\vec{y}$  según la dominancia- $(k - 1)$  (denotado por  $\vec{x} \prec_{(k-1)} \vec{y}$ ) si y sólo si

$$n_e(\vec{x}, \vec{y}) < M \quad y \quad n_b(\vec{x}, \vec{y}) \geq \frac{M - n_e}{k + 1}, \quad (3.5)$$

donde  $M$  es el número de objetivos del problema. Cuando  $k = 0$ , esta relación de preferencia es equivalente a la dominancia de Pareto y los distintos valores de  $k$  corresponden a diferentes subconjuntos de las soluciones óptimas de Pareto. En [95] se comprueba la capacidad de esta relación para discriminar entre soluciones no dominadas tanto en problemas discretos como continuos, aumentando de esta forma la presión de selección en los AEMOs. En [7] se propone una segunda relación de dominancia, la cual considera no sólo el número de objetivos para los que una solución es mejor que otra, sino también la diferencia entre esos valores objetivos. Esta relación de preferencia se basa en reglas difusas para establecer la dominancia entre las soluciones.

Otras relaciones de preferencia se diferencian un poco más de la definición estándar de la dominancia de Pareto. Es el caso de la relación *favour* propuesta por Drechsler et al. [83], la cual se basa en el número de objetivos para los que una solución es mejor que otra. Esta relación considera que una solución  $\vec{x}$  es mejor que otra solución  $\vec{y}$  (denotado por  $\vec{x} \prec_{favour} \vec{y}$ ) si y sólo si  $n_e(\vec{x}, \vec{y}) > n_w(\vec{x}, \vec{y})$ . La relación *favour* puede ser vista como un caso particular de la dominancia- $(k - 1)$  donde  $k = 1$ . Una de las principales deficiencias de la relación *favour* es que no puede lidiar con el fenómeno de *resistencia a la dominancia*. Esto provoca que las soluciones se concentren en los extremos del frente de Pareto [93, 96].

Sülflow et al. [84] proponen la relación de preferencia- $\epsilon$  como una extensión de la relación *favour*. La principal diferencia es que considera el número de objetivos

para los cuales una solución excede por un umbral  $\epsilon$  a la otra solución. En caso de empate, la relación *favour* es utilizada para determinar la mejor solución. La relación de preferencia- $\epsilon$  se utilizó en el NSGA-II y mostró mejor desempeño que la relación *favour* [84].

Las relaciones de preferencia mencionadas comparan las soluciones considerando simultáneamente el valor de todos los objetivos. En cambio, di Pierro et al. [85] proponen utilizar subconjuntos de objetivos al comparar las soluciones. La relación presentada en [85] se basa en el concepto de *orden de eficiencia*. Una solución  $\vec{x}$  es eficiente de orden  $r$  si no es dominada por ninguna otra solución considerando todos los  $\binom{k}{r}$  subconjuntos de objetivos, donde  $k$  representa el número total de objetivos. Se considera entonces que el orden de eficiencia de una solución  $\vec{x}$  es el mínimo valor de  $r$  para el cual  $\vec{x}$  es eficiente. Esta relación se incorporó a NSGA-II para diferenciar entre las soluciones no dominadas, lo cual mejoró la convergencia del algoritmo en problemas de hasta ocho objetivos [85]. Sin embargo, esta relación de preferencia causa pérdida de diversidad en las soluciones encontradas.

Bentley y Wakefield [97] proponen las relaciones de preferencia *average ranking* (AR) y *maximum ranking* (MR). AR ordena los valores de cada objetivo independientemente; de esta manera, obtiene, por cada objetivo, la posición que ocupa una solución con respecto a las demás. La posición general de una solución se obtiene al promediar las posiciones asignadas por cada objetivo. En el caso de MR, no se promedian las posiciones, sino que se asigna el mejor valor. La principal deficiencia de MR es que favorece las soluciones extremas, limitando la convergencia de los AEMOs y deteriorando la diversidad de las soluciones [93]. El estudio realizado por Corne y Knowles [96] mostró que la relación AR obtenía mejores resultados que otras relaciones de preferencia.

Diversos estudios [93, 94, 96] han comprobado que las relaciones de preferencia mencionadas anteriormente mejoran la convergencia de los AEMOs cuando rempazan a la dominancia de Pareto. Sin embargo, varias de estas relaciones [65, 92, 7, 84] necesitan parámetros adicionales, lo cual puede limitar su aplicación en ciertos problemas reales. Además, el conjunto de soluciones óptimas que producen estas relaciones de preferencia es un subconjunto del conjunto de óptimos de Pareto. Esto puede provocar que los AEMOs sólo converjan a regiones específicas del frente de Pareto y se pierda diversidad en las soluciones.

### 3.2.2. Reducción de la dimensión del espacio objetivo

Un gran número de los AEMOs propuestos para lidiar con muchos objetivos intentan convertir el problema original en instancias de menor dimensión (número de objetivos) que puedan solucionarse con técnicas ya existentes. Una etapa fundamental de estos AEMOs consiste en determinar el grado de conflicto entre los objetivos y así conservar lo más posible la relación de dominancia inducida por el problema original. Estos algoritmos se pueden agrupar en dos clases principales. La primera reúne los métodos de partición del espacio, los cuales dividen el espacio objetivo del

problema original en varios subespacios disjuntos de menor dimensión que son explorados independientemente. La segunda clase se refiere a los métodos que reducen el número de objetivos del problema. Estos métodos prescinden de aquellos objetivos con el menor grado de conflicto respecto a los demás objetivos.

Aguirre y Tanaka [98] propusieron un esquema general para dividir el espacio objetivo en varios subespacios, cada uno de los cuales es explorado en distintas generaciones del algoritmo evolutivo. Para dividir el espacio objetivo en subespacios de igual tamaño se utilizaron tres estrategias distintas. La primera asigna de manera aleatoria los objetivos que pertenecerán a cada subespacio. La segunda estrategia establece desde un inicio los objetivos de cada subespacio y se mantiene la misma asignación durante todas las generaciones. La última estrategia traslada un objetivo de un subespacio a otro en cada generación. Este esquema fue incorporado en el NSGA-II y se aplicó en problemas de cuatro a diez objetivos. Los resultados obtenidos mostraron que el método propuesto supera al NSGA-II original tanto en la convergencia como en la diversidad de las soluciones. El mejor desempeño se obtuvo al utilizar la estrategia aleatoria, lo que evidencia la importancia de explorar todos los subespacios posibles.

López et al. [88] proponen una modificación del esquema general de Aguirre y Tanaka [98] y además una nueva estrategia para dividir el espacio objetivo. La estrategia propuesta agrupa los objetivos según el grado de conflicto entre ellos e intenta conservar lo más posible la estructura del problema original. Para determinar el grado de conflicto entre los objetivos, se construye una matriz de correlación utilizando el valor de las soluciones encontradas en cada generación. Este método fue incorporado en el NSGA-II y se comparó su efectividad con respecto a otros tres algoritmos: una estrategia de división aleatoria, el método propuesto por Purshouse y Fleming [99] y la versión original del NSGA-II. Los resultados obtenidos en problemas de hasta 15 objetivos mostraron la superioridad de la estrategia propuesta tanto en convergencia como en distribución de las soluciones. La estrategia basada en la información del conflicto entre los objetivos obtuvo mejores resultados que la estrategia aleatoria. Esto demuestra la importancia de utilizar información específica del problema al construir los subespacios.

A diferencia de los métodos anteriores, que utilizan la información de todos los objetivos, otras técnicas intentan eliminar los objetivos que no son fundamentales para describir el frente de Pareto. Saxena et al. [69] proponen un algoritmo para reducir el número de objetivos basado en el análisis de componentes principales. Este método construye, cada cierto número de generaciones, una matriz de correlación utilizando los valores de las soluciones encontradas. Posteriormente, obtiene las componentes principales de esta matriz (calculando los vectores propios) y con esta información determina los objetivos en mayor conflicto. Este método fue incorporado en NSGA-II y en  $\epsilon$ -MOEA [100] para eliminar sucesivamente los objetivos redundantes (los de menor grado de conflicto) según la aproximación del frente de Pareto obtenida. Los resultados experimentales mostraron la efectividad del algoritmo para solucionar problemas con un gran número de objetivos redundantes, superando a otras técnicas

de reducción propuestas en [87]. Sin embargo, el método presenta deficiencias para encontrar la correcta combinación de objetivos en problemas con un frente de Pareto de alta dimensión (con poca redundancia entre los objetivos).

Brockhoff y Zitzler [87] proponen un método para reducir el número de objetivos que se basa en una idea distinta de conflicto. Según [87] un objetivo está en conflicto con el resto si al eliminarlo se modifican las relaciones de dominancias entre las soluciones. Este criterio se utilizó para construir un algoritmo exacto y otro con una estrategia voraz que solucionarían dos problemas ( $\delta$ -MOSS y  $k$ -EMOSS) relacionados con la reducción de objetivos. El primero de estos problemas consiste en encontrar el menor subconjunto de objetivos que conserve (con un error inferior a  $\delta$ ) las relaciones de dominancia. En cambio, el problema  $k$ -EMOSS radica en descubrir el subconjunto de  $k$  objetivos que produzca el menor error posible. Cada uno de los algoritmos propuestos fue incorporado en el IBEA [91], mejorando su desempeño en varios problemas. Sin embargo, estos algoritmos se limitan a la reducción lineal de los objetivos y consideran que las soluciones están igualmente distribuidas en el espacio objetivo [69].

López et al. [101] proponen dos algoritmos que se basan en una técnica de selección de rasgos para solucionar los problemas  $\delta$ -MOSS y  $k$ -EMOSS. Estos algoritmos utilizan las soluciones no dominadas para construir una matriz de correlación que representa el grado de conflicto entre los objetivos. El conflicto entre los objetivos es utilizado como una distancia para construir vecindades de tamaño  $q$  alrededor de cada objetivo. Posteriormente, se selecciona la vecindad más compacta (con la mínima distancia entre los  $q$  vecinos) y se retiene el objetivo central eliminando el resto de los objetivos en la vecindad. De esta forma, se eliminan los objetivos con el menor conflicto hasta que la cantidad total de objetivos sea igual a  $k$  o no existan vecindades con un error inferior a  $\delta$ . Los resultados experimentales mostraron un desempeño competitivo de estos algoritmos con respecto a los propuestos en [87] y al método basado en el análisis de componentes principales [69]. Sin embargo, resulta necesario analizar el papel del parámetro  $q$  en la efectividad de los algoritmos propuestos y la posibilidad de un proceso adaptativo que reduzca el valor de  $q$ .

Las técnicas de partición del espacio objetivo y los métodos para reducir el número de objetivos se han incorporado en el proceso de búsqueda de varios AEMOs existentes, mejorando la efectividad de estos algoritmos en problemas con muchos objetivos. Sin embargo, estas técnicas de reducción asumen la existencia de objetivos redundantes o la independencia entre subconjuntos de objetivos. Cuando el problema de optimización no cumple con estas características, el espacio objetivo no se reduce suficientemente para ayudar a los AEMOs durante la búsqueda o se prescinde de objetivos que son fundamentales para descubrir el frente de Pareto. Debido a esto, el uso de estas técnicas está limitado a ciertos tipos de problemas.

### 3.2.3. Basados en información de preferencia

En determinados problemas de optimización multiobjetivo, el tomador de decisiones (TD) solo está interesado en una región específica del frente de Pareto. Esto ha propiciado el desarrollo de varios AEMOs que utilizan la información de preferencia del TD para guiar la búsqueda hacia la región de interés. El uso de las preferencias del usuario permite evitar dos de las dificultades que se presentan en problemas con más de tres objetivos. En primer lugar, la búsqueda se puede enfocar en la región de interés para el tomador de decisiones y no es necesario aproximar todo el frente de Pareto; de esta manera, se reduce el número de evaluaciones realizadas. En segundo lugar, la información sobre las preferencias del TD permite diferenciar entre soluciones no dominadas durante el proceso de búsqueda. Esto aumenta la presión de selección en la población y ayuda a la convergencia del AEMO.

Según la interacción del tomador de decisiones con el proceso de búsqueda, los AEMOs basados en información de preferencia se pueden clasificar en: métodos a *priori*, métodos a *posteriori* y métodos interactivos. En un método a *priori*, la información de preferencia se conoce antes de iniciar el proceso de optimización. Si el tomador de decisiones está interesado en una única solución en el frente de Pareto, el problema multiobjetivos se puede transformar en una función escalar. Los métodos a *posteriori* usan la información de preferencia después de terminado el proceso de búsqueda. En este caso, el TD selecciona la solución que prefiere de la aproximación del frente de Pareto. En cambio, los métodos interactivos muestran al tomador de decisiones las soluciones obtenidas en cada etapa del proceso de búsqueda; de esta manera, el TD puede comprender mejor el problema y entregar más información sobre las soluciones deseadas. De las tres variantes existentes, los métodos a *priori* y los interactivos son los de más interés para solucionar problemas con muchos objetivos.

La relación de preferencia propuesta por Fonseca y Fleming [54, 102] fue uno de los primeros intentos de incorporar información de preferencia en un AEMO. Esta relación considera tanto los valores del punto de referencia (solución deseada por el TD) como la prioridad que otorga el TD a cada objetivo. Según esta relación, primero se comparan las soluciones teniendo en cuenta el grupo de objetivos con mayor prioridad. Si los valores de los objetivos en ambas soluciones igualan los valores del punto de referencia o difieren de éstos en igual forma, entonces se comparan las soluciones considerando el siguiente grupo de objetivos con mayor prioridad. Este proceso continúa hasta alcanzar el grupo de objetivos con menor prioridad, donde las soluciones son comparadas utilizando la dominancia de Pareto. Una de las deficiencias de esta relación es que depende de la factibilidad del punto de referencia. Si el punto de referencia se encuentra alejado de la región factible del problema, entonces las soluciones sólo serán comparadas en términos de la prioridad de cada objetivo. Además, esta relación no considera el grado de similitud entre una solución y el punto de referencia.

Deb [103] propone una técnica para transformar un problema de programación por metas en un problema de optimización multiobjetivo que es solucionado con un AEMO. En un problema de optimización por metas, el tomador de decisiones indica los valores que desea alcanzar en cada objetivo y estos valores son incorporados

al problema mediante restricciones. La función objetivo se construye de forma tal que se minimice la diferencia entre el valor de los objetivos originales y los valores establecidos por el TD. El método presentado en [103], al igual que la propuesta de Fonseca y Fleming [102], es sensible a la factibilidad del punto de referencia. Si este punto se encuentra en la región factible, puede impedir la generación de una solución mejor. En cambio, si se encuentra muy alejado de la región factible, la información que brinda este punto de referencia no genera efecto alguno en la búsqueda.

Branke et al. [104] proponen un AEMO que incorpora las preferencias del tomador de decisiones mediante el grado de compromiso entre cada par de objetivos. En este método, el TD debe indicar, por cada par de objetivos  $f_i$  y  $f_j$ , el número de unidades en  $f_i$  que corresponden a un cambio de una unidad en  $f_j$  y viceversa. Los autores proponen una relación de preferencia que utiliza esta información para guiar el AEMO hacia regiones específicas del frente de Pareto. El principal inconveniente de esta propuesta es que requiere el grado de compensación entre cada par de objetivos, lo cual es muy difícil obtener en problemas con un gran número de objetivos. Además, este método sólo puede aplicarse en problemas con un frente de Pareto convexo.

Deb y Sundar [105] modificaron el mecanismo de selección en el NSGA-II para dirigir la búsqueda hacia ciertas regiones del frente de Pareto que estén representadas por puntos de referencia. En el algoritmo propuesto (R-NSGA-II) se utiliza un nuevo criterio de selección para discriminar entre soluciones no dominadas. Este operador asigna mayor relevancia a las soluciones no dominadas que estén más cercanas a los puntos de referencia. R-NSGA-II puede obtener varias soluciones no dominadas alrededor de cada punto de referencia. La deficiencia de este algoritmo es que sólo puede garantizar soluciones débilmente no dominadas, sobre todo en problemas con el frente de Pareto discontinuo [106].

Molina et al. [107] modificaron la dominancia de Pareto para incorporarle información de preferencia. La relación obtenida, llamada dominancia- $g$ , clasifica las soluciones en dos tipos. El primer tipo reúne las soluciones que satisfacen todos los valores del punto de referencia o que no satisfacen ninguno. Las demás soluciones forman parte del segundo tipo. Según la dominancia- $g$ , las soluciones del primer tipo son las preferidas. Esto tiene como consecuencia que las soluciones dominadas son preferidas sobre aquellas con mejores valores en alguno de los objetivos. Esta relación puede ser utilizada en cualquier AEMO y permite obtener un conjunto de soluciones alrededor de cada punto de referencia.

Thiele et al. [90] presentaron una variante del algoritmo evolutivo IBEA [91], en la cual incorporaron una función de utilidad para manejar información de preferencia. Este algoritmo pertenece a la clase de métodos interactivos, pues en cada generación le solicita al tomador de decisiones que especifique puntos de referencia. En esta versión de IBEA, la aptitud de cada individuo se obtiene al dividir el valor del indicador (el cual se maximiza) por el valor de la función de utilidad. El valor de esta función se reduce a medida que la solución evaluada se acerca al punto de referencia. De esta manera, entre dos soluciones con igual valor en el indicador, es preferida la solución más cercana al punto de referencia.

Sindhya et al. [108] también proponen un esquema interactivo basado en un algoritmo evolutivo. En este trabajo, el problema original se transforma en un problema mono-objetivo incorporando la información de preferencia mediante una función de utilidad. El vector de pesos y el punto de referencia que utiliza esta función son entregados por el tomador de decisiones en cada generación del algoritmo. La principal limitación de este algoritmo es que obtiene una única solución, lo cual ocurre por la transformación del problema original en un problema mono-objetivo.

Gong et al. [109] modificaron el algoritmo evolutivo basado en descomposición MOEA/D [9] para obtener un método interactivo que incorpore las preferencias del tomador de decisiones. En cada generación del método propuesto, el TD selecciona su solución preferida entre las encontradas en esa etapa. A continuación, todos los vectores de pesos se redistribuyen alrededor de la solución seleccionada y se realiza otra generación del MOEA/D utilizando los nuevos vectores de pesos. El aspecto negativo de esta técnica es que no permite explorar una vez más las regiones que ya han sido desestimadas. Esto puede provocar que la región descubierta del frente de Pareto no sea la deseada por el TD.

Recientemente, Deb y Jain [16] presentaron NSGA-III, que es una versión del NSGA-II basada en preferencias para lidiar con problemas de muchos objetivos. Este algoritmo utiliza un conjunto de puntos de referencia que pueden ser calculados por el propio método o entregados por el tomador de decisiones. Estos puntos de referencia representan las regiones del frente de Pareto que deben ser descubiertas. Para discriminar entre dos soluciones no dominadas reemplazaron el segundo criterio de selección del NSGA-II por una función de utilidad. El valor de esta función indica la relevancia de una solución para aproximar un punto de referencia. La efectividad de este algoritmo fue analizada en varios problemas de hasta 15 objetivos. Los resultados experimentales mostraron que posee un desempeño competitivo con respecto a MOEA/D. Un variante de NSGA-III para solucionar problemas con restricciones fue propuesta en [110].

López y Coello [106] propusieron una relación de preferencia que permite incorporar en un AEMO las preferencias del tomador de decisiones, sin necesidad de modificar la estructura original del algoritmo. Además, basado en esta relación de preferencia, presentan un método interactivo que requiere una cantidad mínima de información del tomador de decisiones. La relación propuesta divide el espacio objetivo en dos subespacios. Las soluciones presentes en el subespacio más cercano al punto de referencia son comparadas utilizando la dominancia de Pareto y el resto de las soluciones son diferenciadas con base en una función de utilidad. Esta relación permite obtener un conjunto de soluciones no dominadas alrededor del punto de referencia. El algoritmo propuesto fue utilizado para solucionar un problema de ingeniería con instancias de hasta seis objetivos. Los resultados experimentales mostraron la efectividad del método para descubrir soluciones en regiones específicas del frente de Pareto.

Una gran parte de las técnicas propuestas para incorporar información de preferencia en los AEMOs ha mostrado su efectividad al lidiar con problemas de cuatro o más objetivos. Sin embargo, estos métodos requieren de la intervención del toma-

dor de decisiones para identificar las regiones de interés. Además, cuando aumenta el número de objetivos, la cantidad de información requerida se hace mayor y en los métodos interactivos la información se le solicita al TD varias veces durante la búsqueda. Otro problema se presenta durante el análisis que debe realizar el TD para preferir una solución (o región del espacio objetivo) en una etapa intermedia de la búsqueda, pues en problemas con más de tres objetivos resulta compleja la visualización de las soluciones. Por último, la mayor parte de los métodos basados en información de preferencia no son capaces de encontrar soluciones en diferentes regiones del frente de Pareto sin tener información previa (brindada por el TD) sobre su localización.

### 3.2.4. Basados en indicadores

Entre las diversas técnicas que se han desarrollado para mejorar el desempeño de los AEMOs en problemas con más de tres objetivos, los esquemas de selección basados en un indicador de desempeño han sido de las más populares [6, 8, 19]. Un indicador de desempeño es una medida cuantitativa de la calidad con que un conjunto de soluciones no dominadas representa el frente de Pareto. Debido a que la efectividad de los AEMOs es generalmente evaluada con estos indicadores [3], cada vez son más las propuestas que transforman el problema multiobjetivo original en el problema de optimizar uno de estos indicadores [6]. Los indicadores  $n$ -arios comparan la calidad relativa de  $n$  conjuntos de soluciones no dominadas. En el caso de un indicador *unario*, su valor representa la calidad del conjunto de soluciones según un criterio específico (puede ser convergencia, diversidad o ambas). El hipervolumen [111] es un indicador *unario* que se ha convertido en la opción más popular de los AEMOs basados en indicadores [8, 19]. La principal ventaja del hipervolumen es su compatibilidad con la dominancia de Pareto [3]. Además, se ha demostrado que maximizar su valor es equivalente a encontrar el frente de Pareto óptimo [20, 21].

Knowles y Corne [112] presentaron uno de los primeros AEMOs que utiliza el hipervolumen en su esquema de selección. Este algoritmo mantiene una población secundaria con las mejores soluciones encontradas durante el proceso de búsqueda. En cada generación, una nueva solución no dominada sustituye a otra de la población secundaria si al agregarla aumenta el hipervolumen del conjunto. Los autores probaron (bajo ciertas condiciones) la convergencia y buena distribución de las soluciones obtenidas con este algoritmo.

Zitzler y Künzli [91] propusieron un esquema general (*Indicator Based Evolutionary Algorithm* IBEA) para incorporar un indicador arbitrario en el mecanismo de selección de los AEMOs. En este esquema, las soluciones son comparadas utilizando un indicador binario que conserve la relación de dominancia. En [91] se presentan dos variantes de IBEA, una basada en el indicador- $\epsilon$  y la otra basada en una versión binaria del hipervolumen. Debido a que en IBEA no se utiliza la dominancia de Pareto, la capacidad de búsqueda no se deteriora con el aumento del número de objetivos. En varios estudios se ha comprobado la efectividad de este esquema para converger hacia el frente de Pareto y al mismo tiempo se ha detectado su deficiencia en cuanto



a la diversidad de las soluciones encontradas [113, 114].

Beume et al. [18] propusieron el algoritmo SMS-EMOA (*S metric selection Evolutionary Multi-Objective Optimization Algorithm*), el cual utiliza el hipervolumen como criterio de selección en una estrategia evolutiva del tipo  $(\mu + 1)$ . Este algoritmo se basa en la propiedad del hipervolumen, según la cual, maximizar este indicador permite descubrir el frente de Pareto óptimo. Para lograr esto, SMS-EMOA combina el ordenamiento de soluciones no dominadas de NSGA-II [61] con el criterio de selección basado en la contribución al hipervolumen propuesto por Knowles y Corne [112].

En el Algoritmo 3 se describen las etapas principales del SMS-EMOA. Este método inicia con una población aleatoria de  $N$  individuos. En cada generación, se obtiene un nuevo individuo aplicando los operadores de variación (los mismos de NSGA-II) sobre la población actual. A continuación, se realiza un ordenamiento de las soluciones utilizando el método propuesto en NSGA-II y se obtienen  $h$  conjuntos  $(R_i, i = 1, \dots, h)$ . En cada conjunto  $R_i$  las soluciones son no dominadas entre sí y las soluciones de  $R_i$  dominan a las de  $R_j$  si  $i < j$ . Finalmente, para cada solución en  $R_h$  se calcula su contribución al hipervolumen y es eliminada la solución que contribuye en menor medida al hipervolumen de  $R_h$ . Este proceso se repite hasta satisfacer la condición de parada.

---

### Algoritmo 3 SMS-EMOA

---

**Input:** POM, criterio de parada y número de individuos ( $N$ )

**Output:** Aproximación del frente de Pareto.

- 1: Generar una población inicial aleatoria:  $P^{(1)} \leftarrow \{\vec{x}_1^{(1)}, \vec{x}_2^{(1)}, \dots, \vec{x}_N^{(1)}\}$
  - 2: Evaluar cada individuo de la población:  $f(\vec{x}_i)$ ,  $i = 1, 2, \dots, N$
  - 3: **while** no se cumple el criterio de parada **do**
  - 4:   Generar hijo  $\vec{x}^{(t+1)}$  de la población  $P^{(t)}$  utilizando los operadores de variación
  - 5:   Evaluar  $f(\vec{x}^{(t+1)})$
  - 6:    $\{R_1, \dots, R_h\} \leftarrow \text{ordenar}(P^{(t)} \cup \{\vec{x}^{(t+1)}\})$  //Ordenamiento del NSGA-II [61]
  - 7:    $\forall \vec{x} \in R_h : r(\vec{x}) \leftarrow C_{HV}(\vec{x}, R_h)$  //Calcular la contribución al Hipervolumen
  - 8:    $\vec{x}_{min} \leftarrow \arg \min_{\vec{x} \in R_h} r(\vec{x})$
  - 9:    $P^{(t+1)} \leftarrow P^{(t)} \setminus \{\vec{x}_{min}\}$
  - 10:    $t \leftarrow t + 1$
  - 11: **end while**
  - 12: **return**  $P^{(t)}$
- 

SMS-EMOA garantiza que el hipervolumen del conjunto de soluciones no disminuya de una generación a otra. Esto evita que las soluciones se alejen del frente de Pareto, a diferencia de lo que ocurre con los operadores de diversidad de otros AEMOs [68]. Sin embargo, el uso del hipervolumen como criterio de selección presenta ciertas dificultades. En primer lugar, este indicador otorga mayor preferencia a las soluciones que estén en regiones convexas [111], lo cual impide obtener la misma distribución en todo el frente de Pareto. En segundo lugar, el cálculo del hipervolumen

requiere que los valores de los objetivos estén normalizados. Además, Knowles y Corne [112] mostraron cómo la variación del punto de referencia puede causar un cambio drástico en el valor del hipervolumen. Esto provoca que las relaciones de orden entre dos conjuntos, según este indicador, dependan del punto de referencia seleccionado. A pesar de los inconvenientes anteriores, los AEMOs basados en hipervolumen han mostrado un desempeño muy competitivo con respecto a los demás técnicas en problemas con dos y tres objetivos. Sin embargo, la mayor dificultad de utilizar este indicador es que su costo computacional crece de manera exponencial con el número de objetivos [22].

La complejidad de orden exponencial de los algoritmos existentes para calcular el hipervolumen ha limitado el uso de este indicador en problemas con más de tres objetivos [8]. Debido a esto, Brockhoff y Zitzler [23] propusieron un esquema general (SIBEA) para combinar las estrategias de reducción del número de objetivos con la búsqueda evolutiva basada en el hipervolumen. En este trabajo se estudia la efectividad de las técnicas de reducción  $\delta$ -MOSS y  $k$ -EMOSS [87] al ser utilizadas en el esquema propuesto para solucionar problemas de hasta nueve objetivos. Los resultados experimentales mostraron que, considerando el mismo tiempo de cómputo, utilizar  $k$ -EMOSS mejora la convergencia de SIBEA. Esto es posible porque se puede calcular un mayor número de veces el valor del hipervolumen cuando disminuye el número de objetivos. Sin embargo, para que sea efectivo este esquema, se debe reducir el problema a menos de seis objetivos.

Otra de las técnicas empleadas para mejorar la eficiencia de los AEMOs basados en el hipervolumen es la aproximación de este indicador utilizando el muestreo de Monte Carlo. Bader y Zitzler [24] proponen el algoritmo HypE (*Hypervolume Estimation Algorithm for Multi-Objective Optimization*), el cual utiliza un número determinado de muestras para aproximar la contribución de cada individuo al hipervolumen de la población. La precisión de esta aproximación se puede aumentar utilizando un mayor número de muestras en el espacio objetivo, pero esto provocaría un aumento en el tiempo de cómputo del algoritmo. Los resultados experimentales presentados en [24] mostraron el desempeño competitivo de HypE con respecto a NSGA-II [61], SPEA2 [62] y IBEA (basado en el indicador- $\epsilon$ ) [91], especialmente en problemas con más de tres objetivos.

Recientemente, los indicadores  $\Delta_p$  [25] y  $R2$  [26] han sido utilizados dentro del mecanismo de selección de varios AEMOs como alternativas al hipervolumen, principalmente porque poseen un bajo costo computacional. El indicador  $\Delta_p$  se puede ver como la distancia de Hausdorff promediada entre un conjunto de soluciones no dominadas y el frente de Pareto óptimo.  $\Delta_p$  está compuesto por las modificaciones de otros dos indicadores: la distancia generacional y la distancia generacional invertida. De esta manera,  $\Delta_p$  puede evaluar simultáneamente la convergencia y la diversidad de las soluciones. Sin embargo, el cálculo de este indicador requiere de un conjunto que sea representativo del frente de Pareto óptimo. Distintas estrategias han sido propuestas para calcular este conjunto de referencia. Por ejemplo, Gerstl et al. [115] realizan una aproximación lineal de las soluciones no dominadas para construir el con-

junto de referencia. Este método se propuso para problemas con dos objetivos y los resultados experimentales mostraron su efectividad para lograr convergencia y buena distribución de las soluciones. Trautmann et al. [28] presentaron una extensión del método anterior para más de tres objetivos. En esta propuesta, primero se proyectan los vectores objetivos de las soluciones no dominadas en un espacio de dos dimensiones y luego se obtiene la aproximación lineal. Sin embargo, este algoritmo posee un alto costo computacional y solo fue aplicado a problemas con tres objetivos. Rudolph et al. [116] proponen utilizar un método para triangular las soluciones no dominadas y obtener de esta forma el conjunto de referencia. Este método es más eficiente que el presentado en [28] y permite obtener soluciones bien distribuidas en todo el frente de Pareto, pero sólo se aplica a problemas de tres objetivos.

Rodríguez y Coello [27] proponen comparar dos soluciones no dominadas utilizando la contribución de cada una al indicador  $\Delta_p$ . El algoritmo propuesto ( $\Delta_p$ -DDE) construye el conjunto de referencia como un forma escalonada de los individuos no dominados. Los resultados experimentales muestran un desempeño competitivo de  $\Delta_p$ -DDE respecto a SMS-EMOA, especialmente en cuanto al tiempo de ejecución. Las principales limitaciones de esta propuesta son la pérdida de diversidad de las soluciones al aumentar el número de objetivos y su dificultad para lidiar con frentes de Pareto discontinuos.

El indicador  $R2$  [26] ha sido otra de las medidas de desempeño que se ha incorporado recientemente en varios AEMOs. Este indicador fue propuesto inicialmente para comparar dos aproximaciones del frente de Pareto con base en un conjunto de funciones de utilidad. Se ha comprobado que este indicador es débilmente monótono y posee cierta correlación con el hipervolumen [31]. Estas propiedades, y el bajo costo computacional que posee, lo convierten en una buena opción para sustituir al hipervolumen durante la solución de problemas con muchos objetivos. Una de los primeros intentos de incorporar  $R2$  en un AEMO fue el algoritmo  $R2$ -EMOA (*R2 Evolutionary Multi-Objective Algorithm*) propuesto por Trautmann et al. [29]. Este algoritmo es una modificación de SMS-EMOA donde se sustituye el hipervolumen por el indicador  $R2$ , lo cual reduce el tiempo de cómputo y permite su aplicación a problemas con más de tres objetivos. Por su parte, Phan y Suzuki [30] proponen  $R2$ -IBEA (*R2 Indicator Based Evolutionary Algorithm*), el cual es una variante de IBEA [91] que incorpora el indicador  $R2$ . Los resultados experimentales obtenidos en [29] y [30] muestran que el uso del indicador  $R2$  como reemplazo del hipervolumen puede lograr resultados competitivos en problemas con más de tres objetivos.

Las propuestas analizadas reflejan el creciente interés de los investigadores del área por los AEMOs basados en indicadores. Principalmente, por la calidad de las soluciones encontradas con los algoritmos que utilizan el hipervolumen. Sin embargo, el costo computacional de este indicador limita su aplicación a problemas con menos de cuatro objetivos. Esto ha motivado el desarrollo de métodos más eficientes para su cálculo y el uso de indicadores alternativos que posean propiedades similares. Ambas tendencias han logrado avances significativos en cuanto al tiempo de cómputo, pero no en igual medida en cuanto a la calidad de las soluciones.

### 3.2.5. Basados en descomposición

Los métodos basados en descomposición transforman un problema de optimización multiobjetivo en un conjunto de subproblemas de optimización mono-objetivo. Aunque estos métodos son muy frecuentes en la programación matemática, no fue hasta después de la propuesta de Zhang y Li [9] que lograron una mayor presencia en la optimización evolutiva. El algoritmo propuesto en [9] (*Multi-objective Evolutionary Algorithm based on Decomposition* MOEA/D) optimiza simultáneamente todos los subproblemas mediante la evolución de una población de soluciones. Cada individuo de la población es asociado con un subproblema y pertenece a una vecindad de soluciones que comparten información durante la búsqueda. El valor del objetivo de cada subproblema se calcula con una función de utilidad que incorpora el valor de todos los objetivos del problema original. En el Algoritmo 4 se presentan las etapas principales de MOEA/D.

---

#### Algoritmo 4 MOEA/D

---

**Input:** POM, criterio de parada, número de individuos ( $N$ ), función de utilidad ( $u$ ), tamaño de la vecindad ( $T$ ), conjunto de vectores de pesos ( $W = \{\vec{w}_i \mid \vec{w}_i \in [0, 1]^k, \sum_{j=1}^k w_i^j = 1, i = 1, \dots, N\}$ )

**Output:** Aproximación del frente de Pareto.

- 1: Generar una población inicial aleatoria:  $P \leftarrow \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N\}$
  - 2: Evaluar cada individuo de la población:  $\vec{f}(\vec{x}_i)$ ,  $i = 1, 2, \dots, N$
  - 3: Crear la población secundaria:  $PS \leftarrow \emptyset$
  - 4: Calcular los  $T$  vecinos de cada vector:  $\vec{w}_i.V \leftarrow \text{vecindad}(W, \vec{w}_i, T)$ ,  $i = 1, \dots, N$
  - 5: Calcular el punto de referencia  $\vec{z}$ :  $z_i \leftarrow \min_{j=1, \dots, N} f_i(\vec{x}_j)$ ,  $i = 1, \dots, k$
  - 6: **while** no se cumpla el criterio de parada **do**
  - 7:   **for**  $i = 1$  **to**  $N$  **do**
  - 8:      $p, q \leftarrow \text{random}(\vec{w}_i.V)$    //Seleccionar dos soluciones de la vecindad
  - 9:      $\vec{y} \leftarrow \text{generar}(\vec{x}_p, \vec{x}_q)$    //Aplicar los operadores de variación de un AG
  - 10:     Evaluar el hijo:  $\vec{f}(\vec{y})$
  - 11:     Actualizar  $\vec{z}$ :  $\forall j = 1, \dots, k$  si  $z_j > f_j(\vec{y})$  entonces  $z_j \leftarrow f_j(\vec{y})$
  - 12:     Actualizar  $\vec{w}_i.V$ :  $\forall j \in \vec{w}_i.V$  si  $u(\vec{f}(\vec{y}) \mid \vec{w}_j, \vec{z}) \leq u(\vec{f}(\vec{x}_j) \mid \vec{w}_j, \vec{z})$  entonces  $\vec{x}_j \leftarrow \vec{y}$
  - 13:     Actualizar  $PS$ : (a) Eliminar de  $PS$  todos los vectores dominados por  $\vec{f}(\vec{y})$   
                           (b) Agregar  $\vec{f}(\vec{y})$  a  $PS$  si ningún vector de  $PS$  domina a  $\vec{f}(\vec{y})$
  - 14:   **end for**
  - 15: **end while**
  - 16: **return**  $PS$
- 

MOEA/D puede descomponer el problema de optimización multiobjetivo utilizando cualquiera de las funciones de utilidad existentes. Entre las distintas posibilidades, la función de Tchebycheff (ver 3.6) ha sido la opción más frecuente, pues se ha comprobado que para toda solución óptima del frente de Pareto ( $\vec{x}^*$ ) existe un vector de

---

**Algoritmo 5** MOEA/D-DE

**Input:** POM, criterio de parada, número de individuos ( $N$ ), función de utilidad ( $u$ ), tamaño de la vecindad ( $T$ ), máximo número de reemplazos ( $n_r$ ), probabilidad de seleccionar los padres en la vecindad ( $\delta$ ), conjunto de vectores de pesos

$$\left( W = \left\{ \vec{w}_i \mid \vec{w}_i \in [0, 1]^k, \sum_{j=1}^k w_i^j = 1, i = 1, \dots, N \right\} \right)$$

**Output:** Aproximación del frente de Pareto.

```

1: Generar una población inicial aleatoria:  $P \leftarrow \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N\}$ 
2: Evaluar cada individuo de la población:  $\vec{f}(\vec{x}_i), i = 1, 2, \dots, N$ 
3: Calcular los  $T$  vecinos de cada vector:  $\vec{w}_i.V \leftarrow \text{vecindad}(W, \vec{w}_i, T), i = 1, \dots, N$ 
4: Calcular el punto de referencia  $\vec{z}$ :  $z_i \leftarrow \min_{j=1, \dots, N} f_i(\vec{x}_j), i = 1, \dots, k$ 
5: while no se cumpla el criterio de parada do
6:   for  $i = 1$  to  $N$  do
7:      $d \leftarrow \text{random}(0, 1)$ 
8:     if  $d < \delta$  then
9:        $Q \leftarrow \vec{w}_i.V$  //Los padres se tomarán de la vecindad
10:    else
11:       $Q \leftarrow \{1, \dots, N\}$  //o de toda la población
12:    end if
13:     $r_1 \leftarrow i$ 
14:     $r_2, r_3 \leftarrow \text{random}(Q)$ 
15:     $\vec{y} \leftarrow \text{generar}(\vec{x}_{r_1}, \vec{x}_{r_2}, \vec{x}_{r_3})$  //Aplicar los operadores de variación de la ED
16:     $\vec{y} \leftarrow \text{mutación}(\vec{y})$  //Aplicar el operador de mutación
17:    Evaluar el hijo:  $\vec{f}(\vec{y})$ 
18:    Actualizar  $\vec{z}$ :  $\forall j = 1, \dots, k$  si  $z_j > f_j(\vec{y})$  entonces  $z_j \leftarrow f_j(\vec{y})$ 
19:     $c = 0$  //Inicia la actualización de las soluciones
20:    while  $c \neq n_r$  and  $Q \neq \emptyset$  do
21:       $j \leftarrow \text{random}(Q)$ 
22:      si  $u(\vec{f}(\vec{y}) \mid \vec{w}_j, \vec{z}) \leq u(\vec{f}(\vec{x}_j) \mid \vec{w}_j, \vec{z})$  entonces  $\vec{x}_j \leftarrow \vec{y}$ 
23:       $c \leftarrow c + 1$  //No se pueden reemplazar más de  $n_r$  soluciones
24:       $Q \leftarrow Q \setminus \{j\}$ 
25:    end while
26:  end for
27: end while
28: return  $P$ 

```

---

pesos  $\vec{w}$  tal que  $\vec{x}^*$  es el óptimo de 3.6. Como se muestra en el Algoritmo 4, MOEA/D puede minimizar simultáneamente la función 3.6 para  $N$  vectores de pesos distintos, lo que permite obtener  $N$  soluciones del frente de Pareto. Durante el proceso evolutivo, la mejor solución encontrada para cada subproblema se conserva en la población secundaria y sólo puede ser sustituida por las nuevas soluciones que provienen de su vecindad. La vecindad se define para cada vector de pesos  $\vec{w}$  como el conjunto de los  $T$  vectores más cercanos a  $\vec{w}$  (considerando la distancia euclidiana).

$$u(\vec{f}(\vec{x}) \mid \vec{w}_j, \vec{z}) = \max_{i=1, \dots, k} \{w_j^i \mid f_i(\vec{x}) - z_i \mid\} \quad (3.6)$$

En un trabajo posterior, Li y Zhang [10] propusieron una versión de MOEA/D basada en la evolución diferencial (MOEA/D-DE). Esta propuesta se distingue en varios aspectos de MOEA/D. En primer lugar, utiliza la mutación polinomial y los operadores de la evolución diferencial; estos últimos poseen un mejor desempeño en la búsqueda que los operadores genéticos. Además, agrega dos parámetros al algoritmo para evitar la pérdida de la diversidad (en el Algoritmo 5 se describen las etapas fundamentales de MOEA/D-DE). La primera medida utilizada en MOEA/D-DE para mantener la diversidad, consiste en permitir, con una baja probabilidad  $(1 - \delta)$ , que los padres de una solución procedan de regiones distintas a la vecindad del subproblema. Una de las características negativas de MOEA/D, es que una solución puede reemplazar a todos sus vecinos. Esto provoca que se pierda la diversidad en la región que la contiene y resulte difícil para los operadores de variación generar una mejor solución. Para evitar esta dificultad, MOEA/D-DE utiliza el parámetro  $n_r$ , el cual establece el número máximo de soluciones que pueden ser reemplazadas por un mismo individuo.

Diversos estudios han mostrado que la efectividad de MOEA/D y MOEA/D-DE, depende en cierta medida de la función de utilidad y de la distribución de los vectores de pesos. Esto ha motivado el desarrollo de nuevas funciones de utilidad que puedan incorporarse en estos algoritmos y de métodos que generen vectores de pesos uniformemente espaciados. Un ejemplo del primer tipo de propuestas se presenta en el trabajo de Ishibuchi et al. [15]. La propuesta realizada en [15] emplea distintas funciones de utilidad al pasar de una generación a otra y entre distintos individuos. Según los autores, de esta manera se puede lidiar con diferentes tipos de frentes de Pareto. Por su parte, Ma et al. [14] proponen utilizar una modificación de la descomposición de Tchebycheff. Esta modificación hace coincidir la dirección de la solución óptima del subproblema y la dirección del vector de pesos correspondiente, lo cual mejora la distribución de las soluciones encontradas. Otro grupo de métodos se ha concentrado en la generación de los vectores de pesos. Jiang et al. [11] proponen un método para adaptar automáticamente los vectores de pesos según las características geométricas del frente de Pareto. Sin embargo, este método solo fue analizado en problemas de dos y tres objetivos y supone ciertas propiedades del frente de Pareto. Recientemente, algunos trabajos se han enfocado en las técnicas de diseño uniforme [12, 13, 14] o el uso del hipervolumen [30], para construir el conjunto de vectores de pesos uniformemente distribuidos.

# Capítulo 4

## Algoritmo Evolutivo Multiobjetivo Basado en el Algoritmo de Kuhn-Munkres

En este capítulo proponemos un algoritmo evolutivo multiobjetivo (AEMO) con un mecanismo de selección que no utiliza la dominancia de Pareto ni un indicador de desempeño. La motivación principal de este AEMO consiste en evitar los problemas de escalabilidad que presentan los esquemas de selección basados en la dominancia de Pareto y el elevado costo computacional en que se incurre al seleccionar los individuos con base en su contribución al hipervolumen. El algoritmo que presentamos, denominado *evolución diferencial húngara* (EDH), utiliza la evolución diferencial para generar las posibles soluciones y transforma el proceso de selección en un problema de asignación lineal, el cual se resuelve con el algoritmo de Kuhn-Munkres, también conocido como método húngaro.

En la primera sección de este capítulo se discuten las deficiencias de algunos algoritmos basados en descomposición que se relacionan con nuestro trabajo y posteriormente se presenta nuestra propuesta. La segunda sección describe el algoritmo de Kuhn-Munkres, utilizado para solucionar los problemas de asignación lineal en los que transformamos el proceso de selección de cada generación. La construcción de estos problemas de asignación requiere de un conjunto de vectores de pesos uniformemente distribuidos en el espacio de los objetivos. El algoritmo más comúnmente utilizado para generar los vectores de pesos es el método *simplex-lattice*. En la última sección se analizan las principales deficiencias de este método y se propone un algoritmo basado en el diseño uniforme.

### 4.1. Evolución diferencial húngara

En el Capítulo 3 se analizaron las principales deficiencias de los AEMOs actuales al lidiar con problemas de optimización de más de tres objetivos. La mayor dificultad que presentan estos algoritmos cuando crece el número de objetivos consiste en

seleccionar los individuos que deben sobrevivir de una generación a otra. Diversos enfoques se han propuesto para enfrentar este problema, siendo los algoritmos basados en descomposición uno de los más efectivos. En este tipo de algoritmos, el problema de optimización multiobjetivo se transforma en un conjunto de subproblemas de optimización escalar, de tal forma que la solución de cada subproblema represente una región distinta del frente de Pareto.

MOEA/D [9] es considerado el principal exponente de los algoritmos basados en descomposición. Este algoritmo evoluciona de forma simultánea un conjunto de individuos que representan las soluciones de cada subproblema de optimización escalar. De esta manera, permite obtener en una única ejecución varias soluciones que aproximen el frente de Pareto. Sin embargo, MOEA/D presenta dos deficiencias fundamentales que atentan contra la diversidad de las soluciones. En primer lugar, sólo se puede obtener una nueva solución al reproducir individuos de una misma vecindad (que procedan de una misma región del espacio de los objetivos). La segunda deficiencia consiste en que una nueva solución con elevada aptitud puede reemplazar a todas las soluciones de su vecindad. En la figura 4.1 se presenta un ejemplo donde una solución sustituye a todos sus vecinos, debido a que posee un mayor valor de la función de utilidad. Para este ejemplo se consideró que la función de utilidad es la descomposición de Tchebycheff modificada propuesta en [16]

$$u(\vec{f}(\vec{x}) \mid \vec{w}_j, \vec{z}) = \max_{i=1, \dots, k} \left\{ \frac{|f_i(\vec{x}) - z_i|}{w_j^i} \right\} \quad (4.1)$$

y que el punto de referencia  $\vec{z}$  es el origen de coordenadas, con lo cual se puede determinar la aptitud de cada individuo  $x_i, i = 1 \dots 5$ . Los vectores de pesos  $w_1 = (0.45, 0.55), w_2 = (0.5, 0.5), w_3 = (0.55, 0.45)$  y  $w_4 = (0.6, 0.4)$  corresponden a distintos subproblemas en los que MOEA/D descompone el problema multiobjetivo original. En una generación determinada del proceso evolutivo,  $x_1 = (0.6, 0.8), x_2 = (0.7, 0.7), x_3 = (0.75, 0.6)$  y  $x_4 = (0.8, 0.5)$  son las mejores soluciones encontradas para  $w_1, w_2, w_3$  y  $w_4$ , respectivamente. Por su parte,  $x_5 = (0.55, 0.5)$  es una solución generada con los individuos de su vecindad  $x_i, i = 1 \dots 4$ . Utilizando la expresión 4.1 se obtiene que

$$\begin{aligned} u(\vec{f}(x_1) \mid w_1) &= 1.33 & u(\vec{f}(x_5) \mid w_1) &= 1.25 \\ u(\vec{f}(x_2) \mid w_2) &= 1.36 & u(\vec{f}(x_5) \mid w_2) &= 1.11 \\ u(\vec{f}(x_3) \mid w_3) &= 1.40 & u(\vec{f}(x_5) \mid w_3) &= 1.10 \\ u(\vec{f}(x_4) \mid w_4) &= 1.45 & u(\vec{f}(x_5) \mid w_4) &= 1.22. \end{aligned}$$

Según estos valores, el individuo  $x_5$  sustituye al resto de los individuos de su vecindad como mejor solución de cada subproblema. Esto provoca que una misma solución se repita varias veces dentro de la población, disminuyendo la diversidad y, por tanto, las posibles regiones a explorar.



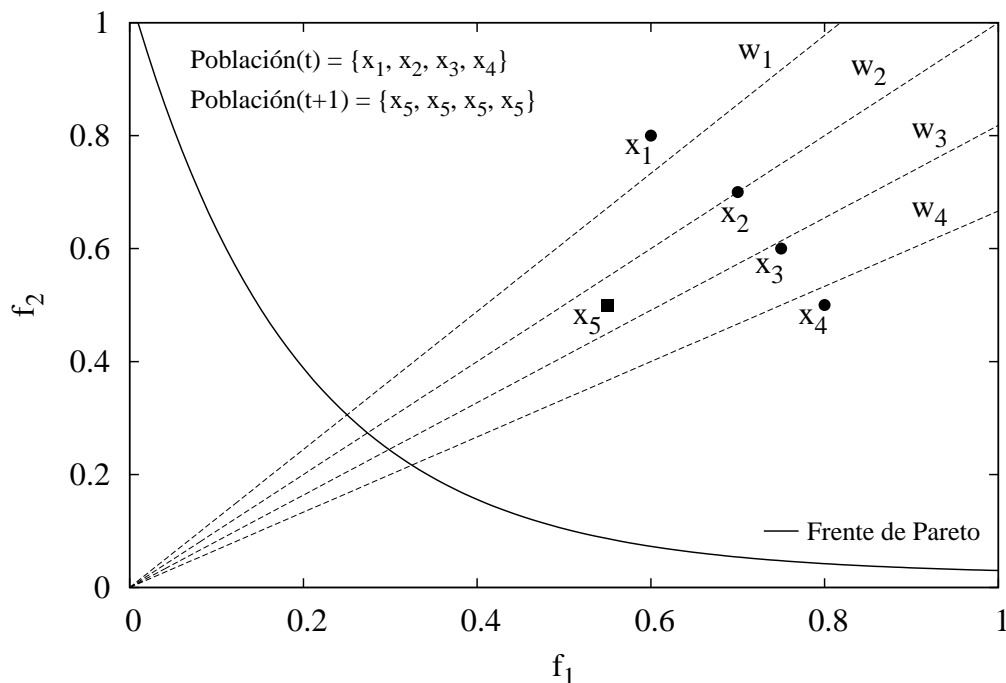


Figura 4.1: Deficiencia del algoritmo MOEA/D en su proceso de actualización de las soluciones vecinas. Para cada vector de pesos  $w_i, i = 1 \dots 4$ , la solución  $x_5$  posee mayor valor de la función de utilidad que las demás soluciones de su vecindad ( $x_i, i = 1 \dots 4$ ). Debido a esto,  $x_5$  pasa a ser la solución de 4 subproblemas y se repite igual número de veces en la población, disminuyendo la diversidad de la misma.

Li y Zhang propusieron en [10] una variante de MOEA/D con el objetivo de evitar las dos deficiencias mencionadas. Esta propuesta, denominada MOEA/D-DE, permite, con cierta probabilidad, que un nuevo individuo sea generado a partir de soluciones de distintas vecindades. Además, limita a un valor predefinido el número de soluciones que pueden ser reemplazadas por un mismo individuo. Sin embargo, tanto MOEA/D como MOEA/D-DE buscan el óptimo de cada subproblema de manera independiente, suponiendo que esto genera el mejor conjunto de soluciones de manera global, lo cual no siempre es correcto, tal y como se muestra en la figura 4.2. Ambos algoritmos generan una nueva solución y buscan en qué subproblema este individuo puede sustituir a su solución actual (ver figura 4.2 (a)), pero no consideran que el individuo reemplazado en un subproblema podría mejorar la solución de otro subproblema (ver figura 4.2 (b)). Esto ocurre porque la asignación de los individuos a cada subproblema se realiza buscando la mejor solución de éste, sin considerar la mejor asignación de manera global.

El algoritmo evolutivo multiobjetivo que se presentará a continuación posee un esquema de selección que evita las deficiencias de MOEA/D y MOEA/D-DE mostradas en las figuras 4.1 y 4.2. Esto lo logra transformando el proceso de selección en un problema de asignación lineal que es solucionado de manera óptima con el algoritmo

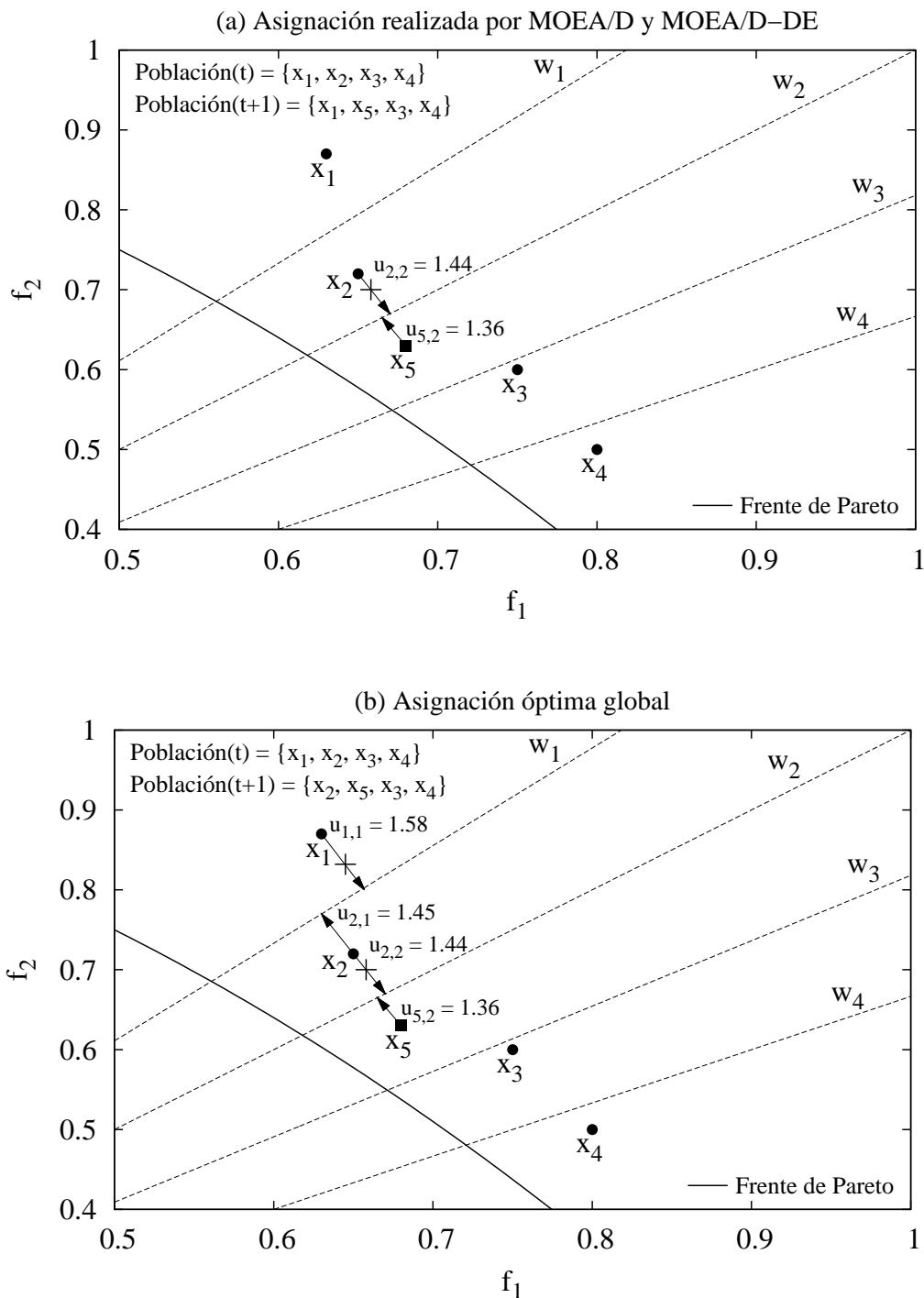


Figura 4.2: Proceso de actualización de las soluciones de una vecindad. En (a) se muestra cómo MOEA/D y MOEA/D-DE asignan el nuevo individuo  $x_5$  al subproblema correspondiente a  $w_2$  y eliminan la solución anterior  $x_2$  sin analizar si puede mejorar la solución de otro subproblema. En (b) no se elimina a  $x_2$ , sino que sustituye a  $x_1$  como solución del subproblema que corresponde a  $w_1$ , lográndose una mejor solución global.

mo de Kuhn-Munkres. El AEMO que proponemos se denomina *evolución diferencial húngara* (EDH), pues adopta los operadores de reproducción de la evolución diferencial para obtener las nuevas soluciones en cada generación y utiliza el método húngaro en su esquema de selección. En el Algoritmo 6 se muestran los pasos principales de EDH.

---

**Algoritmo 6** Evolución diferencial húngara (EDH)

---

**Input:** POM, tamaño de la población ( $n$ ), número máximo de generaciones ( $g_{max}$ ), valores de los parámetros  $C_r$  y  $F$  de la evolución diferencial  $rand/1/bin$

**Output:**  $P_{g_{max}}$  (aproximación de  $\mathcal{P}^*$  y de  $\mathcal{PF}^*$ )

- 1: Generar una población inicial aleatoria:  $P_1 \leftarrow \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n\}$
  - 2: Evaluar cada individuo de la población:  $f(\vec{x}_i)$ ,  $i = 1, 2, \dots, n$
  - 3:  $W \leftarrow$  Conjunto de  $n$  vectores de pesos generados con el Algoritmo 7 (página 51)
  - 4: **for**  $g = 1$  **to**  $g_{max}$  **do**
  - 5:    $P_g^* \leftarrow$  Generar la descendencia de  $P_g$  utilizando los operadores de recombinación de DE/ $rand/1/bin$  (ver Algoritmo 2 en la página 17)
  - 6:   Evaluar cada individuo en  $P_g^*$
  - 7:    $Q_g \leftarrow P_g \cup P_g^*$
  - 8:   Calcular  $\vec{z}^{max}$  y  $\vec{z}^{min}$  mediante (4.2)
  - 9:   Normalizar el valor de los objetivos de cada individuo en  $Q_g$  mediante (4.3)
  - 10:   Generar la matriz de costo  $C$  mediante (4.5) utilizando  $Q_g$  y  $W$
  - 11:    $\mathcal{I} \leftarrow$  Obtener la asignación óptima en  $C$  utilizando el método húngaro presentado en la sección 4.2, en la página 45
  - 12:    $P_{g+1} \leftarrow \{x_i \in Q_g \mid i \in \mathcal{I}, \}$
  - 13: **end for**
- 

En cada generación  $g$  del algoritmo EDH se tiene una población  $P_g$  de  $n$  individuos padres y una población  $P_g^*$  de  $n$  descendientes, obtenidos de  $P_g$  mediante los operadores de recombinación de la evolución diferencial. Estas dos poblaciones se unen para formar el conjunto de soluciones  $Q_g = P_g \cup P_g^*$ . Las  $2n$  soluciones de  $Q_g$  y un conjunto de  $n$  vectores de pesos uniformemente dispersos en el espacio objetivo se utilizan para construir un problema de asignación lineal. Este problema de asignación es equivalente al problema de seleccionar los  $n$  individuos que pasarán a la siguiente generación y se define como: Se tienen  $2n$  soluciones y  $n$  vectores bien distribuidos en el simplex unitario de dimensión  $k - 1$  del espacio objetivo; además, existe un costo por representar cada uno de estos vectores con alguna de las soluciones que conforman la actual aproximación del frente de Pareto; se requiere describir las regiones cubiertas por los  $n$  vectores de pesos utilizando solamente  $n$  individuos, de tal forma que el costo total sea mínimo. La etapa fundamental de la construcción de este problema de asignación consiste en generar una función de costo, de tal forma que al minimizar el costo total de las asignaciones se retengan las soluciones que sean una buena aproximación del frente de Pareto. A continuación, se describe el procedimiento empleado por EDH para construir este problema de asignación lineal.

En primer lugar, el valor de cada objetivo en las  $2n$  soluciones de  $Q_g$  es normalizado, para reducir el espacio objetivo a un hipercubo unitario y así lidiar con funciones objetivo de valores no comparables. Para ello, se construye el vector de valores máximos  $\vec{z}^{max}$  y el vector de valores mínimos  $\vec{z}^{min}$  como

$$\begin{aligned} \vec{z}^{max} &= [z_1^{max}, \dots, z_k^{max}]^T, & z_i^{max} &= \max_{j=1, \dots, 2n} f_i(\vec{x}_j), & i &= 1, \dots, k, \\ \vec{z}^{min} &= [z_1^{min}, \dots, z_k^{min}]^T, & z_i^{min} &= \min_{j=1, \dots, 2n} f_i(\vec{x}_j), & i &= 1, \dots, k, \end{aligned} \quad (4.2)$$

donde  $f_i(\vec{x}_j)$  es el valor del objetivo  $i$ -ésimo en el individuo  $\vec{x}_j \in Q_g$ . El valor normalizado de  $f_i(\vec{x}_j)$  se calcula como

$$\tilde{f}_i(\vec{x}_j) = \frac{f_i(\vec{x}_j) - z_i^{min}}{z_i^{max} - z_i^{min}}, \quad j = 1, \dots, 2n, \quad i = 1, \dots, k. \quad (4.3)$$

Con los valores normalizados de las funciones objetivo y el conjunto

$$W \subset \mathbb{W} = \{\vec{w} \mid \vec{w} \in [0, 1]^k, \sum_{i=1}^k w_i = 1\}, \quad |W| = n, \quad (4.4)$$

de  $n$  vectores de pesos uniformemente dispersos en el s3mplex de dimensi3n  $k - 1$ , se puede calcular el costo  $C_{rj}$  de asignar el individuo  $\vec{x}_j$  al vector de pesos  $\vec{w}_r$  mediante la expresi3n

$$C_{rj} = \max_{i=1, \dots, k} \frac{f_i(\vec{x}_j)}{w_{ri}}, \quad r = 1, \dots, n, \quad j = 1, \dots, 2n. \quad (4.5)$$

En esta expresi3n utilizamos la descomposici3n de Tchebycheff modificada propuesta en [16] en lugar de la descomposici3n de Tchebycheff original. Esta selecci3n es debida a que el vector de pesos de un subproblema y la direcci3n de su soluci3n 3ptima coinciden en la versi3n modificada, pero en el caso de la descomposici3n de Tchebycheff original la relaci3n entre estas dos direcciones es no lineal. Por tanto, con la descomposici3n de Tchebycheff modificada si es posible controlar de manera directa la distribuci3n de las soluciones en el espacio objetivo.

Los valores de la matriz  $C$  obtenida con 4.5 indican cu3n adecuada es una soluci3n para representar cada regi3n del frente de Pareto. Esta matriz de costo unida a las siguientes dos condiciones constituyen nuestro problema de asignaci3n lineal. La soluci3n de este problema consiste en encontrar una combinaci3n de valores de  $C$  cuya suma sea el menor valor posible y se cumplan ambas condiciones.

1. Se debe seleccionar un 3nico valor por cada fila. Esto asegura que s3lo un individuo es asignado a cada regi3n del frente de Pareto.
2. Se puede seleccionar cuando m3s un valor por cada columna. Esto asegura que un mismo individuo no sea asignado a m3s de una regi3n del frente de Pareto.

La definición matemática de este tipo de problemas se presenta en la sección 4.2 (ver la expresión (4.7)). En esa misma sección se explica el método empleado para su solución, que consiste en una variante del algoritmo de Kuhn-Munkres para matrices rectangulares. Como se puede observar en (4.7), la solución de un problema de asignación se puede representar con una matriz. En nuestro problema, las filas de dicha matriz corresponden a los vectores de pesos y las columnas a los individuos de la población, de tal forma que un uno en la entrada  $ij$  implica que el individuo  $j$  fue asignado al vector  $i$ . El conjunto  $\mathcal{I}$  presente en el Algoritmo 6 (en la página 43) contiene todas las columnas de la matriz solución donde existe un uno. La población  $P_{g+1}$  de la siguiente generación estará constituida por todos los individuos que hayan sido asignados. Este procedimiento se repite hasta que se satisfagan las condiciones de parada del algoritmo. La última población de individuos  $P_{gmax}$  es la aproximación que realiza EDH del conjunto de óptimos de Pareto y del frente de Pareto.

## 4.2. Algoritmo de Kuhn-Munkres

Los problemas de asignación conforman una de las clases fundamentales de la optimización combinatoria. En su forma más general se pueden expresar como: dados un conjunto de agentes, un conjunto de tareas, las restricciones sobre cuáles agentes pueden realizar cada tarea y el costo de esa ejecución, se requiere desarrollar todas las tareas asignando un solo agente por tarea y sin repetir un mismo agente en más de una tarea, de manera que el costo total sea mínimo. El caso más simple de los problemas de asignación son los problemas de asignación lineal (PALs). En los PALs, se considera que el costo total es la suma del costo de cada asignación. Además, el número de agentes y de tareas coincide, y cualquier agente puede ser asignado para realizar cada tarea. Formalmente, un PAL se puede formular como se indica en la Definición 8.

**Definición 8 (Problema de asignación lineal)** *Dados un conjunto de agentes  $A = \{a_1, \dots, a_n\}$ , un conjunto de tareas  $T = \{t_1, \dots, t_n\}$ , la función  $C : A \times T \rightarrow \mathbb{R}$  que representa el costo de realizar las tareas de  $T$  con los elementos de  $A$  y sea  $\Phi : A \rightarrow T$  el conjunto de todas las posibles biyecciones entre  $A$  y  $T$ . Se requiere*

$$\min_{\phi \in \Phi} \sum_{a \in A} C(a, \phi(a)) \quad (4.6)$$

Es más común que los problemas de asignación lineal se formulen como problemas de programación lineal entera. Para esto, la función de costo  $C$  se representa como una matriz cuadrada de valores reales, donde cada elemento  $c_{ij} = C(a_i, t_j)$ . Además, el conjunto  $\Phi$  de todas las posibles biyecciones entre  $A$  y  $T$  se representa como el conjunto  $\mathcal{X}$  de todas las posibles matrices de asignación. En una matriz de asignación  $x \in \mathcal{X}$ , las filas representan los agentes y las columnas a las tareas. En caso de ser asignado el agente  $i$  para realizar la tarea  $j$ , la entrada  $x_{ij}$  de la matriz contiene un uno, de lo contrario contiene un cero. De esta manera, un PAL se puede expresar

como el siguiente problema de programación lineal entera.

$$\begin{aligned}
 \min_{x \in \mathcal{X}} \quad & \sum_{i=1}^n \sum_{j=1}^n C_{ij} x_{ij} \\
 \text{sujeto a:} \quad & \sum_{i=1}^n x_{ij} = 1, \forall j \in \{1, \dots, n\}, \\
 & \sum_{j=1}^n x_{ij} \leq 1, \forall i \in \{1, \dots, n\}, \\
 & x_{ij} \in \{0, 1\}, \forall i, j \in \{1, \dots, n\}.
 \end{aligned} \tag{4.7}$$

**Ejemplo de un problema de asignación lineal:** Sea  $P = \{P_1, P_2, P_3\}$  un conjunto de procesadores y  $T = \{T_1, T_2, T_3\}$  un conjunto de tareas. El costo de asignar la tarea  $T_j$  al procesador  $P_i$  está dado por la entrada  $C_{ij}$  de la matriz

$$C = \begin{matrix} & \begin{matrix} T_1 & T_2 & T_3 \end{matrix} \\ \begin{matrix} P_1 \\ P_2 \\ P_3 \end{matrix} & \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{pmatrix} \end{matrix}$$

Se desea minimizar el costo total de ejecutar cada tarea con un procesador distinto.

La solución de este problema se podría obtener mediante un método de fuerza bruta, el cual realizaría una búsqueda exhaustiva de todas las posibles combinaciones para asignar cada tarea a un procesador distinto. De esta manera, se podría calcular el costo total de cada combinación y seleccionar la de menor valor. Sin embargo, el número de combinaciones posibles es  $n!$ , donde  $n$  es el número de tareas a realizar. En el ejemplo presentado, el valor de  $n$  es 3 y, por tanto, solo se analizarían 6 combinaciones, pero para valores mayores de  $n$  este método no es práctico por su alto costo computacional. Una forma de obtener en menor tiempo una posible solución a este problema es utilizar una método voraz. En este caso, se podría asignar a cada tarea el procesador con menor costo que aún no haya sido asignado. En la figura 4.3 se presentan la solución obtenida con una estrategia voraz y la solución óptima del problema (se resaltan las asignaciones realizadas en cada caso). La estrategia voraz consiste en asignar cada tarea, comenzando por la primera, al procesador disponible que la ejecute con el menor costo. Aunque esta estrategia sea muy eficiente en cuanto al costo computacional, no siempre encuentra la solución óptima, tal y como se muestra en la figura 4.3.

Solución voraz del PAL	Solución óptima del PAL
$  C = \begin{matrix} & \begin{matrix} T_1 & T_2 & T_3 \end{matrix} \\ \begin{matrix} P_1 \\ P_2 \\ P_3 \end{matrix} & \begin{pmatrix} \mathbf{1} & 2 & 3 \\ 2 & \mathbf{4} & 6 \\ 3 & 6 & \mathbf{9} \end{pmatrix} \end{matrix}  $	$  C = \begin{matrix} & \begin{matrix} T_1 & T_2 & T_3 \end{matrix} \\ \begin{matrix} P_1 \\ P_2 \\ P_3 \end{matrix} & \begin{pmatrix} 1 & 2 & \mathbf{3} \\ 2 & \mathbf{4} & 6 \\ \mathbf{3} & 6 & 9 \end{pmatrix} \end{matrix}  $

Figura 4.3: Solución de un problema de asignación lineal (PAL).

En 1955, Harold W. Kuhn [117] propuso un algoritmo de optimización combinatoria que soluciona en tiempo polinomial los problemas de asignación lineal. Este trabajo fue pionero en el área y, según su autor, estuvo inspirado en el trabajo previo de los matemáticos húngaros D. König y E. Egerváry; razón por la cual Kuhn denominó a su propuesta el *Método Húngaro*. Posteriormente, James Munkres revisó el trabajo de Kuhn y le realizó importantes modificaciones a los aspectos teóricos del algoritmo. Munkres determinó que el algoritmo posee un tiempo de ejecución de orden polinomial y propuso en 1957 una versión de orden  $O(n^3)$  [118]. Debido a las contribuciones que realizara Munkres al desarrollo del Método Húngaro, actualmente este método también se conoce por el nombre de *Algoritmo de Kuhn-Munkres*.

El algoritmo de Kuhn-Munkres se construyó para solucionar problemas de asignación lineal donde el número de agentes coincide con el número de tareas y, por tanto, la matriz de costo es una matriz cuadrada. Una extensión de este algoritmo para matrices rectangulares fue propuesta por Bourgeois y Lassalle en 1971 [119]. La extensión realizada al algoritmo de Kuhn-Munkres permite solucionar problemas donde el número de agentes y tareas es distinto. A continuación, se presenta la descripción de cada etapa de la extensión propuesta por Bourgeois y Lassalle (en la figura 4.4 se muestra un ejemplo de la ejecución de este algoritmo).

- **Paso 1:** para cada fila de la matriz de costo, encontrar el menor elemento y restarlo a cada elemento de la fila. Al terminar, ir al paso 2.
- **Paso 2:** encontrar un cero en la matriz resultante del paso 1. Si no está marcado con un asterisco ( $0^*$ ), ni existe otro cero marcado en su fila o columna, marcarlo. Repetir para cada cero en la matriz. Ir al paso 3.
- **Paso 3:** cubrir cada columna (marcar todas las celdas de la columna) que contenga un  $0^*$ . Si son cubiertas  $k$  columnas, donde  $k$  es el mínimo entre el número de columnas y el número de filas, las posiciones de los  $0^*$  describen un conjunto completo de asignaciones únicas. En este caso se termina, en caso contrario, ir al paso 4.
- **Paso 4:** encontrar un cero que no haya sido cubierto y marcarlo con una tilde ( $0'$ ). Si no existe un  $0^*$  en su fila, ir al paso 5. En caso contrario, cubrir su fila y descubrir la columna que contiene al  $0^*$ . Continuar de esta forma hasta que no existan ceros sin cubrir. Encontrar el menor valor no cubierto e ir al paso 6.
- **Paso 5:** construir la serie  $(Z_0, Z_1, Z_2, \dots, Z_m)$  alternando  $0'$  y  $0^*$  como sigue.  $Z_0$  es el  $0'$  no cubierto encontrado en el paso 4,  $Z_1$  es el  $0^*$  en la columna de  $Z_0$  y  $Z_2$  es el  $0'$  en la fila de  $Z_1$  (siempre existirá uno). Continuar de esta manera hasta que la serie termine en un  $0'$  (representado por  $Z_m$ ) que no tiene  $0^*$  en su columna. Luego, desmarcar cada  $0^*$  de la serie y transformar cada  $0'$  de la serie en un  $0^*$ . Posteriormente, desmarcar cada  $0'$  que no pertenezca a la serie y descubrir todas las filas y columnas de la matriz. Volver al paso 3.

- Paso 6:** Sumar el menor valor no cubierto a cada elemento de las filas cubiertas y sustraerlo de cada elemento de las columnas no cubiertas. Volver al paso 4 sin alterar los  $0'$ , los  $0^*$  y las columnas y filas cubiertas.

En la figura 4.4 se muestran los pasos realizados durante la ejecución del algoritmo de Kuhn-Munkres para solucionar el problema de asignación lineal presentado anteriormente en esta sección.

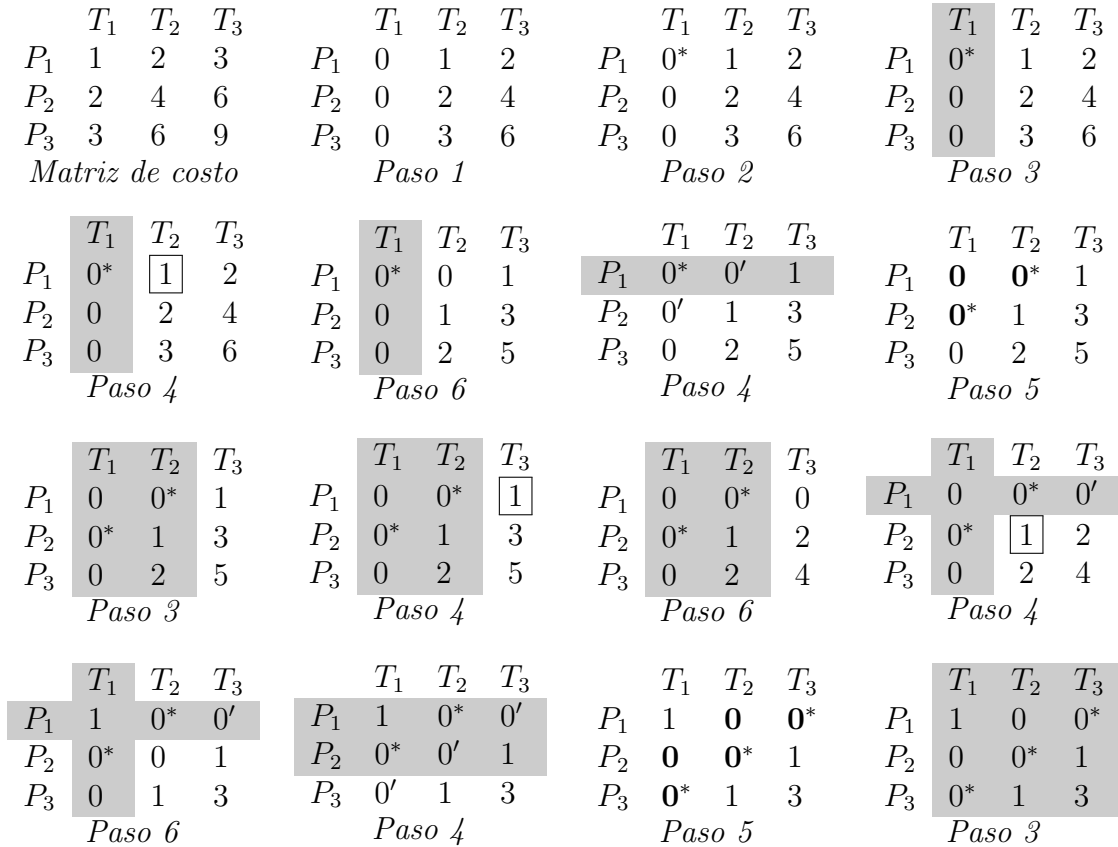


Figura 4.4: Ejemplo de la ejecución del algoritmo de Kuhn-Munkres.

### 4.3. Generación de vectores de pesos

Existen varios AEMOs [9, 10, 16, 29, 30] que requieren, para aproximar el frente de Pareto en problemas con  $k$  objetivos, un conjunto  $W$  (ver 4.4) de vectores uniformemente dispersos en un símplex de dimensión  $k - 1$ . Este conjunto de pesos se puede generar con diversos métodos presentes en la literatura especializada [120, 121]. El método más utilizado en los AEMOs ha sido el *simplex-lattice* [122], el cual produce el conjunto de puntos igualmente espaciados

$$W = \left\{ \vec{w} \mid w_i \in \left\{ 0, \frac{1}{H}, \frac{2}{H}, \dots, \frac{H-1}{H}, 1 \right\}, i = 1, \dots, k, \sum_{i=1}^k w_i = 1 \right\}, \quad (4.8)$$



donde  $H$  representa el número de divisiones en cada objetivo. Sin embargo, este método presenta tres problemas principales [120]:

1. La distribución de los vectores de pesos no es muy uniforme. En la figura 4.5 (a) se muestra el conjunto de 126 vectores de pesos generados con el *simplex-lattice* en un espacio de seis dimensiones. En esta figura se puede observar que cada componente de los vectores posee sólo 5 valores distintos. Por tanto, los AEMOs que utilicen este conjunto de pesos se limitan a explorar vectores de solución con poca diversidad en el valor de cada función objetivo.
2. El número de vectores generados aumenta de manera no lineal con respecto al número de objetivos. Esto es debido a que el tamaño del conjunto  $\mathcal{W}$  es  $\binom{H+k-1}{k-1}$ . En la figura 4.5 (b) se puede observar el crecimiento exponencial del número de vectores con respecto al número de objetivos para un valor fijo de  $H = 15$ . Esto dificulta mantener la misma precisión de muestreo del espacio a medida que aumenta el número de objetivos e impide construir un conjunto de pesos de tamaño arbitrario.
3. La mayor parte de los vectores están distribuidos en la frontera del simplex. La figura 4.5 (c) presenta el porcentaje de vectores de pesos ubicados en la frontera del simplex con respecto al total, a medida que aumenta el número de objetivos (el valor de  $H$  se fijó en 15). Se puede observar que para 7 objetivos apenas un 5% de los vectores no son fronterizos y para 9 objetivos o más esta cifra no alcanza el 1%. Esto muestra que a pesar de generar un gran número de vectores (en 10 dimensiones es del orden de  $10^6$ ) no se representa correctamente el interior del simplex.

Para evitar las deficiencias que posee el *simplex-lattice*, algunos AEMOs han utilizado otros métodos para generar el conjunto de vectores de pesos. En [30] se propone un algoritmo que obtiene los vectores de pesos maximizando el hipervolumen cubierto por éstos en el espacio objetivo. Por otra parte, en [12, 13, 14] se combinan el método *glp* (*good lattice point*) [123] y el diseño uniforme (DU) [120] para calcular los vectores de pesos. Sin embargo, tanto el cálculo del hipervolumen como el método *glp* tienen un alto costo computacional cuando aumenta el número de objetivos.

Recientemente, algunos autores [12, 13, 14] han optado por utilizar métodos de diseño uniforme para generar los vectores de pesos en los AEMOs. El DU es una técnica que busca puntos uniformemente espaciados en un dominio [120] y ha tenido gran aplicación en problemas industriales [124]. En el DU se determina el conjunto de vectores  $\mathcal{P}$  con mejor distribución en un dominio  $\mathcal{D}$  utilizando una medida de uniformidad  $\mathcal{M}$ . Los métodos de DU buscan el conjunto  $\mathcal{P}^* \subset \mathcal{D}$  que posee el menor valor de  $\mathcal{M}$ :

$$\mathcal{P}^* = \arg \min_{\mathcal{P} \subset \mathcal{D}} \mathcal{M}(\mathcal{P}). \quad (4.9)$$

La medida de uniformidad más popular es la *Discrepancia-L<sub>2</sub> Centrada* ( $DC_2$ ) [120]. Sin pérdida de generalidad, se puede considerar el dominio  $\mathcal{D} = [0, 1]^k$ , entonces el

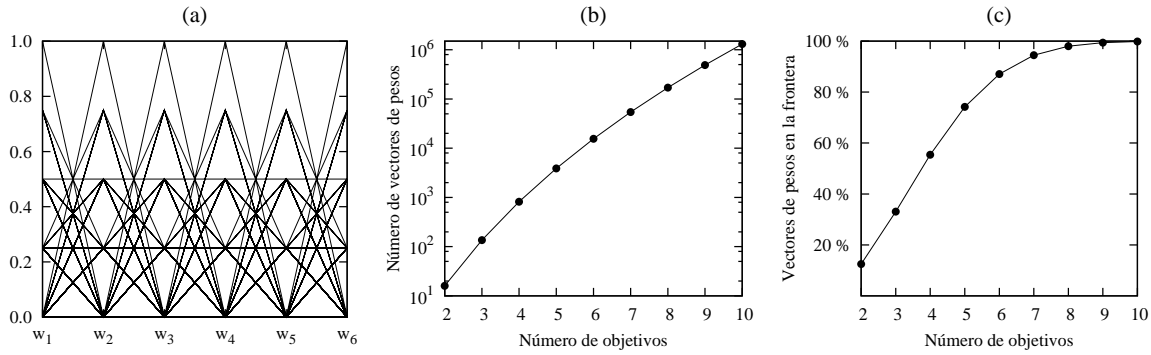


Figura 4.5: Deficiencias del método *simplex-lattice*. En (a) se muestra la alta tasa de repetición del valor de cada componente de los 126 vectores generados en un espacio de 6 dimensiones para  $H = 4$ . En (b) se presenta el crecimiento exponencial del número de vectores y en (c) el aumento de la proporción de vectores fronterizos. Tanto en (b) como en (c) el valor de  $H$  se mantiene en 15 para cualquier número de objetivos.

valor de  $DC_2$  para un conjunto de puntos  $\mathcal{P} = \{\vec{x}_1, \dots, \vec{x}_n\} \subset \mathcal{D}$  se determina por la expresión

$$\begin{aligned}
 DC_2(\mathcal{P})^2 &= \left(\frac{13}{12}\right)^k - \frac{2}{n} \sum_{i=1}^n \prod_{j=1}^k \left(1 + \frac{1}{2}|x_{ij} - \frac{1}{2}| - \frac{1}{2}|x_{ij} - \frac{1}{2}|^2\right) \\
 &+ \frac{1}{n^2} \sum_{i=1}^n \sum_{r=1}^n \prod_{j=1}^k \left(1 + \frac{1}{2}|x_{ij} - \frac{1}{2}| + \frac{1}{2}|x_{rj} - \frac{1}{2}| - \frac{1}{2}|x_{ij} - x_{rj}|\right) \quad (4.10)
 \end{aligned}$$

La búsqueda de  $\mathcal{P}^*$  cuando  $k > 1$  es un problema NP duro, aún para valores pequeños de  $n$ . Debido a esto, se ha propuesto utilizar métodos que aproximen el conjunto  $\mathcal{P}^*$ . Fang y Wang [120] recomiendan diversos métodos cuasi Monte-Carlo que pueden aproximar  $\mathcal{P}^*$  con un valor de  $DC_2$  de orden  $O((\log n)^k/n)$ . Algunos de estos métodos son muy eficientes en cuanto al costo computacional, pero se limitan a determinados valores de  $n$  y  $k$ . Otros métodos, como el *glp*, obtienen buenas aproximaciones de  $\mathcal{P}^*$  para valores arbitrarios de  $n$  y  $k$ , pero requieren un elevado tiempo de cómputo.

En este trabajo, proponemos utilizar el método de Hammersley [125] para aproximar el conjunto  $\mathcal{P}^*$  de puntos uniformemente dispersos en el espacio  $[0, 1]^k$ . Este algoritmo permite obtener un número arbitrario de vectores  $k$ -dimensionales con una buena distribución, cercana a la obtenida por *glp*, pero con un costo computacional muy inferior. El método de Hammersley se basa en la representación *p-ádica* de los números naturales, según la cual todo entero positivo  $m$  se puede expresar de manera única utilizando como base un número primo  $p \geq 2$ :

$$m = \sum_{i=0}^r b_i \times p^i, \quad 0 \leq b_i \leq p-1, \quad i = 0, \dots, r \quad (4.11)$$

---

**Algoritmo 7** Generación de los vectores de pesos

---

**Input:** número de objetivos ( $k$ ), número de vectores de pesos( $n$ )

**Output:**  $W$  (conjunto de vectores de pesos con baja discrepancia)

```

1:  $p \leftarrow$  conjunto con los primeros  $k - 2$  números primos
2:  $U \leftarrow \emptyset$ 
3: for  $i = 1$  to  $n$  do
4:    $u_{i1} \leftarrow (2i - 1)/2n$ 
5:   for  $j = 2$  to  $k - 1$  do
6:      $u_{ij} \leftarrow 0$ 
7:      $f \leftarrow 1/p_{j-1}$ 
8:      $d \leftarrow i$ 
9:     while  $d > 0$  do
10:       $u_{ij} \leftarrow u_{ij} + f \times (d \text{ mód } p_{j-1})$ 
11:       $d \leftarrow \lfloor d/p_{j-1} \rfloor$ 
12:       $f \leftarrow f/p_{j-1}$ 
13:    end while
14:  end for
15:   $U \leftarrow U \cup \{u\}$ 
16: end for
17:  $W \leftarrow$  Aplicar la transformación (4.14) a  $U$ 

```

---

donde  $p^r \leq m < p^{r+1}$ . De la representación 4.11 de todo número natural  $m \geq 1$  se puede derivar su *inverso radical*  $y_p(m)$  mediante la expresión

$$y_p(m) = \sum_{i=0}^r b_i \times p^{-(i+1)} \quad (4.12)$$

donde  $y_p(m) \in (0, 1)$ . Sea  $k \geq 2$  y  $P = \{p_1, \dots, p_{k-1}\}$  un conjunto de  $k - 1$  primos distintos, se denomina *conjunto de Hammersley* (CH) a los  $n$  puntos uniformemente dispersos en el espacio  $[0, 1]^k$  definidos por

$$\vec{x}_i = \left[ \frac{2i-1}{2n}, y_{p_1}(i), \dots, y_{p_{k-1}}(i) \right]^T, \quad i = 1, \dots, n. \quad (4.13)$$

El conjunto  $W$  (ver 4.4) de vectores de pesos uniformemente dispersos en el  $(k - 1)$ -símplex se puede obtener utilizando una aproximación  $U$  del conjunto óptimo  $\mathcal{P}^* \in [0, 1]^{k-1}$ . Para lograr esto, Wang y Fang [126] proponen el Diseño Uniforme de Experimentos con Mezclas. Este método transforma el conjunto  $U = \{\vec{u}_i = [u_{i1}, \dots, u_{i(k-1)}]^T, i = 1, \dots, n\}$  en el conjunto  $W = \{\vec{w}_t = [w_{t1}, \dots, w_{tk}]^T, t = 1, \dots, n\}$  de

la siguiente manera.

$$w_{ti} = (1 - u_{ti}^{\frac{1}{k-i}}) \prod_{j=1}^{i-1} u_{tj}^{\frac{1}{k-j}}, \quad i = 1, \dots, k-1,$$

$$w_{tk} = \prod_{j=1}^{k-1} u_{tj}^{\frac{1}{k-j}}, \quad t = 1, \dots, n. \quad (4.14)$$

En el presente trabajo, se genera el conjunto  $U$  utilizando el método de Hammersley y posteriormente se le aplica a  $U$  la transformación 4.14 para obtener  $W$ . En el Algoritmo 7 se muestra el pseudocódigo para calcular el conjunto de vectores de pesos  $W$ .

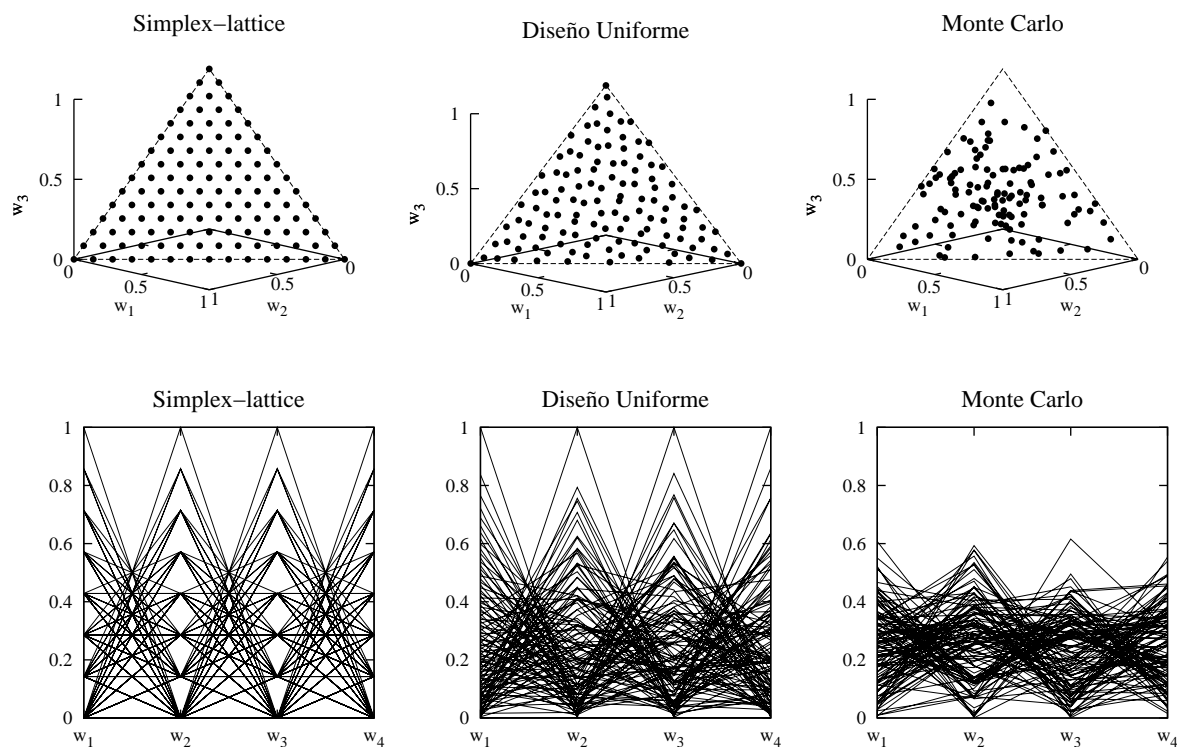


Figura 4.6: Vectores de pesos obtenidos en los símplex de 3 y 4 dimensiones. A la izquierda, el conjunto generado con el método *simplex-lattice*. En el centro, los vectores construidos con la combinación del método de Hammersley y el diseño uniforme de Experimentos con Mezclas. A la derecha se presentan los vectores obtenidos por muestreo aleatorio del espacio.

En la figura 4.6 se muestran los vectores de pesos obtenidos mediante tres técnicas distintas: el *simplex-lattice*, el método de diseño uniforme presentado en el Algoritmo 7 y el muestreo aleatorio. Se puede apreciar en esta figura que el diseño uniforme obtiene mucho menos elementos en la frontera que el *simplex-lattice*, sin perder la buena distribución. Además, la tasa de repetición del valor de cada componente de

Tabla 4.1: Uniformidad de los conjuntos de vectores de pesos generados con el método *simplex-lattice*, diseño uniforme y Monte Carlo según la medida *Discrepancia-L<sub>2</sub> Centrada*.

Objetivos	Número de vectores	Discrepancia-L <sub>2</sub> Centrada		
		<i>Simplex-lattice</i>	Diseño Uniforme	Monte Carlo
3	120	0.332	0.336	0.374
4	120	0.630	0.643	0.703
5	126	0.959	0.987	1.068
6	126	1.308	1.394	1.509
7	210	1.733	1.883	2.021
8	120	2.318	2.476	2.674
9	165	2.938	3.178	3.431
10	220	3.686	4.036	4.343

los vectores de pesos es muy inferior. En cuanto al método de Monte Carlo (fue utilizado el método propuesto en [127]), los vectores generados tienden a concentrarse en el centro del simplex y la uniformidad de su distribución es muy pobre.

En la tabla 4.1 se presenta la uniformidad de la distribución (según la medida  $DC_2$ ) de los conjuntos de vectores mostrados en la figura 4.6 y de otros conjuntos generados en espacios de mayor dimensión. Se puede observar en los valores de esta tabla que el *simplex-lattice* obtiene los mejores resultados, lo cual es debido a que los vectores obtenidos con este método se encuentran igualmente espaciados entre sí. El deterioro de la uniformidad de los conjuntos generados por el método de diseño uniforme se debe a la concentración de los vectores hacia el centro del simplex. Este problema lo presenta de manera más acentuada el método de Monte Carlo, el cual, además, no obtiene vectores bien espaciados entre sí. Sin embargo, utilizar un método de diseño uniforme, por ejemplo, el presentado en el Algoritmo 7, permite evitar los tres problemas principales del *simplex-lattice* discutidos al inicio de esta sección.

