



División de Ciencias Básicas e Ingeniería

Preservación de la Diversidad y Manejo de los Puntos de Referencia en Algoritmos Evolutivos Multiobjetivo

Idónea comunicación de resultados para obtener el grado de

Maestro en Optimización

por

Alberto Rodríguez Sánchez

Asesores:

Dr. Antonin Ponsich

Departamento de Sistemas, UAM Azcapotzalco

Dr. Antonio López Jaimes

Departamento de Matemáticas Aplicadas y Sistemas, UAM

Cuajimalpa

25 de marzo de 2022

Dedicatoria

A la memoria de mi profesor y amigo **Oscar Alvarado Nava** cuyas enseñanzas y consejos me han permitido superar obstáculos profesionales y personales.

Gracias.

Agradecimientos

A todos los docentes que me acompañaron durante este proceso y en especial a mis tutores por su ayuda, paciencia y dedicación.

A mis padres por confiar, por los valores y principios que me han inculcado.

Finalmente a mis amigos que aun desde la distancia siempre me apoyaron.

Abstract

The success of Multi-Objective Evolutionary Algorithms based on Decomposition (*MOEA/D*) has generated great interest in MOEAs using a reference set of weight vectors to promote diversity within non-dominated solutions. However, the quality of the solution set obtained heavily depends on the consistency between the problem's Pareto front shape and the specified weight distribution.

The objective of this thesis is twofold. First, the study focuses on six classical techniques for weight vector generation, either based on mixture design or low discrepancy sequences. The performance of these methods, implemented into *MOEA/D* with PBI, was evaluated and analyzed over several scalable test functions from the literature. Moreover, a novel approach for updating the weight vectors during the evolutionary process is introduced, which periodically uses information from the non-dominated individuals in the current population. These solutions are used to re-compute their corresponding weight vector and mark them as tabu points. Then, new weight vectors are created far from these tabu points through a repulsion criterion based on the Mahalanobis distance. Some preliminary experiments indicate that this on-line weight vector adaptation strategy provides significant benefits when compared to the static Simplex Lattice design

Resumen

El éxito de los algoritmos evolutivos multiobjetivo basados en descomposición (MOEA/D) ha generado un gran interés en los MOEA que utilizan un conjunto de vectores de peso para promover la diversidad dentro de las soluciones no dominadas. Sin embargo, la calidad del conjunto de soluciones obtenido depende en gran medida de la consistencia entre la forma del frente de Pareto del problema y la distribución de peso especificada.

El objetivo de este trabajo es doble. En primer lugar, el estudio se centra en seis técnicas clásicas para la generación de vectores de peso, ya sea basadas en el diseño de mezclas o en secuencias de baja discrepancia. El rendimiento de estos métodos, implementados en MOEA/D con PBI, se evaluó y analizó en varias funciones de prueba escalables de la literatura.

Además, se introduce un enfoque novedoso para actualizar los vectores de peso durante el proceso evolutivo, que utiliza información de los individuos no dominados de la población actual. Estas soluciones se utilizan para volver a calcular su vector de peso correspondiente y marcarlas como puntos prohibidos o tabú. Luego, se crean nuevos vectores de peso lejos de estos puntos tabú a través de un criterio de repulsión basado en la distancia de Mahalanobis. Algunos experimentos preliminares indican que esta estrategia de adaptación dinámica del vector de peso proporciona beneficios significativos en comparación con el diseño estático Simplex Lattice ampliamente utilizado pero que ha mostrado problemas en altas dimensiones.

Contenido

Lista de Figuras	XIII
Lista de Tablas	xv
1. Introducción	3
1.1. Objetivo general	5
1.2. Objetivos particulares	5
2. Antecedentes	7
2.1. Optimización Multiobjetivo	7
2.1.1. Formulación del Problema de Optimización Multiobjetvos	7
2.1.2. Calidad de un conjunto de soluciones del POM	9
2.1.3. Principales algoritmos evolutivos de optimización Multiobjetivo	11
2.2. Dos algoritmos relevantes	14
2.2.1. Marco de trabajo MOEA/D	14
2.2.2. Marco de trabajo NSGA-III	17
2.3. Mantenimiento de la diversidad	19

3. Investigación sobre las técnicas estáticas de generación de vectores/puntos de referencia	23
3.1. Técnicas basadas en diseños de mezclas	24
3.1.1. Simplex Lattice	24
3.1.2. Simplex Lattice de dos capas	25
3.2. Diseño Uniforme	26
3.2.1. Motivación y descripción de la técnica de generación de diseños casi uniformes	29
3.2.2. Implementación	30
3.2.3. Experimentos computacionales y resultados	33
3.3. Secuencias de baja discrepancia	35
3.4. Resumen	37
4. Resultados experimentales sobre técnicas estáticas de generación de vectores/puntos de referencia	39
4.1. Funciones de prueba	39
4.1.1. Funciones de prueba Deb-Thiele-Laumanns-Zitzler	39
4.1.2. Funciones de prueba DTLZ inversas	41
4.1.3. <i>Kite</i> o Papalote	42
4.2. Resultados experimentales con métodos estáticos	43
4.2.1. NSGA-III	44
4.2.2. MOEA/D Chebyshev (TCH)	46
4.2.3. MOEA/D PBI	48
4.3. Resumen	51

5. Métodos para ajuste dinámico de pesos	53
5.1. Motivación	53
5.2. Repulsión de subpoblaciones	54
5.2.1. Estrategia propuesta para la actualización de vectores de pesos	56
5.2.2. Selección Tabú	57
5.2.3. Reubicación y actualización del peso	58
5.2.4. Metodología experimental	58
5.2.5. Resultados numéricos y discusión	59
5.3. Resumen	63
6. Conclusiones y trabajo futuro	65
6.1. Conclusiones	65
6.2. Trabajo futuro	66
Bibliografía	67

Lista de Figuras

2.1.	Relaciones entre convergencia y diversidad en el \mathcal{PF}_{approx}	10
3.1.	Hiperplano unitario generado con $H = 23$, $k = 3$ y el método de Das & Dennis.	25
3.2.	Hiperplanos unitarios en $k = 5$ y $k = 8$ generados con el método de Das & Dennis expresados en gráficas de coordenadas paralelas.	26
3.3.	Hiperplano unitario generado con el método de Simplex lattice de dos capas.	27
3.4.	Hiperplanos unitarios en $k = 5$ y $k = 8$ generados con el método de Simplex lattice de dos capas.	28
3.5.	Gráficas de coordenadas paralelas e Hipervolumen para distintas técnicas de generación de puntos de referencia.	34
5.1.	Hyb-MOEA/D vs. SLD-MOEA/D para DTLZ7.	62
5.2.	Nor-MOEA/D vs. SLD-MOEA/D para función papalote.	62
5.3.	Lin-MOEA/D vs. SLD-MOEA/D para DTLZ ⁻¹	62

Lista de Tablas

4.1. Diseño experimental	43
4.2. Resultados con NSGA-III	45
4.3. Resultados con MOEA/D con TCH	47
4.4. Resultados con MOEA/D PBI	49
5.1. Diseño experimental.	60
5.2. Resultados con Técnicas dinámicas de adaptación de pesos.	61

Índice de algoritmos

2.2.1.Algoritmo MOEA/D	16
2.2.2.Algoritmo NSGA-III	18
3.2.1.Búsqueda Tabú paralela	32
5.2.1.MOEA/D con adaptación de vectores de peso dirigida por dominancia	57
5.2.2.Selección Tabú	58
5.2.3.Reubicación de vectores de peso	59

Introducción

Los problemas de optimización multiobjetivo (**POM**) han sido de gran utilidad en la comunidad científica e industrial, ya que son comunes aquellos problemas que requieren considerar múltiples objetivos, incluyendo tiempo, costos, cantidad, calidad, etc. Estos objetivos se encuentran típicamente en conflicto y se pueden encontrar en diferentes áreas de aplicación, como: ruteo de vehículos, localización, cadenas de suministro, calendarización, entre otros [6].

Esta clase de problemas ha sido tradicionalmente atacada reduciendo el **POM** a un problema mono-objetivo (estrategias de suma ponderada, de restricciones- ϵ , de programación por metas). Sin embargo, desde los años 1990, los esfuerzos de investigación se han dedicado al desarrollo de técnicas metaheurísticas y, particularmente, Algoritmos Evolutivos (AEs), para la solución del POM. Desde el punto de vista metaheurístico, existen tres grandes enfoques para resolver el **POM**:

- Basado en descomposición.
- Basado en dominancia.
- Basado en indicadores.

El enfoque de descomposición explícitamente descompone un problema multiobjetivo en N problemas de optimización escalar y suele introducir técnicas poblacionales para la resolución simultánea de ellos. Bajo este enfoque, es importante seleccionar los problemas escalares conforme a algún criterio que permita mejorar la diversidad del conjunto de soluciones arrojado, dejando la convergencia al submétodo de resolución (que puede ser una técnica exacta, un algoritmo de aproximación, una metaheurística, etc.). Un marco de trabajo popular con este enfoque es *multiobjective evolutionary algorithm based on decomposition (MOEA/D)* [57].

En el enfoque de dominancia, se utiliza la relación de dominancia para inducir un orden parcial en el espacio de los objetivos y, de esta manera, decidir cuándo una solución en dicho espacio es comparable con otra y, en este caso, cuál es la mejor. Cabe mencionar que dicho orden parcial está limitado en espacios de altas dimensiones puesto que muchas soluciones serán no comparables, mermando en consecuencia la efectividad de los procedimientos de selección. El algoritmo genético de ordenamiento por no dominancia (*NSGA-III*) [9] es un marco de trabajo popular con esta estrategia, que incorpora además técnicas de nichos para mantener la diversidad.

Finalmente, el enfoque basado en indicadores convierte el problema multiobjetivo original en un problema mono o multiobjetivo diferente, que optimiza el valor de algún indicador de desempeño, representando la calidad del conjunto de soluciones obtenidas tanto en términos de convergencia como de dispersión y uniformidad de su distribución. Un indicador popular es el hipervolumen ya que es el único acorde a la definición de optimalidad de Pareto (i.e., las soluciones Pareto-óptimas son aquellas que maximizan el hipervolumen). Sin embargo, calcular el hipervolumen se vuelve costoso conforme aumenta la dimensión del problema multiobjetivo original. Por ejemplo, la mejor forma conocida de calcular el hipervolumen tiene una complejidad de $O(n^2m^3)$ [51], donde n es el número de soluciones aproximadas y m el número de objetivos.

Un conjunto de soluciones candidatas entregadas por cualquiera de los enfoques antes mencionados debe de cumplir con criterios de convergencia y diversidad para ser considerado aceptable. Esta meta de obtener un conjunto de soluciones diversas y uniformemente distribuidas se ve dificultada cuando el **POM** tiene alguna de las siguientes características:

- Convexidad o no convexidad del frente óptimo de Pareto.
- Discontinuidad del frente óptimo de Pareto.
- Densidad no uniforme de las soluciones en el frente óptimo de Pareto.
- Degeneración del frente óptimo de Pareto.

Para asegurar la diversidad de las soluciones determinadas, muchos de los enfoques recientes basados en dominancia y descomposición utilizan puntos y/o vectores de referencia, que representan direcciones de búsqueda en el espacio objetivo. Si bien se han estudiado distintos métodos para generarlos (ver capítulos siguientes), sus valores se mantienen clásicamente fijos durante el transcurso de la ejecución del algoritmo. Esta estrategia de puntos/vectores de referencia estáticos experimenta problemas cuando el frente óptimo presenta las fuentes

de dificultad antes mencionadas.

Existen distintas aproximaciones a la adaptación dinámica de vectores de pesos, por ejemplo, Qi et al proponen un algoritmo de dos fases que inicializa los vectores de referencia como un subconjunto de vectores de pesos que permite aproximar el frente de Pareto y posteriormente substituir los vectores de pesos en regiones mas densas y agregar vectores nuevos en regiones menos pobladas, sin embargo, los resultados muestran los retos de la adaptación de vectores de pesos como lo son la reducción de la convergencia, la disminución del rendimiento de los algoritmos aun en problemas sencillos y la dificultad de que una sola técnica de adaptación para las diversas formas del frente de Pareto. De la optimización multimodal vienen técnicas que incentivan la búsqueda en regiones poco explotables pero prometedoras para encontrar múltiples óptimos locales y globales, estas técnicas de exploración pueden ser utilizadas para reubicar vectores de peso en algoritmos evolutivos multiobjetivo. En este trabajo se utilizan ideas provenientes de la repulsión de sub poblaciones como criterio de reubicación de vectores de pesos.

Por lo tanto, las hipótesis centrales del trabajo son las siguientes:

- La técnica de inicialización de los puntos de referencia/vectores de pesos tiene un impacto directo en el desempeño de los algoritmos evolutivos multiobjetivo (MOEAs por sus siglas en inglés).
- Es posible actualizar los puntos de referencia/vectores de pesos en el transcurso de la búsqueda adaptando la estrategia de repulsión de subpoblaciones al contexto multiobjetivo.

1.1. Objetivo general

Adaptar e integrar estrategias de generación y actualización de los vectores de pesos en el *MOEA/D* y de los puntos de referencia en el *NSGA-III*, para evaluar su impacto sobre el desempeño en términos de diversidad de estos dos algoritmos evolutivos multiobjetivo.

1.2. Objetivos particulares

- Seleccionar problemas de prueba clásicos en Optimización Multiobjetivo, escalables en número de dimensiones y que incluyan frentes de Pareto con características y formas
-

complicadas.

- Implementar los algoritmos evolutivos multiobjetivo (MOEAs) clásicos, *MOEA/D* y *NSGA-III* y adaptarlos a la resolución de las instancias propuestas en el punto anterior.
- Identificar al menos dos técnicas estáticas de generación de puntos de referencia relevantes para MOEAs, e implementarlos para ser utilizados con *MOEA/D* y *NSGA-III*.
- Realizar experimentos computacionales exhaustivos sobre las instancias seleccionadas, con los dos MOEAs y las técnicas de generación de vectores de pesos. Realizar comparaciones basadas en algunos indicadores de desempeño.
- Desarrollar e implementar una técnica de actualización dinámica de los puntos de referencia, basada en la estrategia de repulsión de subpoblaciones, para al menos uno de los dos MOEAs utilizados anteriormente.
- Realizar experimentos computacionales exhaustivos sobre las instancias seleccionadas, con la técnica de actualización dinámica de los puntos de referencia integrada a un MOEA y compara con la versión estática correspondiente.

Después de esta introducción, el presente trabajo se organiza de la siguiente manera. En el capítulo 2, se introducen brevemente los conceptos de la optimización multiobjetivo y el estado del arte en la adaptación dinámica de los vectores de pesos. En el capítulo 3, se describe la técnica de generación de diseños casi uniformes en base a una búsqueda tabú paralela. El capítulo 4 muestra los resultados en términos de discrepancia y pruebas en funciones benchmark bien conocidas. El capítulo 5 se presenta la técnica de adaptación dinámica de vectores de peso basada en repulsión de subpoblaciones y algunas variaciones cuyos resultados se comparan con las técnicas estáticas, finalmente el capítulo 6 muestra las conclusiones y perspectivas de trabajo futuro sobre las adaptación dinámica de vectores de peso.

Antecedentes

Un Problema de Optimización Multiobjetivo (**POM**) (llamado también multicriterio o vectorial) puede definirse como el problema de encontrar[39]:

Un conjunto de vectores de variables de decisión que satisfacen un cierto conjunto de restricciones y optimizan un conjunto de funciones objetivo. Estas funciones forman una descripción matemática de los criterios de desempeño que suelen estar en conflicto unos con otros y que se suelen medir en unidades diferentes. El término “optimizar” en este caso toma pues un significado diferente al del caso de problemas mono-objetivo.

En este capítulo, se presentan, en un primer tiempo, conceptos básicos de optimización multiobjetivo. Luego, se describen dos algoritmos evolutivos multiobjetivo (MOEA) clásicos, de particular relevancia para este trabajo: *MOEA/D* y *NSGA-III* y la relevancia del uso de vectores de pesos para dirigir la búsqueda en el espacio de los objetivos. Se introducen dos clases de técnicas de generación de vectores de pesos y se resaltan sus características más importantes. Finalmente una semblanza del estado del arte en el ámbito de la generación dinámica de los vectores de pesos y la técnica de repulsión de subpoblaciones.

2.1. Optimización Multiobjetivo

2.1.1. Formulación del Problema de Optimización Multiobjetivos

El Problema de Optimización Multiobjetivo (**POM**) es definido en su forma general de la siguiente manera:

$$\text{mín(máx)} \overrightarrow{F(\mathbf{x})} = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})] \quad (2.1)$$

S.A:

$$g_i(\mathbf{x}) \leq 0, 1 \leq i \leq m \quad (2.1a)$$

$$h_j(\mathbf{x}) = 0, 1 \leq j \leq p \quad (2.1b)$$

$$\mathbf{x} \in \Omega \subset \mathbb{R}^n \quad (2.1c)$$

Donde las funciones g_i representan las m restricciones de desigualdad y h_j las p restricciones de igualdad, que definen el espacio factible \mathcal{F} .

Las funciones $f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})$ se encuentran en conflicto de tal manera que mejorar el valor de una de ellas decrementa necesariamente el del resto de las funciones.

Se busca el conjunto de vectores $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ que optimicen la función \overrightarrow{F} .

Dominancia

Se dice de un vector $\mathbf{u} = (u_1, \dots, u_n)$ que domina a otro $\mathbf{v} = (v_1, \dots, v_n)$ (denotado $\mathbf{u} \preceq \mathbf{v}$) si y sólo si \mathbf{u} es parcialmente menor que \mathbf{v} , es decir, $\forall i \in \{1, \dots, n\}, u_i \leq v_i$ y $\exists i \in \{1, \dots, n\}$ tal que $u_i < v_i$.

En el contexto del **POM**, \mathbf{u} y \mathbf{v} se encuentran en el espacio de los objetivos.

Optimalidad de Pareto

Para un problema multiobjetivo dado $\overrightarrow{F(\mathbf{x})}$, el conjunto de óptimos de Pareto (P^*) del POM descrito anteriormente es definido como:

$$P^* = \{x \in \Omega \mid \nexists x_0 \in \Omega : \overrightarrow{F(x_0)} \preceq \overrightarrow{F(x)}\}$$

Es decir, los vectores $x \in P^*$ son los que no son dominados por ninguna otra solución del espacio de búsqueda. La imagen en el espacio de los objetivos del conjunto de óptimos de Pareto se denomina frente de Pareto (\mathcal{PF}).

Puntos de referencia

El frente de Pareto real se encuentra acotado entre dos puntos especiales denominados ideal z^* y de nadir z^{nad}

El punto ideal $z^* \in \mathbb{R}^k$ minimiza todas las funciones objetivo $\overrightarrow{F(\mathbf{x})}$:

$$z^* = \min_{1 \leq i \leq k} (f_i(x) | x \in \mathcal{PF}_{true}) \quad (2.2)$$

El punto de nadir $z^{nad} \in \mathbb{R}^k$ es construido utilizando los peores valores en el \mathcal{PF}_{true}

$$z^{nad} = \max_{1 \leq i \leq k} (f_i(x) | x \in \mathcal{PF}_{true}) \quad (2.3)$$

2.1.2. Calidad de un conjunto de soluciones del POM

Hay al menos tres características que debe cumplir un conjunto de soluciones de un problema multiobjetivo:

- **Precisión:** Se refiere a la convergencia del conjunto de soluciones no-dominadas (frente aproximado, denotado como \mathcal{PF}_{aprox}). Evalúa qué tan lejos se encuentra del frente de Pareto óptimo (o frente de Pareto real, denotado como \mathcal{PF}_{true}). Cuando el frente de Pareto no se conoce, se puede emplear un conjunto de referencia.
- **Uniformidad:** Mide la uniformidad de la distribución espacial de las soluciones aproximadas, generalmente en el espacio de los objetivos, es decir, la distancia relativa entre las soluciones encontradas.
- **Dispersión:** Se refiere al rango de valores cubierto por el frente aproximado, generalmente en espacio de los objetivos.

Estas últimas dos características están relacionadas y suelen denominarse en conjunto con el término diversidad.

La Figura 2.1 muestra de forma gráfica estas características y su efecto en el resultado final del cálculo de \mathcal{PF}_{aprox} .

Un indicador es una función $I : \mathbb{R}^n \rightarrow \mathbb{R}^+$ cuyo valor permite medir la calidad de uno o más frentes aproximados \mathcal{PF}_{aprox} de tamaño n . Algunos indicadores utilizados comúnmente y de particular relevancia en este trabajo son:

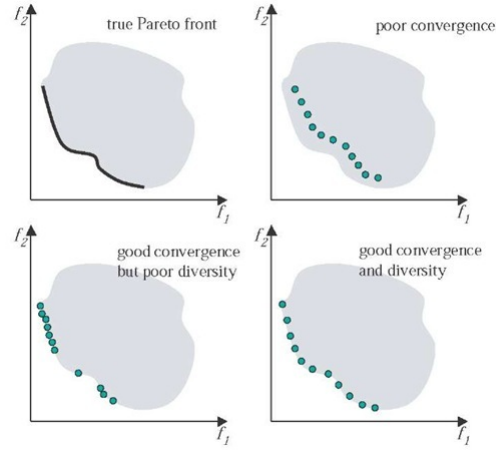


Figura 2.1: Relaciones entre convergencia y diversidad en el \mathcal{PF}_{aprox}

- Distancia generacional: La distancia generacional reporta qué tan lejos, en promedio, se encuentra, en el espacio de los objetivos, el frente aproximado, \mathcal{PF}_{aprox} , del frente óptimo, \mathcal{PF}_{true} .

$$GD(\mathcal{PF}_{aprox}, \mathcal{PF}_{true}) = \frac{(\sum_{i=1}^{|\mathcal{PF}_{true}|} d_i^p(\mathcal{PF}_{aprox}, \mathcal{PF}_{true}))^{\frac{1}{p}}}{|\mathcal{PF}_{aprox}|}, \quad (2.4)$$

donde:

- d_i es la distancia euclidiana mínima entre el punto $p_i \in \mathcal{PF}_{aprox}$ y \mathcal{PF}_{true} .
- p es el parámetro usualmente usado en 1 o 2.
- Distancia generacional inversa: La distancia generacional inversa reporta qué tan lejos, en promedio, se encuentra un conjunto de vectores uniformemente distribuidos en el frente óptimo, \mathcal{PF}_{true} , del frente aproximado, \mathcal{PF}_{aprox} .

$$iGD(\mathcal{PF}_{aprox}, \mathcal{PF}_{true}) = \frac{(\sum_{i=1}^{|\mathcal{PF}_{true}|} d_i(\mathcal{PF}_{aprox}, \mathcal{PF}_{true}))^{\frac{1}{p}}}{|\mathcal{PF}_{true}|}, \quad (2.5)$$

donde d_i es la distancia euclidiana mínima entre el punto $p_i \in \mathcal{PF}_{aprox}$ y \mathcal{PF}_{true} .

- Delta Diversidad: El indicador de diversidad Δ , este último indicador mide dos elementos: *i*) qué tan uniformemente se distribuyen los puntos de un conjunto de aproximación en el espacio objetivo (independientemente de la convergencia), y *ii*) la distancia mínima desde cualquier punto del conjunto a cada uno de los puntos extremos del frente de Pareto real \mathcal{PF}_{true} .

$$\Delta(\mathcal{PF}_{true}, E_i) = \frac{\sum_{i=1}^k d(E_i, \mathcal{PF}_{true}) + \sum_{p_i \in \mathcal{PF}_{true}} |d(p_i, \mathcal{PF}_{true}) - \bar{d}|}{\sum_{i=1}^k d(E_i, \mathcal{PF}_{true}) + (|\mathcal{PF}_{true}| - k)\bar{d}}, \quad (2.6)$$

donde:

- $d(p_i, \mathcal{PF}_{true})$ es la distancia euclidiana mínima entre el punto $p_i \in \mathcal{PF}_{true}$ y el resto de los elementos en \mathcal{PF}_{true} en el espacio de los objetivos.
 - \bar{d} es el promedio de $d(p_i, \mathcal{PF}_{true})$.
 - E es el conjunto de los puntos extremos del frente real \mathcal{PF}_{true} .
- Hipervolumen: El indicador hipervolumen, también conocido como la métrica S , determina el tamaño de la porción de espacio objetivo que está dominada por las soluciones de un conjunto \mathcal{PF}_{aprox} , colectivamente y delimitado por un punto de referencia $r \in \mathbb{R}^n$

$$\mathbf{HV}(\mathcal{PF}_{true}, r) = \Lambda \left(\bigcup_{a \in \mathcal{PF}_{aprox}} (x|a \prec x \prec r) \right), \quad (2.7)$$

donde Λ denota la medida de Lebesgue y r debe estar dominada por todos los elementos en A . Se prefieren los valores más altos de este indicador.

Una propiedad matemática del hipervolumen es que su maximización es equivalente a alcanzar el conjunto óptimo de Pareto [15]. Esto ha sido experimentalmente valioso y se ha observado que tales soluciones también tienen una buena distribución a lo largo del frente de \mathcal{PF}_{aprox} . Sin embargo, el principal inconveniente de utilizar la hipervolumen es su alto costo computacional, que crece exponencialmente con el número de objetivos

2.1.3. Principales algoritmos evolutivos de optimización Multiobjetivo

A lo largo de los años, los algoritmos evolutivos multiobjetivo han mostrado ser efectivos en una gran variedad de problemas de interés académico y práctico. Esto principalmente por la habilidad de encontrar aproximaciones (discretas) del frente de Pareto real (\mathcal{PF}_{true}) en una sola corrida a diferencia de la mayoría de los métodos basados técnicas exactas que requieren tantas ejecuciones como soluciones sean requeridas sin contar con el problema de encontrar el conjunto de pesos dentro de las áreas de interés que permitan convertir el problema multiobjetivo en mono objetivo. Tampoco se asumen características particulares del problema como si éste es diferenciable o continuo.

El primer algoritmo evolutivo multiobjetivo atribuye a David Schaffer en la década de los 80, la idea de asignar una aptitud conforme a la dominancia de Pareto viene de Goldberg

para resolver los problemas expresados en los trabajos de [5]. La idea principal era utilizar el ordenamiento no dominado como parte del criterio de selección para llevar a la población hacia el frente no dominado. Estas ideas se desarrollaron dentro de lo que se conoce la primera generación dentro de los MOEA principalmente enfocada en guiar la búsqueda hacia el \mathcal{PF}_{true} , la segunda empieza a tomar en cuenta el concepto de nicho de tal manera que se restringen la cruce y la mutación dentro de estos subconjuntos de individuos con el fin de resolver los problemas de diversidad presentes en la primera generación de MOEA.

En general, los algoritmos evolutivos multiobjetivo mimetizan los principios de selección natural y tienen las siguientes particularidades:

- Operadores de Selección, mutación y cruce.
- Criterios de supervivencia.
- Una población de individuos o soluciones que “evoluciona” durante la ejecución del algoritmo.
- Un conjunto de reglas que se aplican iterativamente a lo largo de “generaciones”.

Cada algoritmo asigna un valor a cada individuo que representa la convergencia, la diversidad o ambas, de tal manera que estos individuos puedan ser comparados y los mejores (en términos de dicho valor) tienen mayor probabilidad de pasar a la siguiente generación.

Desde el punto de vista metaheurístico, existen tres grandes enfoques para resolver el **POM** con MOEA:

- Basado en descomposición.
- Basado en dominancia.
- Basado en indicadores.

El enfoque de descomposición explícitamente descompone un problema multiobjetivo en N problemas de optimización escalar y suele introducir técnicas poblacionales para la resolución simultánea de ellos. Bajo este enfoque, es importante seleccionar los problemas escalares conforme a algún criterio que permita mejorar la diversidad del conjunto de soluciones arrojado, dejando la convergencia al submétodo de resolución (que puede ser una técnica exacta, un algoritmo de aproximación, una metaheurística, etc.). Un marco de trabajo popular con este enfoque es *multiobjective evolutionary algorithm based on decomposition (MOEA/D)* [57].

En el enfoque de dominancia, se utiliza la relación de dominancia para inducir un orden parcial en el espacio de los objetivos y, de esta manera, decidir cuándo una solución en dicho espacio es comparable con otra y, en este caso, cuál es la mejor. Cabe mencionar que dicho orden parcial está limitado en espacios de altas dimensiones puesto que muchas soluciones serán no comparables, mermando en consecuencia la efectividad de los procedimientos de selección. El algoritmo genético de ordenamiento por no dominancia (*NSGA-III*) [9] es un marco de trabajo popular con esta estrategia, que incorpora además técnicas de nichos para mantener la diversidad.

Finalmente, el enfoque basado en indicadores convierte el problema multiobjetivo original en un problema mono o multiobjetivo diferente, que optimiza el valor de algún indicador de desempeño, representando la calidad del conjunto de soluciones obtenidas tanto en términos de convergencia como de dispersión y uniformidad de su distribución.

De particular interés para este estudio son los métodos basados en la descomposición, por ejemplo MOEA/D[58], *NSGA-III*[10] o LIBEA[54]. Si bien *NSGA-III* es considerado basado en dominancia utiliza un conjunto de puntos de referencia para mejorar la diversidad de las soluciones no-dominadas.

Una posible razón por la que los MOEA de descomposición son populares es que pueden resolver problemas exitosamente con más de tres objetivos. Es decir, llamados problemas de optimización de muchos objetivos (Many objective Optimization Problems: MaOPs). La principal desventaja de las técnicas basadas en dominancia de Pareto es que, a medida que se agregan más objetivos, su capacidad para discernir soluciones buenas y malas, se diluye gradualmente el progreso para llegar al frente óptimo haciendo que este sea casi imposible de alcanzar. Otra ventaja de la descomposición de los MOEA es que las soluciones en el conjunto del frente óptimo de Pareto están notablemente bien distribuidas.

Sin embargo, como algunos estudios han demostrado, la distribución de las soluciones depende de varios factores como:

- Función de escalarización.
- Geometría del frente de Pareto.
- Método de generación de los puntos de referencia.

Por ejemplo, generalmente el método de intersección de límite de penalización (PBI) produce una mejor distribución que la obtenida mediante la función de escalarización de Chebyshev. Además, la distribución depende de varias características del frente de Pareto, a saber,

convexidad, continuidad o degeneración.

Sin embargo, el conjunto de puntos de referencia es el elemento más importante sobre la diversidad/distribución de las soluciones finales. Das y Dennis [8] propusieron la técnica del simplex lattice dentro del marco de optimización Multiobjetivo, que se utilizó en muchos de los primeros procesos de descomposición en MOEA.

2.2. Dos algoritmos relevantes

Aquí se expone brevemente el funcionamiento de los marcos de trabajo *MOEA/D* y *NSGA-III*. Ambos marcos de trabajo han sido seleccionados en virtud de ser los más representativos de sus respectivos enfoques de resolución (descomposición y dominancia, respectivamente), además de ser ampliamente citados y usados como referencia comparativa de las técnicas recién propuestas en el estado del arte.

2.2.1. Marco de trabajo MOEA/D

El *MOEA/D* fue desarrollado por Qingfu Zhang y Hui Li de la Universidad de Essex en 2007 [57].

Es un algoritmo de optimización inspirado en las técnicas de descomposición y los algoritmos genéticos. El proceso de descomposición requiere de un conjunto de vectores de pesos utilizado para transformar el MOP en un problema de optimización escalar donde cada peso indica la dirección de búsqueda en el espacio de los objetivos, que deben cumplir con estar distribuidos uniformemente en el espacio de las soluciones. Es requerida además, una función de escalarización que admita diferentes vectores de pesos. En principio, las funciones de sumas ponderadas (g^{ws}), de Chebyshev (g^{te}) y penalty boundary intersection (g^{pbi}) son usadas en la propuesta original del algoritmo y se definen como sigue:

1. La función de sumas ponderadas define el siguiente problema de optimización escalar:

$$\text{mín } g^{ws}(x|w) = \sum_{i=1}^k w_i f_i(x) \quad (2.8a)$$

s.a:

$$\sum_{i=1}^k w_i = 1 \quad (2.8b)$$

$$w_i \forall i \in \{1, \dots, k\} \geq 0 \quad (2.8c)$$

Esta función suele ofrecer buenos resultados cuando \mathcal{PF}_{true} es convexo, de lo contrario, no es capaz de representar cada punto del \mathcal{PF}_{true} .

2. La función de Chebyshev define el siguiente problema de optimización escalar:

$$\text{mín } g^{te}(x|w) = \max_{1 \leq i \leq k} w_i(f_i(x) - z_i^*) \quad (2.9)$$

Se ha probado que la función de pesos de Chebyshev puede generar cualquier punto en \mathcal{PF}_{true} sin importar la forma del frente de Pareto.

3. La función penalty boundary intersection (PBI) define el siguiente problema de optimización escalar:

$$\text{mín } g^{pbi}(x|w) = d_1 + \theta d_2, \quad (2.10a)$$

$$d_1 = \left| F(x) \cdot \frac{w}{\|w\|} \right| \quad (2.10b)$$

$$d_2 = \left\| F(x) - d_1 \frac{w}{\|w\|} \right\| \quad (2.10c)$$

Una vez seleccionada la función de escalarización es posible definir N sub-problemas de optimización escalares dados por N diferentes parametrizaciones o vectores de pesos establecidos previamente. La resolución de estos problemas se realiza de forma simultánea y colaborativa. Cada individuo de la población resuelve uno de los sub-problemas escalares, intercambiando información (al momento de la generación de hijos por cruza y del reemplazo de individuos) con las soluciones candidatas de los T problemas escalares más cercanos.

Como marco de trabajo, *MOEA/D* se distingue por tener las siguientes características:

- Una técnica de generación de vectores de pesos basada en un análisis geométrico sobre el simplejo del problema [38, 8, 37].
- El uso de un motor de búsqueda basado en Algoritmos Evolutivos en la mejora de soluciones garantiza que éstas mejoran hasta óptimos locales/globales. Esta propiedad está demostrada en el teorema de esquemas de Holland [25] para codificaciones binarias, y confirmada en el caso general de algoritmos genéticos con elitismo en [44].

Cuando se usa *MOEA/D* para problemas con frentes no continuos, muchos subproblemas se agrupan en los puntos de quiebre o extremos lo que significa un desperdicio de tiempo de computación [41] y un decremento de la diversidad.

Pseudocódigo MOEA/D

El pseudocódigo 2.2.1 muestra de forma general los elementos del marco de trabajo *MOEA/D*, el algoritmo requiere como datos iniciales la condición de paro, usualmente el número máximo de generaciones, los subproblemas a considerar, un tamaño de vecindario y los parámetros usuales de cruce y mutación de los AG.

Algoritmo 2.2.1: Algoritmo MOEA/D

Data: *IterMax*: condición de paro, *N*: subproblemas a considerar, *T*: Tamaño de vecindario, *P_{cruza}*, *P_{mutacion}*

Result: *PE*: Frente Aproximado

```

1 Poblacion ← PoblacionInicial(Poblacionsize, NumObjetivos);
2  $W = w_1, \dots, w_N \leftarrow$  VectoresDePesoUniformementeDistribuidos(N);
3  $B_0, \dots, B_N \leftarrow$  VecinosCercanos(T, W);
4 while  $\neg$ Paro(IterMax) do
5   for  $i := 1$  a N do
6     Selección ← SeleccionPadres( $B_i$ );
7      $C_i \leftarrow$  CruzayMutacion(Selección, Pcruza, Pmutacion);
8     if  $C_i \preceq x \in B_i$  then
9        $x \leftarrow C_i$ ;
10  PE ← ActualizarPE(Poblacion)
11 return (PE)

```

La línea numerada con 1 indica la inicialización de la población, la línea 2 asocia los subproblemas con los vectores de pesos que guiarán la búsqueda. En la línea 3 se ejecuta para cada vector de pesos la búsqueda de sus vecinos cercanos con el fin de establecer los nichos de búsqueda que serán utilizados durante la ejecución del algoritmo.

La línea 4 establece el ciclo principal asociado a la condición de paro. El ciclo interno (línea 5) contiene los elementos clásicos del algoritmo genético como son selección, cruce y mutación. La línea 8 es la condición que identifica si una solución es mejor que la actual en el vecindario, en caso de ser verdad, substituye la mejor solución asociada a un subproblema con la solución actual.

La línea 10 actualiza un archivo con las mejores soluciones encontradas hasta el momento, mismo archivo que es devuelto como resultado del algoritmo en la línea 11.

2.2.2. Marco de trabajo NSGA-III

El marco de trabajo *NSGA-III* fue propuesto por K. Deb y H. Jain en 2014 [9]. Utiliza los conceptos enunciados por Goldberg sobre ordenamiento basado en dominancia en algoritmos genéticos [19]. Es una adaptación de *NSGA-II* para mejorar su rendimiento en problemas con más de 3 objetivos.

Como en *NSGA-II*, en cada generación t , se toma como base una población padre (P_t) de tamaño N en el dominio del problema para producir utilizando las operaciones de selección, cruce y mutación genera una población descendiente (Q_t). Ambas poblaciones son combinadas y ordenadas en K niveles conforme a los criterios de dominancia (*Non dominated sorting*). Los N mejores miembros son seleccionados para formar la siguiente población padre (P_{t+1}), esto es tomando los miembros de los mejores $l - 1$ niveles de dominancia que aún no completan N individuos. Para completar los N miembros de P_{t+1} , es requerido un procedimiento para seleccionar a los individuos faltantes entre los miembros del nivel l . Aquí radica la diferencia fundamental entre *NSGA-III* y su predecesor. *NSGA-III* propone sustituir el operador *Crowding distance* (que evalúa la densidad de las soluciones en distintas regiones del espacio de los objetivos) por un método de nichos centrados sobre puntos de referencia en el espacio de los objetivos, de tal manera que ayuden a mejorar la diversidad de la población. Para esto, *NSGA-III* genera puntos de referencia utilizando un método propuesto por Das & Dennis en 1998 [8] y un diseño de dos capas en casos de altas dimensiones [9]. En simplex lattice design se definen K puntos estructurados sobre el simplejo M dimensional (al igual que se generan los pesos en *MOEA/D*).

Con los puntos de referencia y las magnitudes de las funciones objetivo normalizadas, se calcula la distancia ortogonal entre cada solución en el nivel l y se asocia a su punto de referencia más cercano. Se utiliza posteriormente un conteo de nichos para seleccionar a los miembros del nivel l en los nichos menos poblados. Esto se repite hasta cumplir el criterio de paro.

Pseudocódigo NSGA-III

El ciclo principal de *NSGA-III* es mostrado en el pseudocódigo 2.2.2 solo se muestra la parte central del algoritmo ejecutada durante cada generación. Son requeridos puntos de referencia estructurados en un simplejo unitario y una población padre, el resultado es la población inicial de la siguiente iteración.

Algoritmo 2.2.2: Algoritmo NSGA-III

```

Data:  $H$ : puntos de referencia estructurados  $Z^{(s)}$ , Población Padre  $P_t$ 
Result:  $P_{t+1}$ 
1  $S_t = \emptyset, i = 1;$ 
2  $Q_t = \text{Recombinacion} + \text{Mutacion}(P_t);$ 
3  $R_t = P_t \cup Q_t;$ 
4  $(F_1, F_2, \dots) = \text{FastNondominatedSort}(R_t);$ 
5 repeat
6    $S_t = S_t \cup F_i;$ 
7    $i = i + 1;$ 
8 until  $|S_t| \geq N;$ 
9  $F_i = F_i;$ 
10 if  $|S_t| == N$  then
11    $P_{t+1} = S_t;$ 
12   break;
13 else                                     /* Eleccion de los  $K$  puntos restantes de  $F_i$  */
14    $P_{t+1} = \bigcup_{j=1}^{i-1} F_j;$ 
15    $K = N - |P_{t+1}|;$ 
16    $\text{Normalizar}(f^n, S_t, Z^r, Z^s);$ 
17    $(\pi(s), d(s)) = \text{Asociar}(S_t, Z^r);$ 
18   for  $j \in Z^r$  do
19      $\rho_j = \sum_{s \in S_t \setminus f_j} (\pi(s) == j);$ 
20    $P_{t+1} = \text{Nichos}(K, \rho_j, \pi, d, Z^r, F_i, P_{t+1});$ 
21 return  $(P_{t+1})$ 

```

La línea 1 reinicializa los valores de los frentes no dominados y sus índices. La línea dos toma la población actual y genera un segundo conjunto de igual tamaño por medio de la recombinación y mutación. La línea 3 hace la unión de los ambos conjuntos (población actual P_t y nuevos individuos Q_t). La cuarta línea genera los llamados frentes no dominados sucesivos, de tal manera que el frente con número más bajo no es dominado por nadie, el siguiente solo es dominado por el anterior y así sucesivamente.

El ciclo de mostrado entre la quinta línea y octava línea genera la población siguiente S_t con los individuos presentes en los primeros frentes hasta alcanzar el tamaño fijo de población N además de incrementar un índice i al último frente agregado. En caso de completar los N individuos requeridos para la generación siguiente las líneas 10 a 12 asignan el valor de S_t a P_{t+1} y comienza una nueva generación.

De lo contrario, en las líneas 14 y 15, la unión de los frentes sucesivos desde 1 hasta el índice $i - 1$ se convierten en la primera parte de la población siguiente P_{t+1} y la diferencia entre

la cardinalidad de dicho conjunto y N es la cantidad de individuos restantes para ingresar a la población siguiente. La línea 16 normaliza los objetivos y genera una proyección de los puntos de referencia sobre un hiper plano que contiene las soluciones extremas de la población actual Z^r . En la línea 17 se asocian las soluciones en S_t con la proyección de referencia Z^r . Ésto es requerido para estimar la densidad de soluciones asociadas a los puntos de referencia mostrado en el ciclo de las líneas 18 – 19. Finalmente la población siguiente es completada agregando los K individuos en las regiones menos pobladas entregadas por la función de nichos.

2.3. Mantenimiento de la diversidad

Se han realizado varios esfuerzos enfocados a la mejora de la diversidad con ambas estrategias (descomposición con el *MOEA/D* y dominancia con el *NSGA-III*), la determinación de los vectores de pesos o puntos de referencia constituye claramente un factor crítico para obtener una buena diversidad del frente aproximado. Además de las estrategias mencionadas en la sección anterior, en las que los vectores/puntos de referencia se mantienen estáticos a lo largo de la búsqueda. Por otro lado, un área reciente de investigación consiste en modificar los puntos/vectores de referencia de forma dinámica durante el transcurso de la optimización, de tal manera que éstos se ajusten a la geometría del \mathcal{PF}_{true} :

- Kukkonen & Deb (2006) proponen el uso de una distancia de vecindario para determinar el nivel de dispersión de las soluciones, como criterio de “poda” del conjunto de soluciones no-dominadas, favoreciendo aquellas en regiones del espacio donde las soluciones son escasas [31].
- La idea introducida en [21] consiste en aproximar el frente de Pareto \mathcal{PF} utilizando una interpolación lineal de las soluciones no-dominadas como método de mejora de los vectores de pesos en *MOEA/D*.
- Jiang et al. (2011) proponen un parámetro l que calibra la generación de subproblemas lineales que maximizan el hipervolumen en una región, éstos son resueltos con el método simplex, de tal manera que los vectores resultantes modelan la convexidad de la región [30].
- Li et al. (2011) introducen una preselección de vectores de pesos de un conjunto σ generado con simplex lattice. Estos vectores de pesos son reevaluados con respecto a la distancia euclidiana a la solución no-dominada más cercana y en su caso reemplazados por algún vector en σ más cercano a dicha solución [33].
- En [29] desarrollan una distribución adaptativa de los puntos de referencia para asegurar la diversidad, reubicando los puntos referencia que no han sido asociados a soluciones factibles hacia posiciones vecinas donde se espera tengan soluciones asociadas.

- Yutao Qi et al. (2014) proponen una estrategia de modificación de los vectores de pesos, en base a la generación de medidas de dispersión promoviendo la generación de vectores adicionales en regiones con alta dispersión o, al contrario, la eliminación de ciertos vectores de regiones densamente pobladas [41].
- Yutao Qi et al. (2014), basándose en su trabajo anterior, redefinen el vecindario en términos de triangulación de Delaunay como alternativa a los k vecinos más cercanos, de tal manera que los triángulos tiendan a ser equiláteros separando y modificando los vectores de pesos en el transcurso del algoritmo [40].
- Cheng et al. (2016) propusieron un algoritmo evolutivo guiado por vector de referencia (RVEA) para la optimización de muchos objetivos (Many-Objective Optimization). En RVEA, la estrategia de adaptación de vectores de pesos se usa para tratar problemas con \mathcal{PF} irregular. En su estrategia, los vectores de pesos sin individuos asociados que se eliminan y se reemplazan con vectores generados aleatoriamente dentro del rango especificado. Obviamente, este método puede mejorar el rendimiento en problemas con \mathcal{PF} discontinuo. Según lo declarado por los autores, “sin embargo, dado que los vectores de referencia reemplazados son generados globalmente, la densidad de las soluciones local no está garantizada” [4].
- Cai et al. (2018) proponen dos tipos de ajustes a los vectores de pesos para problemas de muchos objetivos, uno de ellos dirigido al número de vectores de pesos y el otro a las posiciones de los vectores de pesos utilizando la dominancia de Pareto para identificar la efectividad de cada uno de ellos [3].
- Asafuddoula et al. (2018) utilizan las primeras generaciones como un periodo de aprendizaje para detectar regiones donde es requerido agregar más vectores de pesos y mantienen un archivo externo con los vectores iniciales para su futuro uso, es decir aun dependen fuertemente de la distribución de pesos original [2].
- Zhang et al. (2018) proponen una técnica de adaptación dinámica de vectores de pesos para problemas bi objetivo donde se aprovecha la característica de aglomeración de soluciones en los bordes discontinuos del \mathcal{PF} para identificar vectores que requieren ser substituidos y se generan nuevos vectores en base a la interpolación lineal teniendo excelentes resultados en términos de diversidad [55].
- Li & Yao (2018) desarrollan una técnica de adaptación dinámica de vectores de pesos utilizando un archivo de soluciones no-dominadas que aproximan el \mathcal{PF} para detectar regiones poco pobladas o prometedoras donde generar nuevos vectores a la vez que retira vectores en regiones muy pobladas [34].

A pesar de los estudios anteriores y de los progresos realizados en el área de ajuste dinámico de puntos de referencia usados en MOEAs, los principales problemas siguen vigentes:

- Adaptar vectores de pesos afecta la convergencia del algoritmo [17].
- Adaptar vectores de pesos puede afectar la convergencia de un algoritmo incluso con frentes continuos.
- La mayoría de las técnicas de adaptación de vectores de pesos funcionan bien sobre frentes con algunas características conocidas o supuestas, pero este desempeño no es siempre parejo para problemas con características diferentes.

Como se mencionó anteriormente, esta última área ha sido poco explorada y el presente trabajo propone el desarrollo de una estrategia para ajustar dinámicamente los vectores/puntos de referencia.

Investigación sobre las técnicas estáticas de generación de vectores/puntos de referencia

Como se explicó en el capítulo anterior, varios MOEAs recientes, como *MOEA/D* o *NSGA-III*, requieren en su modo operativo de un conjunto de puntos de referencia distribuidos uniformemente. El estudio de las distribuciones de puntos/vectores en el espacio multidimensional con propiedades predecibles ha generado un interés creciente por técnicas de generación de vectores de pesos, tanto por sus propiedades teóricas como por su desempeño en la práctica. Este tema se ha estudiado previamente en el marco de diseño de mezclas, donde se debe combinar un conjunto de m ingredientes para cumplir criterios específicos y cuyas composiciones respectivas deben sumar 1. Por otro lado, la teoría de la discrepancia, que trata de distribuir uniformemente un conjunto de vectores sobre un objeto matemático mediante la minimización de una función de discrepancia, ha aportado otra perspectiva para la generación de vectores de pesos.

Así, este capítulo provee los conceptos fundamentales sobre estas dos clases de métodos de generación, primero aquellos basados en conceptos geométricos y, por otro lado, técnicas que utilizan funciones de discrepancia. La Sección 3.1 cubre el problema de generación de vectores de pesos en los contextos de diseño de mezclas y secuencias de baja discrepancia. A su vez, introduce el diseño de simplex lattice y su variante de dos capas, además de exponer algunos problemas derivados de su uso en altas dimensiones. La Sección 3.2 incluye una introducción al método del diseño uniforme y la búsqueda tabú desarrollada para aproximar diseños uniformes. Las secuencias de baja discrepancia son descritas en la Sección 3.3. Finalmente, la Sección 3.4 cierra el capítulo con un breve resumen. Parte de los resultados en este capítulo han sido publicados en el artículo “A parallel tabu search heuristic to approximate uniform designs for reference set based MOEAs” [42] presentado en Proceedings of International Conference on Evolutionary Multi-Criterion Optimization (EMO2019).

3.1. Técnicas basadas en diseños de mezclas

Los experimentos con mezclas son experimentos en los cuales las variables son las proporciones de los ingredientes en una mezcla. Un ejemplo es el problema de determinar las proporciones de cada ingrediente en una mezcla de polímeros, con la meta de sintetizar productos plásticos con la mayor resistencia a la tracción. Experimentos similares son muy comunes en la industria. El problema de decidir cómo mezclar ingredientes para optimizar algún criterio se llama diseño experimental con mezclas. Un diseño de n corridas para mezclas de k ingredientes es un conjunto de n puntos en el dominio del simplejo unitario de dimensión k , T_k :

$$T_k = \{(\lambda_1, \dots, \lambda_k) : \forall j \in \{1, \dots, k\}, \lambda_j \geq 0 \wedge \lambda_1 + \dots + \lambda_k = 1\}. \quad (3.1)$$

Una gran variedad de trabajos en la literatura especializada introduce muchos tipos de diseños. El más conocido es el diseño Simplex Lattice, introducido por Scheffé en 1958 [46], junto con sus modelos polinomiales correspondientes. Posteriormente, el mismo autor desarrolló un diseño alternativo al enrejado general, el diseño centroide simplex. Más tarde, Cornell [7] sugirió el diseño axial y propuso una revisión exhaustiva de casi todos los artículos de estadística abordando los diseños experimentales con mezclas y análisis de datos. Para una descripción exhaustiva de estas técnicas, se refiere el lector a las fuentes antes citadas. A continuación, se presentan dos estrategias relevantes en el contexto de este trabajo, ya que han sido utilizadas con MOEAs clásicos.

3.1.1. Simplex Lattice

El diseño Simplex Lattice es un diseño de llenado de espacio que crea una cuadrícula triangular de puntos, que no incluye necesariamente el centroide. Desde el punto de vista de teoría de mezclas, se considera que una mezcla implica un número entero de k componentes (correspondiendo al número de objetivos), y además sea H un número entero positivo tal que la proporción de cada componente en la mezcla puede tomar $(H+1)$ posiciones equidistantes entre 0 y 1 (incluidos). Entonces, si λ_i es la composición de la mezcla en la componente i :

$$\forall i \in \{1, \dots, k\}, \lambda_i \in \left\{0, \frac{1}{H}, \frac{2}{H}, \dots, \frac{H-1}{H}, 1\right\}. \quad (3.2)$$

Este diseño se puede calcular de manera eficiente utilizando el enfoque sistemático de Das y Dennis [8] y se usa ampliamente en la mayoría de los MOEAs basados en puntos de referencia.

Para determinar n puntos sobre T_k son requeridos H y k , que son parámetros a ajustar. Entonces el número de componentes n es calculado como sigue:

$$n = \binom{H + k - 1}{k}. \quad (3.3)$$

Cada punto generado $\lambda^{(\ell)} = (\lambda_1^{(\ell)}, \lambda_2^{(\ell)}, \dots, \lambda_H^{(\ell)})$ con $1 \leq \ell \leq n$ satisface la restricción de convexidad:

$$\sum_{i=1}^H \lambda_i^{(\ell)} = 1. \quad (3.4)$$

El diseño Simplex Lattice se ilustra en la Figura 3.1 para $k = 3$ dimensiones, mientras que en la Figura 3.2 se muestra con gráficas de coordenadas paralelas para $k = 5$ y $k = 8$.

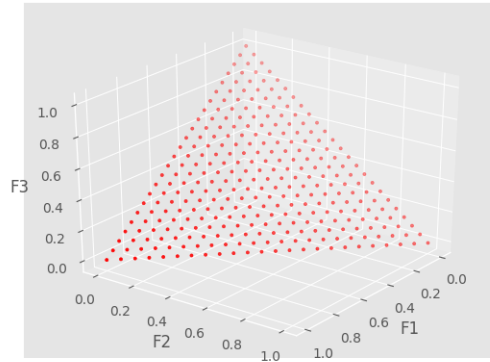


Figura 3.1: Hiperplano unitario generado con $H = 23$, $k = 3$ y el método de Das & Dennis.

3.1.2. Simplex Lattice de dos capas

Un reconocido inconveniente del diseño de Simplex Lattice es que genera muchos puntos en la frontera (las orillas) del simplejo [14, 47], particularmente cuando aumenta el parámetro H o la cantidad de dimensiones k . Para evitar este problema, Deb & Jain proponen en [9] la unión de dos conjuntos generados con Simplex Lattice: el primer conjunto mantiene la forma y el tamaño originales del Simplex Lattice (capa externa) mientras que el otro conjunto es escalado según un factor $f \in (0, 1)$, para cubrir la región central del simplejo (capa interna). Este nuevo diseño se introdujo originalmente dentro del marco *NSGA-III*, para producir los puntos de referencia utilizados para tratar problemas con muchos objetivos.

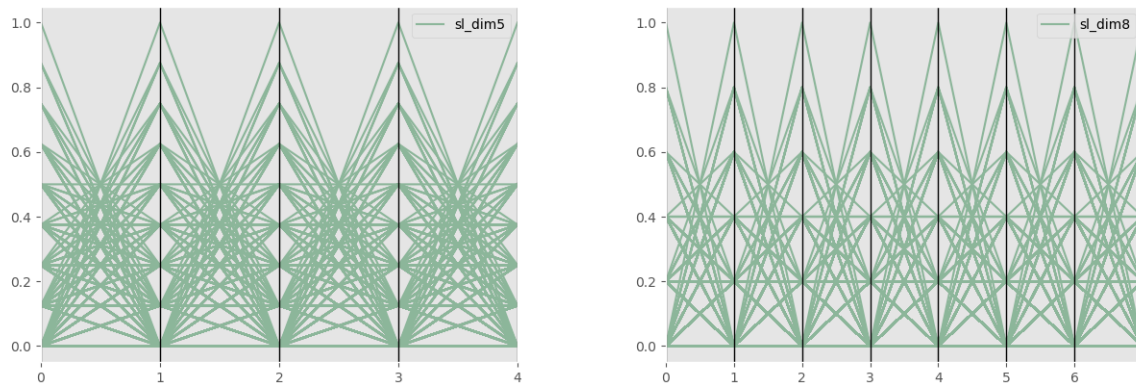


Figura 3.2: Hiperplanos unitarios en $k = 5$ y $k = 8$ generados con el método de Das & Dennis expresados en gráficas de coordenadas paralelas.

La serie de gráficas de la Figura 3.3 muestra, para tres dimensiones, las capas interna y externa, que se combinan en la última gráfica para construir el diseño final de dos capas. En la Figura 3.4 se ilustra con gráficas de coordenadas paralelas para $k = 5$ y $k = 8$.

Las siguientes secciones presentan la segunda clase de diseños, basada en medidas de discrepancia y que, al contrario de los enfoques geométricos (como los diseños de Simplex Lattice sencillo y de dos capas), utiliza alguna función midiendo la “disimilitud” entre los puntos de un conjunto. Bajo este enfoque, una menor discrepancia indica un mejor cubrimiento del espacio geométrico de interés (aquí, el simplejo unitario). Particularmente, se abordan dos técnicas: el Diseño Uniforme y los diseños basados en secuencias de baja discrepancia.

3.2. Diseño Uniforme

En 1980, Fang [12] propone el Diseño Uniforme (UD) como un tipo de diseño estadístico experimental. El UD consiste en construir los puntos experimentales de tal manera que estén uniformemente distribuidos sobre un dominio experimental, de acuerdo a una medida de discrepancia [14]. Se demostró en [50] que el Diseño Uniforme presenta la ventaja de ser robusto a los cambios en el modelo y el reetiquetado de los ejes.

Sea k el número de factores de un experimento (o número de objetivos en el contexto de optimización multiobjetivo) sobre el dominio estándar T_k y q es el número de posibles valores que cada factor puede tomar. El objetivo es elegir un conjunto de n vectores $P_n = \{\lambda^1, \dots, \lambda^n\} \subset T_k$ tal que estos vectores se dispersen uniformemente sobre T_k . Sea $D(P_n)$ una medida de la no-uniformidad (i.e., discrepancia) de P_n . El objetivo es determinar un conjunto

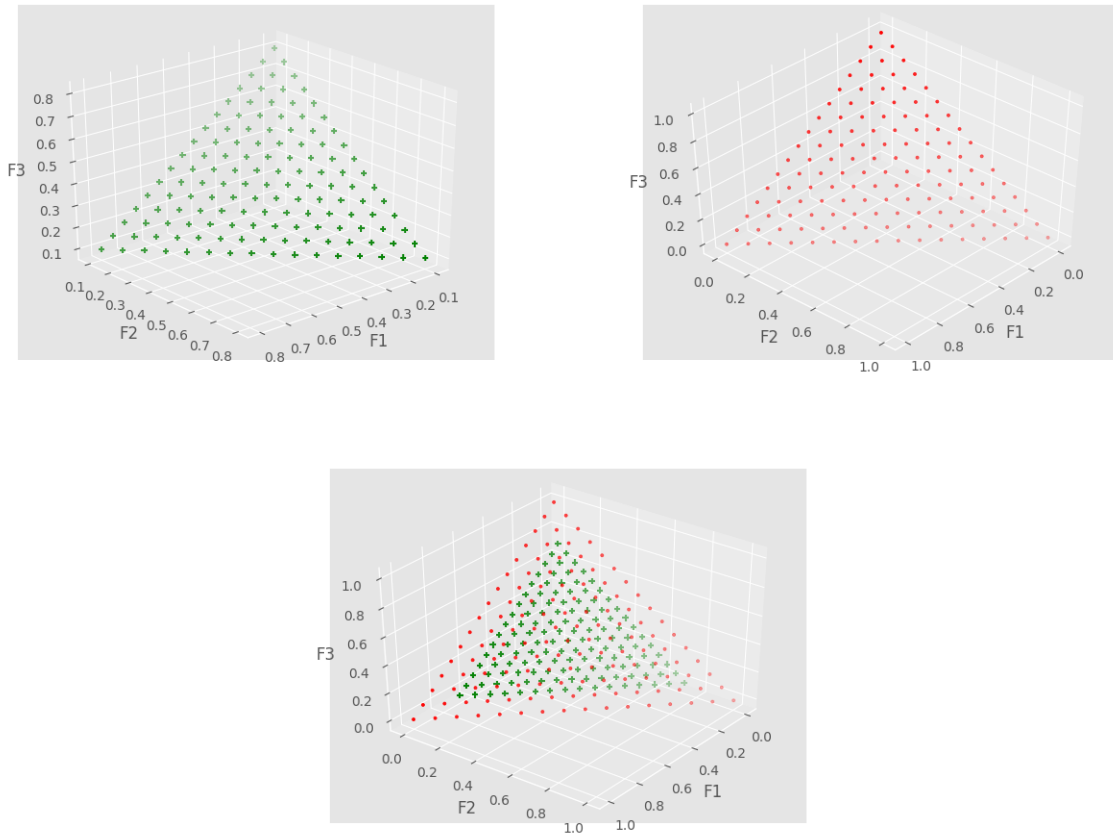


Figura 3.3: Hiperplano unitario generado con el método de Simplex lattice de dos capas.

P_n^* minimizando D o, de manera equivalente, maximizando la uniformidad de n puntos sobre T_k . El problema de optimización asociado se puede describir de la siguiente manera:

$$\begin{aligned} \min D(P_n) \\ \text{s.t. } P_n \in T_k \end{aligned} \tag{3.5}$$

El elemento base del UD, asociado al conjunto P_n , es el diseño de tipo U . Un diseño de tipo U es una matriz U_{n,q^k} de tamaño $n \times k$, en la que cada columna tiene q posibles entradas, $1, \dots, q$, apareciendo igual cantidad de veces. La matriz inducida de U , llamada $X = (x_{ij})$, es definida como $x_{ij} = \frac{u_{ij}-0.5}{q}$. Entonces, un diseño de tipo U , U_{n,q^k} , se convierte en un UD cuando su matriz inducida X_{n,q^k} tiene la menor discrepancia en el conjunto de todas las matrices inducidas X posibles en el dominio T_k .

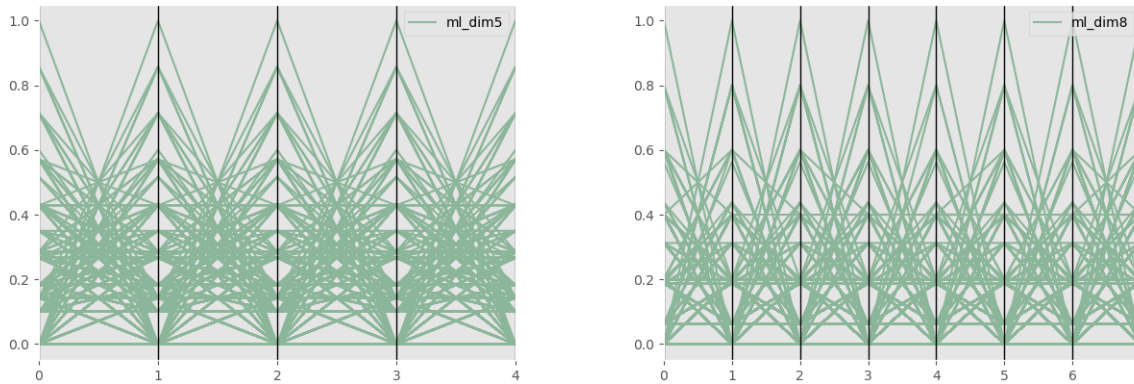


Figura 3.4: Hiperplanos unitarios en $k = 5$ y $k = 8$ generados con el método de Simplex lattice de dos capas.

Fue demostrado que el problema de encontrar un diseño uniforme, bajo una métrica de discrepancia dada D , es NP-duro cuando el número de corridas/puntos n tiende a $+\infty$ y el número de factores/dimensiones k es mayor que 1 [13]. Debido a esta complejidad, se han propuesto muchos métodos constructivos para producir diseños uniformes o casi uniformes. Entre ellos destacan, por ejemplo, las secuencias de baja discrepancia (véase Sección 3.3), o el método *Good Lattice Point* (GLP), descrito en detalle a continuación ya que la estrategia desarrollada en este trabajo de investigación para producir diseños casi uniformes adopta un enfoque basado en GLP.

El método *Good Lattice Point* (GLP) fue originalmente propuesto por Korobov en 1959 para la evaluación numérica de integrales multivariadas [48]. Entre algunas de las aplicaciones de los conjuntos de puntos generados con este método, se encuentran los métodos Cuasi-Monte Carlo, diseños uniformes y experimentos computacionales. Para generar una matriz de tipo U utilizando GLP, el primer paso consiste en determinar un conjunto $H_n \subset \mathbb{N}$, tal que sus elementos son menores a n y su máximo común divisor (MCD) es 1 (son coprimos). Esto se expresa en la Ecuación 3.6:

$$H_n = \{h : h < n | \text{MCD}(h, n) = 1\} \quad (3.6)$$

Si $w = |H_n|$, entonces para cualesquiera k elementos distintos seleccionados de $H_n = \{h_1, \dots, h_w\}$, se puede generar una matriz $U = (u_{ij})$ de tamaño $n \times k$ donde $u_{ij} = i \cdot ((h_j \text{ mód } n) + 1)$. Finalmente, la última fila de la matriz generada por GLP debe eliminarse para obtener la matriz final de tipo U .

Como se mencionó antes, una vez que se tiene una matriz de tipo U , es necesario evaluar

la calidad de dicha matriz mediante alguna función de discrepancia que mide la no-uniformidad del conjunto P_n correspondiente. En la literatura especializada hay más de una definición de discrepancia D , por ejemplo: la discrepancia en estrella (D^*), la discrepancia centrada en L_2 (CD , la más utilizada) o el ajuste alrededor de la L_2 -discrepancia (WD). La discrepancia L_2 centrada, denotada por CD_2 , se usa en la implementación de UD, porque su proceso de computación no es complejo y porque esta técnica es invariante al reetiquetado de los ejes de coordenados, es decir, es más robusta a los cambios. A continuación se presenta esta medida, utilizada en el presente trabajo; las fórmulas correspondientes para otras discrepancias se pueden encontrar en [11].

Como consecuencia de las explicaciones anteriores, el problema de determinar un diseño uniforme se puede definir utilizando la matriz inducida X de U de la siguiente manera:

$$\begin{aligned} \operatorname{argmin} CD_2(X) = & \left(\frac{13}{12}\right)^m - \left(\frac{2}{n}\right) \sum_{k=1}^n \prod_{i=1}^m \left(1 + \frac{1}{2}|X_{ki} - \frac{1}{2}| - \frac{1}{2}|X_{ki} - \frac{1}{2}|^2\right) \\ & + \left(\frac{1}{n^2}\right) \sum_{k=1}^n \sum_{j=1}^n \prod_{i=1}^m \left(1 + \frac{1}{2}|X_{ki} - \frac{1}{2}| + \frac{1}{2}|X_{ji} - \frac{1}{2}| - \frac{1}{2}|X_{ki} - X_{ji}|\right). \end{aligned} \quad (3.7)$$

La matriz U tiene que construirse de alguna manera, por ejemplo mediante el método GLP. El diseño de tipo U , U^* , cuya matriz inducida X^* tiene una discrepancia mínima sobre la medida de discrepancia CD_2 se denomina diseño uniforme. Además, un diseño de tipo U con una medida de discrepancia baja se puede denotar como un diseño casi uniforme [35].

Tan et al. [47] introducen el uso del diseño uniforme como vectores de pesos en $MOEA/D$ con resultados prometedores, pero restringiendo el número de objetivos a $2 \leq k \leq 4$, principalmente por la complejidad asociada al cálculo de diseños uniformes para un mayor número de corridas y dimensiones.

3.2.1. Motivación y descripción de la técnica de generación de diseños casi uniformes

El enfoque adoptado en este trabajo consiste en resolver el problema de optimización definido en la Ecuación 3.2. para producir una matriz de tipo U , se utiliza el método GLP que es computacionalmente costoso, por lo que la evaluación de una solución del problema de optimización requiere de un tiempo considerable, alrededor de 32 segundos. Además, el tamaño de espacio de solución aumenta drásticamente con el número de dimensiones k y de puntos generados n (número de combinaciones de k elementos seleccionados del conjunto H_n). Para $k = 8$, $n = 800$ y $|H_n| = 320$ recorrer las 2, 496, 767, 727, 736, 920 entradas del

espacio de búsqueda tardaría 2,533,503,528.906058 años con la computadora disponible para el proyecto.

Por lo tanto, en este trabajo en particular, se selecciona un método de optimización metaheurístico (para poder controlar el tiempo de cómputo) y basado en trayectorias (métodos poblacionales como algoritmos genéticos ocasionan que el tiempo de computación se vuelva prohibitivo) para resolver el problema de encontrar diseños casi uniformes [35, 13]. Además, una implementación paralela permite distribuir varios procesos de búsqueda en diferentes procesadores, con el fin de mejorar la exploración con un tiempo de cálculo razonable.

La Búsqueda Tabú es una técnica metaheurística introducida en la década de los 80 por Fred Glover [18]. Es una extensión simple del método de ascenso a la montaña, con características adicionales para evitar quedar atrapado en los óptimos locales:

- Se realiza una evaluación exhaustiva del vecindario de la solución actual para identificar al mejor vecino y poder concluir eventualmente sobre la optimalidad local de esta solución en el vecindario considerado. Independientemente de su calidad, el mejor vecino siempre se acepta como la nueva solución.
- Las mejores soluciones siempre se guardan como soluciones élite, sin embargo, no inciden en la selección de las trayectorias de búsqueda actuales.
- Dado que el modo de operación mencionado anteriormente puede llevar a ciclos en la trayectoria de búsqueda, una lista tabú registra las últimas soluciones visitadas, o los últimos movimientos realizados, con el fin de prohibir un retorno a esas soluciones (o los movimientos inversos que conducirían a soluciones ya visitadas). Por lo tanto, el próximo candidato es elegido entre aquellos vecinos de la solución actual que no son tabú.
- Cuando la lista tabú informa sobre movimientos prohibidos, generalmente se incluye un criterio de aspiración para romper el estado tabú de ciertos movimientos, que podrían conducir a soluciones no visitadas.
- La lista tabú puede truncarse para respetar el tamaño permitido, eliminando los elementos más antiguos.

3.2.2. Implementación

La implementación desarrollada en este trabajo incluye el *Good Lattice Point* como un mecanismo integrado para generar el diseño de tipo U con una entrada específica. Como se menciona en la sección anterior, k elementos se seleccionan de un conjunto H_n definido en la

Ecuación 3.6, para generar una matriz de tipo U , su matriz inducida X y calcular la discrepancia correspondiente $CD_2(X)$ (consulte la Ecuación 3.8). Así, el número de entradas posibles crece rápidamente con k y con el número de puntos generados, n , ya que es igual a $\binom{|H_n|}{k}$, donde $|H_n|$ crece de manera cuadrática con n , para $k = 8$ dimensiones, se elige por ejemplo $n = 800$, entonces $|H_n| = 320$ y el número de entradas posibles es 2, 496, 767, 727, 736, 920.

Por lo tanto, en la heurística paralelizada basada en Búsqueda Tabú que se que se implementó en este trabajo, la codificación de la solución es simplemente la entrada de la técnica GLP, es decir, un conjunto de k índices que indican los elementos de H_n a seleccionar para calcular el diseño de tipo U . Así, una solución S se puede representar como $S = \{s_1, \dots, s_k\}$ con, $\forall i, j \in \{1, \dots, k\}, s_i \in H_n$ y $s_i \neq s_j$. Para la inicialización, las soluciones son aleatorias. Cuando se debe realizar un movimiento, se elige un índice aleatorio $j \in \{1, \dots, k\}$ y s_j se reemplaza por un elemento extraído al azar de H_n , diferente de todos los elementos s_i en la solución actual. El resto de la implementación propuesta es canónica, sólo algunos procesos fueron modificados.

En primer lugar, con el fin de promover la exploración del espacio de búsqueda, el proceso se distribuye entre W trabajadores, o agentes. Todos los agentes trabajan independientemente unos de otros y se asignan a su propio procesador para realizar búsquedas paralelas asíncronas. La única información que comparten es la lista tabú, que es común para todos ellos para evitar múltiples evaluaciones de la misma solución. Por lo tanto, algunos procedimientos, como la producción de nuevas soluciones (que hace uso de la lista tabú) o el truncamiento de la lista tabú, no se pueden realizar simultáneamente para varios agentes (estos procesos se identifican con el comentario "Sección crítica" en el siguiente pseudocódigo).

Además, la razón para compartir la lista tabú entre todos los agentes es que cada uno informa la solución completa $S = \{s_1, \dots, s_k\}$, en lugar de registrar los elementos involucrados en un movimiento como se hace comúnmente en muchas implementaciones de búsqueda tabú. Esto es posible debido al uso razonable de la memoria para almacenar este vector (k es igual a 8 como máximo en este estudio). Por lo tanto, los agentes comparten la misma lista tabú para evitar evaluar nuevamente una solución ya visitada por otro agente. Finalmente, no se necesita un criterio de aspiración, ya que la información contenida en la lista tabú (soluciones en lugar de movimientos) no impide considerar soluciones no visitadas.

La última característica de la presente implementación es que, debido al tamaño antes mencionado del espacio de búsqueda, los miles de vecinos de una solución no se pueden generar ya que su evaluación implicaría un tiempo de cálculo excesivo. Para reducir lo más posible esta carga computacional, solo se generan dos vecinos (y el mejor reemplaza a la solución actual, aunque esto deteriore el valor de la función objetivo).

Las mejores soluciones encontradas por los W trabajadores se mantienen en una lista élite que será evaluada de elemento a elemento para retomar a la mejor solución encontrada

globalmente en la ejecución del algoritmo.

Algoritmo 3.2.1: Búsqueda Tabú paralela

Data: $IterMax$: Condición de paro, W : Trabajadores, T_{max} : tamaño lista tabú, n : Número de puntos esperados, dim : número de dimensiones

Result: NUD : Nearly Uniform Design

```

1  $H_n \leftarrow$  BuscarCoprinos( $n$ );
2 Soluciones  $\leftarrow$  ConjuntosDisjuntosAleatorios( $H_n, dim, W$ );
3  $bestdis \leftarrow \infty$ ,  $bestsol \leftarrow \emptyset$ ,  $TabuList \leftarrow \emptyset$ ;
4 while  $\neg$ Paro( $IterMax$ ) do
   | /* Parallel For */
5   for  $i := 1$  to  $W$  do
6     |  $U_i \leftarrow$  GLP( $Soluciones_i$ );
7     |  $X_i \leftarrow$  MatrizInducida( $U_i$ );
8     |  $discrepancia_i \leftarrow CD_2(X_i)$ ;
9     | if  $discrepancia_i < bestdis_i$  and  $Soluciones_i \notin TabuList$  then
10    | |  $bestsol_i \leftarrow Soluciones_i$ ;
11    | |  $bestdis_i \leftarrow discrepancia_i$ ;
12    |  $TabuList \leftarrow TabuList \cup Soluciones_i$ ;
13    | /* Sección crítica */
14    |  $Soluciones_i \leftarrow$  UpdateSol( $Soluciones_i, H_n$ );
15    | if  $|TabuList| > T_{max}$  then
16    | |  $TabuList \leftarrow$  MantenerListaTabu( $TabuList$ );
17  |  $NUD \leftarrow$  mín( $Soluciones, bestdis$ );
18 return ( $NUD$ )

```

El proceso global se describe en el Algoritmo 17. Primero, el conjunto H_n se genera solo una vez, al comienzo del procedimiento (línea 1). Luego, la solución de cada agente se crea seleccionando k (dim en el pseudocódigo) elementos distintos de H_n , ya sea al inicio de la búsqueda (línea 2: W produce diferentes soluciones) o cuando se produce un vecino de la solución actual (línea 13). Posteriormente, para cada agente, el diseño de tipo U resultante se genera con la estrategia GLP utilizando como entrada los k elementos seleccionados de H_n (línea 6). La matriz inducida correspondiente X_i y el valor $CD_2(X_i)$ se calculan en las líneas 7-8 para evaluar la discrepancia del diseño generado. La mejor solución encontrada por cada trabajador y la lista tabú se actualizan en las líneas 9-12.

Luego, dentro de la “sección crítica”, las siguientes operaciones no pueden realizarse simultáneamente ya que implican acceso a la lista de tabú compartida. Cada trabajador produce dos nuevos candidatos en el vecindario de su solución actual (línea 13) y avanza hacia la mejor

solución en términos de CD_2 . Esto podría causar un deterioro de la función objetivo actual, sin embargo, la mejor solución encontrada por el trabajador se mantiene en una lista externa para evitar su pérdida. Finalmente, la lista tabú puede truncarse para respetar el tamaño permitido, eliminando los elementos más antiguos.

3.2.3. Experimentos computacionales y resultados

Esta sección presenta los experimentos realizados para obtener diseños casi uniformes resolviendo el problema de optimización con el algoritmo de búsqueda tabú paralela descrito en la subsección anterior. Primero, se llevaron a cabo 10 ejecuciones para el caso de 3 objetivos para evaluar la robustez de la técnica de solución propuesta. Además, para este caso, la discrepancia mínima se puede conocer a través de una evaluación exhaustiva del espacio de búsqueda, de tamaño razonable ya que si se desean generar 100 puntos para 3 objetivos, el número de combinaciones asociadas a las entradas del método GLP es 2300. El valor de la mejor solución es $CD_2^* = 2.835514900301206e^{-05}$, lo que permitirá evaluar la calidad de las soluciones. El ajuste de parámetros para este experimento es: $W = 4$ trabajadores, $IterMax = 5,000$ iteraciones (lo que implica 40,000 evaluaciones del objetivo, ya que se generan 2 vecinos por iteración) y un tamaño de lista tabú establecido en 1,000 soluciones.

En 10 ejecuciones, la heurística paralela identifica 6 veces la solución óptima, mientras que el valor medio de CD_2 es igual a $2.8355149004366532e^{-05}$ (es decir, un incremento de $4.77e^{-9}$ % con respecto al óptimo) y la desviación estándar es igual a 6.53^{-16} , lo que demuestra que el algoritmo propuesto puede encontrar de manera robusta diseños casi uniformes. El tiempo promedio de una ejecución es de 20 minutos, cuando se realiza en un equipo HP ProLiant BL465c G7 con 24 procesadores AMD Opteron (tm) 6174 a 2.2 GHz y 128 GB en RAM.

Para las dimensiones 5 y 8, los parámetros son: $W = 25$, $IterMax = 50,000$ (lo que significa 2.5×10^6 evaluaciones de función) y un tamaño de lista tabú igual a 10,000. Esto implica un tiempo de cómputo mucho mayor al caso de 3 dimensiones: aproximadamente 7 horas y 6 días para 5 y 8 objetivos, respectivamente. Por lo tanto, sólo fue posible, debido a limitaciones de tiempo, realizar una ejecución y no se obtuvieron resultados estadísticos.

Sin embargo, las soluciones finales pueden compararse visualmente con las obtenidas con los diseños Simplex Lattice (SLD) y de dos capas (2LD). La Figura 3.5 muestra la distribución de los conjuntos de puntos resultantes en gráficos de coordenadas paralelas, así como el hipervolumen HV correspondiente (usando 1.18^k como punto de referencia).

Las gráficas de coordenadas paralelas ilustran una tendencia esperada: la SLD produce muchos puntos en el límite del simplejo, alcanzando los puntos para cada objetivo. Por otro

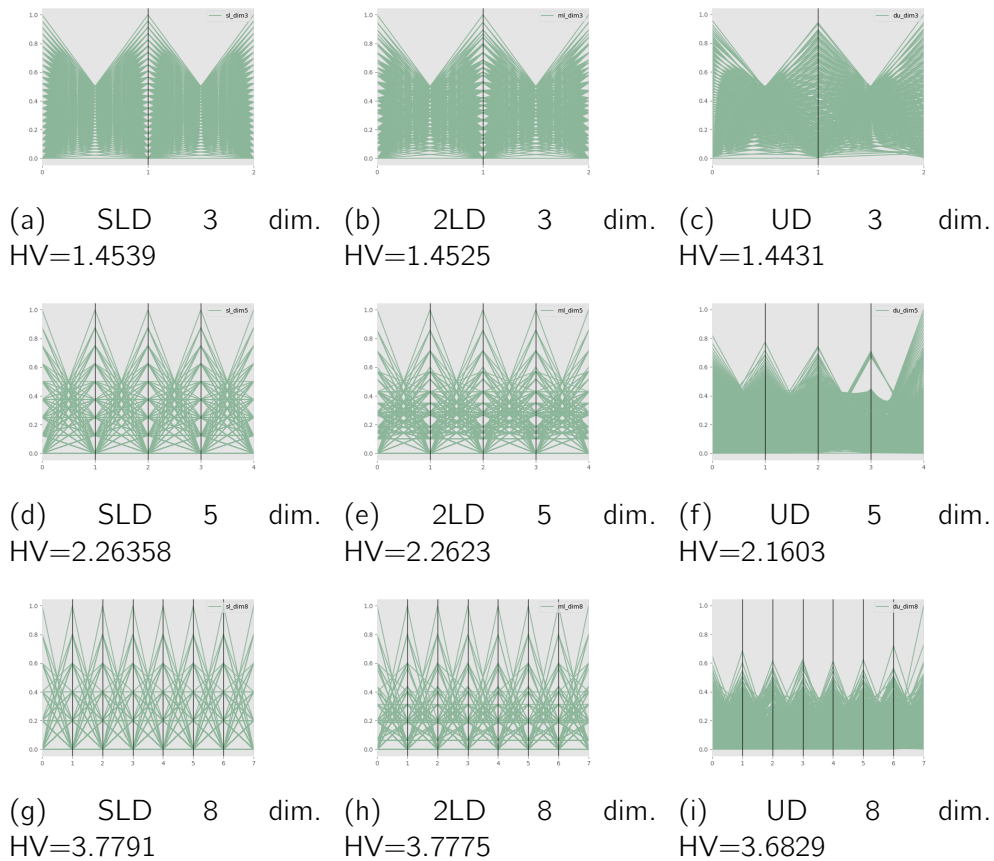


Figura 3.5: Gráficas de coordenadas paralelas e Hipervolumen para distintas técnicas de generación de puntos de referencia.

lado, 2LD también tiene muchos puntos en el límite del simplejo (capa externa), aunque el número de puntos dentro del simplejo es mayor que en el caso de SLD (debido a la capa interna). Al contrario, los diseños casi uniformes obtenidos pierden casi sistemáticamente los puntos extremos (este comportamiento se acentúa cuando la dimensión crece), pero tiene una distribución mucho más densa en las partes internas del simplejo. Como consecuencia, podría esperarse que los diseños casi uniformes calculados aquí con la heurística paralela proporcionen mejores estimaciones para muchos objetivos. Es de tomar en cuenta que la misma tendencia es capturada por el indicador HV, que siempre es mejor para la SLD.

3.3. Secuencias de baja discrepancia

Como ya fue mencionado, la teoría de la discrepancia (también llamada teoría de irregularidades de distribución) es una rama de las matemáticas que aborda el problema de distribuir puntos de manera uniforme sobre algún objeto geométrico y medir los errores sobre dicha distribución. Esta teoría fue iniciada por contribuciones teóricas como el teorema de equidistribución de Weyl y el teorema de Roth [43]. La discrepancia de un conjunto de puntos mide la no uniformidad de dichos puntos colocados (sin pérdida de generalidad) en un cubo unitario $[0, 1]^k$, donde $k > 0$ denota la dimensión del cubo unitario.

Las secuencias de baja discrepancia también se denominan secuencias cuasi-aleatorias o sub-aleatorias, debido a su uso común como reemplazo de números aleatorios distribuidos uniformemente. Al igual que las estrategias de *Good Lattice Point* y Diseño Uniforme, son capaces de producir conjuntos con discrepancias típicamente bajas. Esto se demuestra en [26], particularmente en el caso del método de *Good Lattice Point*. Existe un gran número de secuencias de baja discrepancia, muchas son mejoras de la secuencia de van der Corput [24] (distribución sobre una recta). Por ejemplo, la secuencia de Halton, al igual de la de Hammersley, extienden el desarrollo de la secuencia de Van Der Corput a múltiples dimensiones, mientras que la secuencia de Sobol realiza una partición binaria en el espacio definido por el cubo unitario.

La ventaja de estos métodos es que son deterministas y tienen un tiempo de cálculo reducido. Sin embargo, la única garantía que proveen estos métodos en términos de discrepancia es que ésta solo será baja con respecto a alguna medida de discrepancia, como por ejemplo CD_2 . Lo anterior podría implicar que existirán situaciones no deseadas en cuestiones de diversidad como puntos demasiado cercanos o muy alejados entre sí.

Halton

En una dimensión, la secuencia de van der Corput [24] tiene base 2, comienza desde cero y está confinada al intervalo $[0, 1)$. Sea n un número entero representado en la base b y L la longitud de dicha representación, el n -ésimo número de van der Corput se define como sigue:

$$g_b(n) = \sum_{j=0}^{L-1} d_j(n) \cdot b^{-j-1}, \quad (3.8)$$

donde d_j es el valor en la posición j de n .

El método de Halton es la extensión de esta secuencia para más de una dimensión. Es construida por un método determinístico que usa números coprimos como bases de secuencias de van der Corput para cada componente dimensional. Además, cada una de estas bases tiene que ser un número coprimo con respecto a la cantidad de puntos a construir. El conjunto de los primeros n números en la secuencia de Halton se pueden describir como:

$$Halton(n) = \{g_{b_1}(n), \dots, g_{b_k}(n)\}, \quad (3.9)$$

donde k es una dimensión arbitraria y b_1, \dots, b_k son números coprimos arbitrarios que cumplen con ser mayores que 1.

A pesar de que las secuencias de Halton estándar se desempeñan muy bien en dimensiones bajas, se han observado problemas de correlación entre secuencias generadas a partir de primos superiores. La secuencia de Halton empieza a exhibir una estructura para altas dimensiones ($k \geq 14$) donde su distribución es insatisfactoria en términos de discrepancia.

Hammersley

Hammersley es también una adaptación de la secuencia de van der Corput a más de una dimensión y se construye con base a números coprimos, b_1, \dots, b_{k-1} , tal que $b_i > 1, \forall i \in \{1, \dots, k-1\}$. Conociendo k y N , número de puntos a generar, cada miembro en el conjunto de Hammersley k dimensional de tamaño N se define por [22]:

$$Hammersley(n) = \{g_{b_1}(n), \dots, g_{b_{k-1}}(\frac{n}{N})\}, \quad (3.10)$$

donde k es una dimensión arbitraria y b_1, \dots, b_{k-1} son números coprimos arbitrarios que cumplen con ser mayores que 1. Cabe recalcar que una secuencia de Hammersley es idéntica a la de Halton salvo que sustituye la última dimensión por un barrido lineal.

Sobol

La secuencia de Sobol tiene la misma base para todas las dimensiones (base 2), realizando particiones binarias del espacio del cubo unitario $[0, 1]^k$ y reordenando los elementos vectoriales dentro de cada dimensión. Este proceso es complejo y se basa en un conjunto de “números de dirección”, $\{v_k\}$ dados por:

$$v_k = \frac{p_k}{2^k}, \quad (3.11)$$

donde p_k son los números nones menores a 2^k y v_k es elegido para satisfacer una relación de recurrencia utilizando los coeficientes de un polinomio primitivo en el campo binario de Galois GF_2 [16]. Los puntos generados con Sobol suelen mantener sus propiedades en altas dimensiones.

Zapotecas et al. [52] introducen el uso de secuencias de baja discrepancia como vectores de pesos en el marco de trabajo *MOEA/D*.

3.4. Resumen

Este capítulo introdujo los conceptos básicos de generación de vectores de pesos en el contexto de diseño de mezclas y estadística combinatoria. Se encuentra que métodos para generar vectores de pesos dentro del espacio experimental T_k pueden responder a las necesidades específicas del problema, aunque tratar casos de alta dimensionalidad afecta las distribuciones que tienden a tener un buen desempeño en dimensión 2 y dimensión 3.

Algunos métodos que suelen generar buenas distribuciones en el espacio de muchas dimensiones suelen ser costosos en tiempo de cómputo, sin embargo, es posible aproximar esas distribuciones (y sus propiedades) por medio de distintas técnicas, en este trabajo se elige una variación de la Búsqueda Tabú para generar diseños casi uniformes. Otras secuencias bien conocidas como Halton, Sobol y Hammersley son usadas para generar un conjunto base de experimentos que permiten estudiar el impacto de la dimensionalidad en distintos grupos problemas.

Todas estas técnicas ya han sido empleadas en el contexto de MOEAs, como herramienta para mantener diversidad. Se les denota a partir de ahora como estáticas, ya que se mantienen sin cambiar durante una ejecución del MOEA.

El siguiente capítulo presenta un estudio experimental exhaustivo para determinar y comparar el desempeño de estos métodos, en diferentes condiciones (dimensionalidad, paradigma

de búsqueda, geometría del frente de Pareto).

Resultados experimentales sobre técnicas estáticas de generación de vectores/puntos de referencia

En este capítulo, se definen las funciones de prueba utilizadas para realizar los experimentos computacionales y se presentan los resultados obtenidos, que se discuten y analizan en un segundo tiempo.

4.1. Funciones de prueba

En esta sección comparamos las distintas técnicas de generación estática de vectores de pesos en funciones de prueba bien conocidas, $DTLZ1$, $DTLZ2$ y $DTLZ7$ de CEC'09 [56], Kite (Papalote) [53], y las funciones invertidas $DTLZ1^{-1}$, $DTLZ2^{-1}$ y $DTLZ7^{-1}$ [28]. La decisión de tomar este conjunto de funciones de prueba tiene como principal objetivo comprobar que los vectores de pesos y las distintas geometrías del frente de Pareto influyen significativamente en el resultado final de los algoritmos clásicos $MOEA/D$ y $NSGA-III$.

4.1.1. Funciones de prueba Deb-Thiele-Laumanns-Zitzler

La suite de problemas de prueba Deb-Thiele-Laumanns-Zitzler [56] incluye nueve problemas que son escalables a cualquier número k de objetivos. Para este estudio se han seleccionado tres funciones representativas cuyas geometrías del \mathcal{PF}_{true} son de interés para el presente estudio.

DTLZ1

La dificultad que presenta este problema es la convergencia al frente de Pareto real \mathcal{PF}_{true} . El espacio de búsqueda contiene frentes óptimos de Pareto locales, donde un MOEA puede quedar atrapado antes de alcanzar el frente óptimo de Pareto global.

$$\begin{aligned}
 f_1(\mathbf{x}) &= 0.5(1 + g(\mathbf{y})) \prod_{i=1}^{k-1} x_i \\
 f_{j=2:k-1}(\mathbf{x}) &= 0.5(1 + g(\mathbf{y}))(1 - x_{k-j+1}) \prod_{i=1}^{k-1} x_i \\
 f_k(\mathbf{x}) &= 0.5(1 + g(\mathbf{y}))(1 - x_1) \\
 g(\mathbf{y}) &= 100 \left(\left(n + \sum_{i=1}^n (y_i - 0.5)^2 - \cos(20\pi(y_i - 0.5)) \right) \right)
 \end{aligned}$$

DTLZ2

Es una función clásica con una geometría cóncava del frente de Pareto óptimo.

$$\begin{aligned}
 f_1(\mathbf{x}) &= 0.5(1 + g(\mathbf{y})) \prod_{i=1}^{k-1} \cos x_i \frac{\pi}{2} \\
 f_{j=2:k-1}(\mathbf{x}) &= (1 + g(\mathbf{y})) \left(\prod_{i=1}^{k-j} \cos x_i \frac{\pi}{2} \right) \sin x_{k-j+1} \frac{\pi}{2} \\
 f_k(\mathbf{x}) &= 0.5(1 + g(\mathbf{y})) \sin x_1 \frac{\pi}{2} \\
 g(\mathbf{y}) &= \sum_{i=1}^n (y_i - 0.5)^2
 \end{aligned}$$

DTLZ7

El \mathcal{PF}_{true} es disconexo y cuenta con un número de componentes conexos que depende del número de objetivos. Este problema pone a prueba la capacidad de un MOEA para mantener la subpoblación en las diferentes partes de \mathcal{PF}_{true} .

$$f_{j=1:k-1}(\mathbf{x}) = x_j$$

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

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

4.1.2. Funciones de prueba DTLZ inversas

DTLZ1⁻¹

El problema *DTLZ1*⁻¹ es una versión rotada de *DTLZ1*, generando grandes espacios sin solución, siendo un frente interesante para el estudio de la distribución de los vectores de pesos y sus efectos en las soluciones.

$$f_1(\mathbf{x}) = -0.5(1 + g(\mathbf{y})) \prod_{i=1}^{k-1} x_i$$

$$f_{j=2:k-1}(\mathbf{x}) = -0.5(1 + g(\mathbf{y}))(1 - x_{k-j+1}) \prod_{i=1}^{k-1} x_i$$

$$f_k(\mathbf{x}) = -0.5(1 + g(\mathbf{y}))(1 - x_1)$$

$$g(\mathbf{y}) = 100 \left((n + \sum_{i=1}^n (y_i - 0.5)^2 - \cos(20\pi(y_i - 0.5))) \right)$$

DTLZ2⁻¹

Tiene una forma triangular inversa (rotada con respecto a *DTLZ2*) y una curvatura cóncava. Si bien, tiene una superficie amplia, presenta problemas para la diversidad en las fronteras del \mathcal{PF}_{true} donde los MOEAs tradicionales tienen una alta densidad de soluciones.

$$\begin{aligned}
f_1(\mathbf{x}) &= -0.5(1 + g(\mathbf{y})) \prod_{i=1}^{k-1} \cos x_i \frac{\pi}{2} \\
f_{j=2:k-1}(\mathbf{x}) &= -1(1 + g(\mathbf{y})) \left(\prod_{i=1}^{k-j} \cos x_i \frac{\pi}{2} \right) \sin x_{k-j+1} \frac{\pi}{2} \\
f_k(\mathbf{x}) &= -0.5(1 + g(\mathbf{y})) \sin x_1 \frac{\pi}{2} \\
g(\mathbf{y}) &= \sum_{i=1}^n (y_i - 0.5)^2
\end{aligned}$$

DTLZ7⁻¹

El espacio de búsqueda es disconexo. Este problema pone a prueba la capacidad de un MOEA para mantener la subpoblación en diferentes \mathcal{PF}_{true} . Los vectores de pesos requieren abarcar amplias áreas donde las pendientes provocan zonas densas en soluciones en los MOEAs clásicos.

$$\begin{aligned}
f_{j=1:k-1}(\mathbf{x}) &= -x_j \\
f_k(\mathbf{x}) &= -1(1 + g(\mathbf{y})) \left(k - \sum_{i=1}^{k-1} \left[\frac{f_i(1 + \sin 3\pi f_i)}{1 + g(\mathbf{y})} \right] \right) \\
g(\mathbf{y}) &= - \left(1 + \frac{9}{k} \sum_{i=1}^n y_i \right)
\end{aligned}$$

4.1.3. *Kite* o *Papalote*

El problema de optimización del Papalote [53] que tiene un \mathcal{PF}_{true} no convexo con una pendiente alta cerca de uno de sus extremos que se convierte en un desafío para la generación de vectores de pesos, puesto que tenderá a atraer una gran cantidad de soluciones sobre esta pendiente.

$$f_{j=1:k-1}(\mathbf{x}) = (1 + g(\mathbf{y}))x_j$$

$$f_k(\mathbf{x}) = (1 + g(\mathbf{y})) \left(\frac{1 - \prod_{i=1}^{k-1} \frac{9x_i+1}{10}}{1 - 0.1^{k-1}} \right)$$

$$g(\mathbf{y}) = \sum_{i=0}^n (y_i - 0.5)^2$$

Todos los algoritmos fueron implementados usando codificación en números reales y con parámetros idénticos enumerados en la Tabla 4.1. Estos parámetros son los que recomiendan los autores de los algoritmos después de amplias experimentaciones numéricas. Se usan estos valores para no sesgar los resultados obtenidos aquí, ajustando los mismos en cada algoritmo a cada función de prueba.

Parámetro	Valor
Probabilidad Mutación (PM)	$\frac{1}{n}$
Probabilidad de Cruza (SBX)	1.0
Número de Vecinos	20
Θ (PBI)	5.0
Generaciones	500, 1000, 2000
HV referencia kite, $DTLZ1, 2$	$\{1.18\}^k$
HV referencia $DTLZ7$	$\{11\}^k$
HV referencia $DTLZ1^{-1}, 2^{-1}, 7^{-1}$	$\{0\}^k$
Población inicial para 3 dimensiones	300
Población inicial para 5 dimensiones	500
Población inicial para 8 dimensiones	792

Cuadro 4.1: Diseño experimental

4.2. Resultados experimentales con métodos estáticos

En esta sección, se presenta la evaluación de los seis métodos estáticos para generar puntos de referencia, a saber: Diseño simplex lattice (SLD), Diseño simplex lattice de dos capas (TLD), Diseño uniforme (UD), Secuencia de Halton (HS), Secuencia de Hammersley (HMS)) y la secuencia de Sobol (SS). El rendimiento de estas técnicas se evalúa considerando la convergencia mediante el uso de Hipervolumen (HV) y la distribución uniforme a través de la Distancia generacional invertida (IGD) y el indicador de diversidad Δ .

4.2.1. NSGA-III

La Tabla 4.2 muestra los resultados del marco de trabajo *NSGA-III* con los seis conjuntos de puntos de referencia disponibles en 3, 5 y 8 dimensiones para las 7 funciones de prueba descritas en la sección 4.1. Cada una de estas pruebas fue repetida 50 veces la semilla del algoritmo de generación de números pseudoaleatorios entre corridas con parámetros idénticos (conjunto de puntos de referencia, dimensión y función de prueba) con el fin de asegurar la significancia estadística.

Los valores marcados en negrita representan los mejores valores encontrados para cada indicador utilizado. En el caso *NSGA-III*, el diseño *Símplex Lattice* (SLD), introducido por Scheffé en 1958 [46], muestra claras ventajas contra sus similares en prácticamente todas las funciones de prueba, el uso método de nichos centrados sobre puntos de referencia en el espacio de los objetivos introducido en *NSGA-III* se ve beneficiado por la densidad de vectores de pesos en las fronteras del *Símplex*.

En el caso de *DTLZ1*, SLD presenta los mejores resultados en términos de HV, IGD y Δ -diversidad para 3, 5 y 7 dimensiones. Mientras que las técnicas basadas en baja discrepancia no resaltan en ningún caso.

El diseño de dos capas sobresale en el indicador Δ -diversidad para los problemas *DTLZ2* y *DTLZ7*⁻¹ para 5 y 8 dimensiones. En particular, la alta densidad de vectores de pesos centrados en este diseño beneficia la búsqueda en frentes sin soluciones asociados a los vectores de pesos en las fronteras del *Símplex*.

Para *DTLZ1*⁻¹, SLD muestra los mejores valores para 3 y 5 dimensiones en IGD, sin embargo, tiene el peor resultado en este mismo indicador para 8 dimensiones. En cambio, TLD presenta consistentemente los mejores resultados en el indicador HV para 3, 5 y 8 dimensiones. Los vectores de pesos generados con la técnica de baja discrepancia UD, presentan los mejores resultados para 3 y 5 dimensiones en el indicador de Δ -diversidad.

Para *DTLZ2*⁻¹, todas las técnicas de generación de vectores de pesos tiene un pobre rendimiento en el indicador de convergencia HV, el mejor resultado se logra con SLD, pero aun por debajo de los resultados publicados por Ishibushi [28]. La técnica de baja discrepancia UD muestra excelentes resultados para esta función de prueba en términos de los indicadores IGD y Δ -diversidad.

Por último, para la función papalote (*kite*) las técnicas basadas en diseño de mezclas muestran consistentemente mejores resultados que sus contrapartes basadas en baja discrepancia.

4.2.2. MOEA/D Chebyshev (TCH)

La Tabla 4.3 muestra los resultados del marco de trabajo *MOEA/D con la función de escalarización TCH* con los seis conjuntos de puntos de referencia disponibles en 3, 5 y 8 dimensiones para las 7 funciones de prueba descritas en la Sección 4.1. Cada una de estas pruebas es repetida 50 veces variando la semilla del algoritmo de generación de números pseudoaleatorios entre corridas con parámetros idénticos (conjunto de puntos de referencia, dimensión y función de prueba) con el fin de asegurar la significancia estadística.

Los valores marcados en negrita representan en los mejores valores encontrados para cada indicador utilizado. En funciones con un frente continuo como *DTLZ1* y *DTLZ2*, SLD muestra resultados sólidos como la mejor técnica de generación de vectores de pesos en los indicadores HV, en el caso de IGD, es el tercer mejor resultado en 3 dimensiones para *DTLZ1* y la mejor técnica en 5 y 8 dimensiones. En Δ -diversidad TLD muestra excelentes resultados en 5 y 8 dimensiones para *DTLZ2*.

Para *DTLZ7*, las técnicas basadas en diseño de mezclas SLD y TLD, tienen mejores resultados en general para los indicadores HV e IGD. En términos de Δ -diversidad las técnicas basadas en baja discrepancia muestran mejores resultados conforme crece el número de dimensiones.

Considerando *DTLZ1*⁻¹, los métodos de generación de vectores de pesos basados en diseño de mezclas muestran peores resultados conforme aumenta el número de dimensiones, en particular se observa que la característica de mantener una gran cantidad de vectores de pesos en las fronteras del simplex reduce la efectividad en esta función de prueba puesto que no cubre gran parte de la superficie del \mathcal{PF}_{true} .

El caso particular de *DTLZ2*⁻¹ se ve dominado por los métodos de baja discrepancia cuando se usan 5 o más objetivos, aunque TLD funciona bien en algunos casos con 8 objetivos. Consistentemente UD, presenta los mejores resultados para el indicador IGD y Δ -diversidad.

En *DTLZ7*⁻¹, se observa una ventaja considerable en términos del indicador Δ -diversidad para el método TLD. Las técnicas de generación de vectores de pesos basadas en diseño de mezclas muestran un mejor rendimiento en este problema.

Por último, la función de prueba del papalote (*kite*) muestra resultados diversos donde no es posible determinar si alguna técnica de generación de vectores de pesos es mejor en algún sentido que sus similares independientemente de si es basada en diseño de mezclas o baja discrepancia. Demostrando que el problema del Papalote [53] a pesar de ser conexo es desafiante por las pendientes pronunciadas que presenta la geometría de su frente.

4.2.3. MOEA/D PBI

La Tabla 4.4 muestra los resultados del marco de trabajo *MOEA/D con la función de escalarización PBI* con los seis conjuntos de puntos de referencia disponibles en 3, 5 y 8 dimensiones para las 7 funciones de prueba descritas en la Sección 4.1. Cada una de estas pruebas es repetida 50 veces variando la semilla del algoritmo de generación de números pseudoaleatorios entre corridas con parámetros idénticos (conjunto de puntos de referencia, dimensión y función de prueba) con el fin de asegurar la significancia estadística.

Los valores marcados en negrita representan en los mejores valores encontrados para cada indicador utilizado. Como se esperaba para *DTLZ1*, tal y como se muestra en la Tabla 4.4, SLD presenta el mejor rendimiento tanto en convergencia como en diversidad para cualquier número de objetivos. La explicación de este resultado se debe a que el frente de Pareto de *DTLZ1* tiene la misma forma del simplex en el que se despliegan los puntos de referencia generados por SLD. Por otro lado, aunque TLD en la distribución de la segunda capa está pensada para tratar con formas de \mathcal{PF}_{true} irregulares y de altas dimensiones. Por esta razón, en *DTLZ1*, la segunda capa resulta algo contraproducente como se observa en términos de uniformidad (Δ -diversidad), donde exhibe un pobre rendimiento. Este hecho también se puede observar en la distribución ligeramente irregular de las soluciones en comparación con la producida por SLD. Sin embargo, dado que TLD cubre toda la superficie del \mathcal{PF}_{true} , mantiene una buena convergencia considerando el HV independientemente del número de objetivos. Los cuatro diseños basados en baja discrepancia (UD, HS, HMS y SS) presentan un rendimiento ligeramente peor en términos de HV e IGD. Aunque, en lo que respecta a la Δ -diversidad, UD no tiene el rendimiento esperado, no escala bien con el número de objetivos.

En cuanto a *DTLZ2*, las observaciones fueron similares a las de *DTLZ1*. Para 3 y 5 objetivos, SLD y TLD alcanzaron los mejores valores para los indicadores HV, IGD y Δ -diversidad. Sin embargo, para 8 objetivos, TLD obtuvo la mejor distribución de los puntos en términos de Δ -diversidad, mientras que SLD es sólo el cuarto mejor método. Una posible explicación de esto es que SLD no genera suficientes puntos en la región media de los frentes de Pareto de *DTLZ2*. Sin embargo, debido a la capa interna, TLD puede cubrir esos huecos dejados por SLD en dimensiones altas.

Por lo tanto, en MOPs con \mathcal{PF}_{true} lineales o simétricos convexos como en *DTLZ1* y *DTLZ2*, la mejor opción es utilizar métodos de generación de vectores de pesos basados en diseño mezclas como SLD o TLD para conseguir una mejor distribución.

Considerando *DTLZ7*, podemos comprobar que SLD tiene una buena convergencia con respecto a HV e IGD sólo para 3 objetivos. En cambio, para 8 objetivos, es el peor considerando IGD y el segundo peor en HV. En términos de Δ -diversidad, presenta uno de los peores valores para cualquier número de objetivos. Un comportamiento interesante es el que se observa en los

diseños basados en baja discrepancia. En general, mejoran su rendimiento cuando aumenta el número de objetivos, en particular HS y HMS. Por ejemplo, en lo que respecta al hipervolumen, HMS es la peor técnica para 3 objetivos, luego la cuarta mejor, y finalmente, una de las mejores técnicas cuando se utilizan 8 objetivos. En términos de diversidad, HS y UD son, en general, las mejores técnicas para cualquier número de objetivos.

Aunque el problema de prueba del papalote (*kite*) tiene un \mathcal{PF}_{true} convexo, podría suponer un reto para los métodos de generación de pesos porque su frente no convexo tiene una pendiente pronunciada cerca de uno de los extremos. De forma similar al rendimiento observado en *DTLZ7*, aquí SLD muestra el mejor rendimiento tanto en HV como en IGD para 3 objetivos, pero es el peor para 5 objetivos. En términos de diversidad, es el peor para cualquier caso. Por otro lado, los métodos de baja discrepancia (HS, HMS y SS) muestran un gran rendimiento de convergencia (HV e IGD) para 5 y 8 objetivos. Aunque UD es un método basado en la discrepancia, obtiene un pobre rendimiento para 5 objetivos en cuanto a HV y Δ -diversidad, y es el peor método en cuanto a IGD. En conclusión, para problemas no convexos con formas \mathcal{PF}_{true} difíciles, como los de *DTLZ7* y el problema de prueba del papalote, las técnicas de diseño de mezclas sólo son útiles con un número bajo de objetivos. Para 5 y 8 objetivos, los métodos basados en secuencias de baja discrepancia proporcionan una mejor distribución, especialmente HS y HMS.

Por último, con respecto a los problemas invertidos, $DTLZ1^{-1}$, $DTLZ2^{-1}$ y $DTLZ7^{-1}$, se obtuvieron los siguientes resultados. $DTLZ1^{-1}$ es difícil para SLD y TLD porque muchos de sus puntos de referencia no se cruzan con el \mathcal{PF}_{true} . En general, analizando los valores de los indicadores de la Tabla 4.4, los métodos de baja discrepancia mejoran sus prestaciones tanto en términos de convergencia como de diversidad a medida que crece el número de objetivos. En dicha tabla, se puede observar que, para 8 objetivos, considerando Δ -diversidad e IGD, HMS, HS y SS son las mejores técnicas. De nuevo, SLD y TLD sólo funcionan bien para 3 objetivos. Para 8 objetivos, son los peores métodos en cuanto a los tres indicadores. UD presenta un rendimiento regular para cualquier número de objetivos.

Aunque en $DTLZ2^{-1}$, las técnicas de baja discrepancia mejoran su rendimiento como en $DTLZ1^{-1}$, la diferencia importante es que en términos de convergencia utilizando HV, tanto SLD como TLD mantienen un buen rendimiento para 5 y 8 objetivos. En cambio, en cuanto a la Δ -diversidad, SLD y TLD presentan los peores valores para 8 objetivos.

En $DTLZ7^{-1}$, los métodos basados en secuencias de baja discrepancia mejoran el rendimiento, pero no de forma tan notable como en $DTLZ1^{-1}$. En términos de HV, estos métodos superan a SLD y TLD hasta los 8 objetivos. En cuanto a la diversidad con respecto a IGD, TLD tiene un buen rendimiento para cualquier número de objetivos. Como en otros problemas, en general, UD no consigue un buen rendimiento en cuanto a convergencia y diversidad. Sin embargo, para algunos objetivos obtiene buenos valores de Δ -diversidad. Esto significa que, al menos, consigue cubrir los puntos extremos del frente de Pareto.

En general, en lo que respecta a los MOP con un \mathcal{PF}_{true} invertido, los métodos de baja discrepancia son útiles para 5 o más objetivos, aunque TLD funciona bien en algunos casos con 8 objetivos.

4.3. Resumen

En esta primera comparación de las técnicas de generación estática, se obtuvieron las siguientes observaciones generales. Para problemas con frentes de Pareto convexos lineales o simétricos, técnicas basadas en diseño de mezclas (Simplex Lattice y Simplex lattice de dos capas) obtuvieron los mejores resultados tanto en convergencia como en distribución para cualquier número de objetivos. Con respecto a las funciones con frente de Pareto no convexos o disconexos, los métodos de diseño basados en baja discrepancia (Diseño uniforme, Halton, Hammersley y Sobol) proporcionaron un buen rendimiento para los objetivos 5 y 8. Finalmente, para los MOP con frente de Pareto invertido, las técnicas de baja discrepancia mejoran su rendimiento a medida que aumenta el número de objetivos.

Métodos para ajuste dinámico de pesos

5.1. Motivación

La variedad de técnicas de generación, como las abordadas en los capítulos anteriores, ofrece una amplia gama de alternativas para inicializar los vectores de pesos utilizados dentro de un MOEA basado en un conjunto de referencia. Sin embargo, la premisa para que estos generadores de vectores de peso funcionen bien es que el frente de Pareto del problema tenga una forma regular, es decir, similar a un simplex. Como se ha observado anteriormente, una forma irregular del frente de Pareto, como un frente simplex-invertido, disconexo, degenerado o mal escalado, puede implicar problemas críticos de diversidad: muchos vectores de peso corresponden a la misma solución única en el conjunto de soluciones final o, a la inversa, algunas regiones del frente de Pareto sólo se describen escasamente.

Una solución intuitiva a este problema es la actualización adaptativa de los vectores de peso durante el proceso de optimización. Se han realizado varios intentos interesantes en esta línea. En los primeros estudios en esta área, los investigadores consideraron vectores de peso generados aleatoriamente, por ejemplo, Ishibushi y Murata [27] intentan producir nuevas direcciones de búsqueda aleatorias en cada generación. Más recientemente, Li et al. [32] introducen una población externa, que se utiliza para almacenar soluciones prometedoras obtenidas de la relación ϵ -dominancia, para ayudar a generar los pesos aleatorios. Los estudios anteriores muestran que este tipo de estrategias pueden ser útiles para tratar con frentes de Pareto disconexos.

Por otro lado, una tendencia popular dentro de la adaptación de pesos consiste en añadir/eliminar vectores de pesos en regiones dispersas/densas del espacio objetivo para promover la diversificación. Por ejemplo, en [33], cada peso se ajusta periódicamente para aumentar la distancia de su solución correspondiente con respecto a las demás soluciones. En [30], se propone un método de ajuste de pesos adaptativo, que muestrea la curva de regresión de los vectores de

pesos con base en un archivo externo. Posteriormente, proponen un método de adaptación de pesos que entrena un mapa auto-organizado en función de la población actual [20].

Además, en [41] se propone una estrategia de adaptación de pesos (MOEA/D-AWA) para frentes de Pareto discontinuos, inspirada en MOEA/D-DRA [59] donde calcula una función de utilidad por cada subproblema, de esta manera selecciona qué subproblemas deben ser priorizados distribuyendo el esfuerzo computacional de acuerdo a la utilidad del subproblema. MOEA/D-AWA tiene dos fases: primero, se utiliza un conjunto de vectores de pesos preestablecidos hasta que se considera que la población está lo suficientemente cerca del frente de Pareto. Después, los vectores de pesos se ajustan periódicamente eliminando los vectores de pesos apuntando a regiones pobladas del espacio objetivo y creando nuevos vectores de pesos en las regiones con baja densidad de soluciones. En el mismo estudio, los autores también proponen una proyección simplex como método para generar nuevos vectores de peso sobre el simplex unitario Tm , y muestran la teoría detrás de la agrupación de soluciones en los bordes del frente de Pareto.

Más recientemente, Zhang et al. propusieron una adaptación del vector de pesos diseñada a través de una interpolación lineal para problemas de optimización bi-objetivo con frentes de Pareto discontinuos [55]. Por último, otras investigaciones ajustan de forma adaptativa las direcciones de búsqueda en función de la distribución de la población evolutiva en MOEAs no basados en la descomposición, lo que también puede ser visto como una adaptación de los vectores de peso [49].

5.2. Repulsión de subpoblaciones

En 2017, Ahrari et al.[1] introducen una técnica de nichos basada en la repulsión de subpoblaciones, pensada como una técnica de optimización multimodal usando la exploración intensiva del espacio, determinando soluciones élite y con base en ellas establece subregiones nuevas de búsqueda. Esta técnica incluye selección de soluciones élite, cambio dinámico de tamaño de los nichos, reinicios guiados y reducción de cálculo evitando regiones ya exploradas previamente.

En el trabajo de Ahrari et al.[1], se utiliza CMA-ES (Covariance Matrix Adaptation Evolutionary Strategies)[23] como motor de búsqueda. La población global se divide en varias subpoblaciones. Toda subpoblación P_i tiene un tamaño fijo λ , sus propios parámetros de cruce y mutación (desviación estándar σ_{mean_i} y matriz de covarianza C_i), un centroide (x_{mean_i}) y miembros élite. Existe también un conjunto de puntos prohibidos (tabú) y_k (constituidos por las mejores soluciones actuales o los centroides de las soluciones actuales o los centros de las subpoblaciones) que restringen la generación de nuevas soluciones en su entorno, con base a

una métrica [1]. De esta forma, las nuevas soluciones aceptables se encontrarán en regiones poco exploradas del espacio de búsqueda. Una solución se considera aceptable si satisface el criterio de estar al menos a una distancia D de todos los puntos prohibidos, de lo contrario, la solución es rechazada.

El efecto general de esta estrategia es una remodelación de la distribución espacial de las soluciones alcanzadas de tal manera que las subpoblaciones exploran en regiones del espacio aún no visitadas previamente, prohibiendo que dos subpoblaciones busquen en la misma región del espacio de búsqueda.

Esta estrategia basada en nichos utiliza como métrica una variación de la distancia de Mahalanobis [36], llamada distancia tabú normalizada, una versión escalada de la distancia euclidiana, que permite comprobar que una solución está lo suficientemente alejada de los puntos prohibidos. La distancia tabú normalizada (D_{ij-k}) de la j -ésima solución (x_{ij}) de P_i , dada la matriz de covarianza C_i y la desviación estándar σ_{mean_i} la distancia al punto prohibido y_k , se define como sigue:

$$D_{ij-k} = \frac{(x_{ij} - y_k)^T C_i^{-1} (x_{ij} - y_k)}{\sigma_{mean_i}}. \quad (5.1)$$

Una consecuencia del uso de esta distancia en particular es que D_{ij-k} es inversamente proporcional a σ_{mean_i} , es decir, cuando los elementos de la subpoblación i convergen hacia algún óptimo local (su variabilidad disminuye), la distancia D_{ij-k} aumenta. Este comportamiento garantiza que las nuevas soluciones sean significativamente diferentes de y_k .

En términos de manejo de la población y para evitar un exceso de puntos tabú, la estrategia de repulsión de subpoblaciones implementa al menos dos mecanismos de control:

- Una probabilidad de aceptación para evitar largos procesos de selección con soluciones que no cumplen con ser aceptables con el criterio de distancia mínima.

Esta probabilidad se calcula de acuerdo a la matriz de covarianza de tal manera que se reduce conforme avanzan los procesos evolutivos agotando la posibilidad de encontrar nuevas regiones de búsqueda.

- Un mecanismo de control del número de puntos tabú, que eventualmente puede eliminar el estatus tabú de un punto de acuerdo a su relevancia. Esta relevancia se determina en términos de la función objetivo y de la edad de la subpoblación (si fue asignada recientemente o no).

Los mecanismos de repulsión de subpoblaciones basadas en puntos tabú para promover la búsqueda de soluciones diversas, así como la introducción del uso de distancias normadas para

poder trabajar en espacios de grandes dimensiones, representan las principales aportaciones de [1] en optimización multi-modal. En el presente trabajo, se pretende utilizar esta metodología y adaptarla al contexto multi-objetivo con el ajuste de puntos de referencia.

5.2.1. Estrategia propuesta para la actualización de vectores de pesos

La estrategia de adaptación dinámica y parte de sus resultados fueron publicados en “Generation techniques and a novel on-line adaptation strategy for weight vectors within decomposition-based MOEAs” [45] presentado en Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO 2019).

La estrategia propuesta en este trabajo pretende aprender del proceso de búsqueda para modificar algunos vectores de pesos que resultan inútiles. Así, después de una fase de trabajo normal del *MOEA/D* canónico, es decir, cuando se considera que se alcanza cierta convergencia, se selecciona cada punto no dominado como solución tabú. Su correspondiente vector de pesos se reajusta en función de la posición de la solución. Esto deja un número igual de vectores de peso y soluciones que no participan en esta asignación. Los vectores sin ninguna solución cercana (en términos de distancia de Malahanobis) se recalculan mediante una combinación de los vectores de peso tabú.

La estructura general del *MOEA/D* con una técnica de adaptación de pesos se muestra en el Algoritmo 5.2.1. Las primeras líneas son el algoritmo *MOEA/D* clásico. En la línea 10, se actualiza el archivo externo. Luego, en la línea 11, se realiza una prueba sobre la frecuencia de actualización de pesos (WUF) (cada 5% del número total de generaciones). En caso de adaptación de pesos, se identifican las soluciones no dominadas (línea 12) y se utilizan para clasificar a los vectores tabú y marcarlos como reubicables (línea 13). En la línea 14, los vectores tabú se actualizan según sus correspondientes soluciones no dominadas y en la línea 15, se generan nuevos valores para los vectores reubicables. En la línea 16 es necesaria una última operación de actualización de vecindario.

Cuando se optimiza un MOP, las soluciones actuales no dominadas, con el progreso de la evolución, se aproximan gradualmente al frente de Pareto, siendo así más probable que reflejen la forma del frente de Pareto.

Un conjunto de vectores de peso probablemente evoluciona lentamente en comparación con la población evolutiva que es conducida por la función escalar en los algoritmos basados en la descomposición, el conjunto puede ser capaz de proporcionar nuevas direcciones de búsqueda que son inexploradas por la población conducida por la función escalar.

También, como introdujeron Zhang y Qui [55, 41], las soluciones en los algoritmos basa-

dos en la descomposición tienen la tendencia a formar grupos alrededor de las fronteras del verdadero frente de Pareto. Aprovechando las observaciones anteriores es posible proponer una nueva estrategia de adaptación en línea para los vectores de peso dentro de los MOEAs basados en la descomposición. La estructura general se muestra en el Algoritmo 5.2.2.

Algoritmo 5.2.1: MOEA/D con adaptación de vectores de peso dirigida por dominancia

Data: $IterMax$: condición de paro, N : subproblemas, T : tamaño de vecindario, WUF : frecuencia de adaptación de vectores de peso, P_{cross} : cruza, P_{mut} : mutación, $meth$: método de reubicación de vectores de peso

Result: PE : Frente de Pareto aproximado, W : Vectores de peso

```

1 Population  $\leftarrow$  InitialPopulation( $Populationsize$ ,  $NumObjectives$ );
2  $W = w_1, \dots, w_N \leftarrow$  UniformlyDistributedWeights( $N$ );
3  $B_0, \dots, B_N \leftarrow$  NearestNeighbours( $T$ ,  $W$ );
4 while  $\neg Stop(IterMax)$  do
5   for  $i := 1$  to  $N$  do
6     Selection  $\leftarrow$  ParentSelection( $B_i$ );
7      $C_i \leftarrow$  CrossAndMutation(Selection,  $P_{cross}$ ,  $P_{mut}$ );
8     if  $F_{pbi}(C_i, z^*) < x F_{pbi}(B_i, z^*)$  then
9        $x \leftarrow C_i$ ;
10   $PE \leftarrow$  UpdatePE( $Population$ );
11  if test( $WUF$ ) then
12     $NonDom \leftarrow$  NonDominatedSolutions( $PE$ );
13     $TabuP, RelocP \leftarrow$  TabuSel( $NonDom, population, T$ );
14     $Wtmp \leftarrow$  WUpdate( $TabuP, N$ );
15     $W \leftarrow$  WRelocation( $RelocP, Wtmp, N, meth$ );
16     $B_0, \dots, B_N \leftarrow$  NearestNeighbours( $T$ ,  $W$ );
17 return ( $PE, W$ )

```

5.2.2. Selección Tabú

En la estrategia propuesta pretendemos añadir vectores de peso (en la población evolutiva) cuyas direcciones/áreas de búsqueda son prometedoras pero poco desarrolladas. Para ello, utilizamos una selección o clasificación de las soluciones actuales no dominadas en la población evolutiva dentro de puntos reubicables y puntos Tabú, un concepto inspirado en la repulsión de subpoblaciones propuesta por Ahrari[1]. Cada punto Tabú tiene la mínima distancia perpendicular entre las soluciones de la vecindad y el vector de pesos actual. Después de marcar una solución como punto Tabu, todos sus vecinos a una distancia de Mahalanobis de 0.5 se clasifican como puntos reubicables, como se explica en el trabajo de Ahrari. Con

esta estrategia, queremos repeler todas las soluciones similares en la vecindad y reubicar esos vectores de peso en otras zonas. El marco general se explica en el Algoritmo 5.2.3.

Algoritmo 5.2.2: Selección Tabú

Data: *NonDom*: Soluciones no dominadas, *pop*: Población, *T*: Tamaño de vecindario

Result: *TabuP*: Puntos Tabu, *RelocableP*: Puntos reubicables

```

1 TabuP ← ∅ ;
2 RelocableP ← ∅ ;
3 for i := 1 to sizeof(NonDom) do
4   if MinPerpendicularDistance(NonDomi, pop, T) then
5     TabuP ← TabuP ∪ NonDomi;
6     for j := 1 to T do
7       if MahalanobisDist(NonDomi, NomDomNeighbour(j)) ≤ 0.5 then
8         RelocableP ← RelocableP ∪ NomDomNeighbour(j);
9 return (TabuP, RelocableP)

```

5.2.3. Reubicación y actualización del peso

El proceso de actualización es la proyección sobre el simplex unitario T_m de los puntos Tabú en la población evolutiva, este paso tiene la consecuencia de delimitar el espacio de búsqueda utilizando la información de los puntos no dominados reduciendo el desperdicio de esfuerzo computacional en vectores de peso con solución alejados de la frontera del frente de Pareto.

La reubicación de los pesos se realiza mediante uno de los tres métodos siguientes: 1) interpolación lineal entre dos pesos de puntos Tabú; 2) un punto aleatorio entre los pesos de puntos Tabú utilizando una distribución normal multivariada; 3) una combinación de los dos métodos anteriores utilizando el punto generado por la interpolación lineal como media de la distribución normal multivariada. Para todas las soluciones etiquetadas como reubicables se sustituye su vector de pesos real por uno generado por los métodos mencionados. Este proceso se muestra en el Algoritmo 5.2.3.

5.2.4. Metodología experimental

En esta sección, se prueba la técnica de adaptación de pesos utilizando únicamente el diseño simplex lattice como generador inicial de vectores de pesos (dentro de MOEA/D-PBI). Por lo tanto, las comparaciones sólo incluyen el SLD (SLD-MOEA/D) y las tres versiones del

Algoritmo 5.2.3: Reubicación de vectores de peso

Data: *RelocableP*: puntos reubicables, *Wtemp*: pesos temporales, *meth*: método de reubicación de vectores de peso

Result: *W*: nuevos pesos

```

1 for  $i := 1$  to  $\text{sizeof}(\text{RelocableP})$  do
2    $w_1, w_2 \leftarrow \text{getTwoRandomReferences}(Wtemp)$ ;
3   if  $meth == \text{linear}$  then
4      $W \leftarrow Wtemp \cup \text{randomPointBetweenWeights}(w_1, w_2)$ ;
5   else if  $meth == \text{normal}$  then
6      $middleP \leftarrow \text{middlePoint}(w_1, w_2)$ ;
7      $stdDev \leftarrow \text{StdDev}(w_1, w_2)$ ;
8      $tmp \leftarrow \text{MultiVariateNormalDist}(middleP, StdDev)$ ;
9      $W \leftarrow W \cup \text{SimplexProjection}(tmp)$ ;
10  else if  $meth == \text{hibrid}$  then
11     $centerP \leftarrow \text{randomPointBetweenWeights}(w_1, w_2)$ ;
12     $stdDev \leftarrow \text{StdDev}(w_1, w_2)$ ;
13     $tmp \leftarrow \text{MultiVariateNormalDist}(centerP, StdDev)$ ;
14     $W \leftarrow W \cup \text{SimplexProjection}(tmp)$ ;
15 return ( $wtemp$ )

```

método de adaptación de pesos basado en la dominancia (Lin, Nor, Hyb-MOEA/D). Además, el estudio sólo se centra en las funciones de prueba DTLZ1, DTLZ2, DTLZ7, $DTLZ1^{-1}$, $DTLZ1^{-2}$, $DTLZ1^{-7}$ y Papalote (Kite), que presentan características interesantes en sus frentes de Pareto reales. Se realizaron 50 ejecuciones para cada técnica en DTLZ1, DTLZ2, DTLZ7, $DTLZ1^{-1}$, $DTLZ1^{-2}$, $DTLZ1^{-7}$ y Papalote (Kite) para 3, 5 y 8 dimensiones. Los mismos tres indicadores de rendimiento (HV, IGD y Δ -Diversidad, mediana y desviación estándar) se presentan en la Tabla 5.2 (también se realizaron pruebas basadas en rangos de Wilcoxon para todas las comparaciones por pares de indicadores).

Todos los algoritmos fueron implementados usando codificación en números reales y con parámetros idénticos enumerados en la Tabla 4.1. La selección de los mismos se basa en la literatura específica sobre los algoritmos que dan origen a este trabajo facilitando posteriores comparaciones.

5.2.5. Resultados numéricos y discusión

Los resultados están presentados en la Tabla 5.2. En la función de prueba DTLZ1. Se muestra que el diseño simplex lattice domina en términos de hipervolumen y Δ -diversidad

Parámetro	Valor
Probabilidad Mutación (PM)	$\frac{1}{n}$
Probabilidad de Cruza (SBX)	1.0
Número de Vecinos	20
Θ (PBI)	5.0
Generaciones	1000, 2000, 4000
HV referencia kite, $DTLZ_{1,2}$	$\{1.18\}^k$
HV referencia $DTLZ_7$	$\{11\}^k$
HV referencia $DTLZ_{1^{-1}, 2^{-1}, 7^{-1}}$	$\{0\}^k$
Población inicial para 3 dimensiones	300
Población inicial para 5 dimensiones	500
Población inicial para 8 dimensiones	792

Cuadro 5.1: Diseño experimental.

para 5 y 8 dimensiones. Las técnicas adaptativas difieren poco en sus resultados con simplex lattice, es decir realizan muy pocas reubicaciones de puntos, los resultados son en la práctica iguales dada la poca variación (desviación estándar) en dicha función.

Para $DTLZ_2$. El Algoritmo presenta excelentes resultados en las técnicas de adaptación de vectores de pesos, en particular el uso de la distribución normal multivariada, donde se reduce drásticamente la métrica IGD.

$DTLZ_7$ es de particular interés debido a sus varios frentes óptimos disconexos, de manera que muchos puntos de la superficie simplex están orientados hacia "regiones en blanco". Como se ilustra en la Figura 5.1 (donde el conjunto de referencia aparece en rojo y el frente aproximado en azul), la técnica de adaptación HybMOEA/D transforma la distribución de pesos, que finalmente coincide con la ubicación aproximada de los cuatro frentes disconexos. Por lo tanto, se consigue una mejor descripción de los cuatro componentes. SLD-MOEA/D encuentra muchas soluciones en los límites, mientras que sólo unos pocos puntos se distribuyen (de forma bastante uniforme) en los componentes restantes. Estas tendencias se ven confirmadas por los indicadores de rendimiento: HV son similares para todas las técnicas (aunque las pruebas basadas en el rango de Wilcoxon siempre concluyen en una diferencia significativa), excepto para 8 dimensiones donde las técnicas dinámicas superan a SLD. También son sistemáticamente mejores que SLD para IGD y se obtienen resultados compartidos para la Δ -diversidad.

Las funciones Papalote y $DTLZ_{1^{-1}}$ comparten algunas características: su forma del frente es compleja para los MOEAs basados en puntos de referencia, ya que muchos puntos simplex están orientados hacia regiones sin soluciones. Como consecuencia, muchas soluciones son encontradas por SLD-MOEA/D en los límites de los frentes mientras que las regiones interiores

Test	Ind.	dim	SLD-MOEA/D	Lin-MOEA/D	Nor-MOEA/D	Hyb-MOEA/D	
DTLZ1	HV	3	1.45277 (0.00070)	1.6193 (5.7001e-05)	1.6193 (6.5901e-05)	1.6193 (5.6703e-05)	
		5	2.28700 (0.00000)	2.287 (4.4949e-07)	2.287 (4.4949e-07)	2.287 (4.4949e-07)	
		8	3.75885 (0.00000)	3.7589 (1.2337e-07)	3.7589 (1.2278e-07)	3.7589 (1.2337e-07)	
	IGD	3	0.29045 (0.00080)	0.0090524 (0.00013753)	0.0090524 (0.0001376)	0.0090523 (0.00013764)	
		5	0.03773 (0.00001)	0.037705 (1.0541e-05)	0.037709 (1.0674e-05)	0.037704 (1.0574e-05)	
		8	0.05901 (0.00001)	0.058985 (2.5211e-05)	0.058995 (2.5211e-05)	0.058985 (2.5211e-05)	
	Δ	3	0.09742 (0.00024)	0.059351 (0.0018348)	0.061075 (0.013873)	0.059354 (0.0018546)	
		5	0.00934 (0.00000)	0.009364 (1.7869e-05)	0.0093639 (1.7408e-05)	0.009364 (1.7869e-05)	
		8	0.00652 (0.00009)	0.0068103 (0.00014647)	0.0068103 (0.00014647)	0.0068103 (0.00014647)	
	DTLZ2	HV	3	1.08595 (0.00001)	1.6193 (5.7001e-05)	1.6193 (6.5901e-05)	1.6193 (5.6703e-05)
			5	2.02144 (0.00040)	2.287 (4.851e-07)	2.287 (4.6359e-07)	2.287 (4.4949e-07)
			8	3.66827 (0.00002)	3.7589 (1.2337e-07)	3.7589 (1.2278e-07)	3.7589 (1.2337e-07)
IGD		3	0.01863 (0.00000)	0.0090524 (0.00013753)	0.0090524 (0.0001376)	0.0090523 (0.00013764)	
		5	0.11687 (0.00000)	0.037705 (1.0274e-05)	0.037704 (1.0574e-05)	0.037709 (1.0573e-05)	
		8	0.46880 (0.00000)	0.058985 (2.5221e-05)	0.058985 (2.5211e-05)	0.058985 (2.5213e-05)	
Δ		3	0.21380 (0.00012)	0.059351 (0.0018348)	0.061075 (0.013873)	0.059354 (0.0018546)	
		5	0.16372 (0.00001)	0.009364 (1.7869e-05)	0.0093639 (1.7408e-05)	0.009364 (1.7869e-05)	
		8	0.15656 (0.00001)	0.0068107 (0.00014647)	0.0068104 (0.00016662)	0.0068103 (0.00014647)	
DTLZ7		HV	3	990.94528 (55.68816)	952.71 (90.868)	954.68 (91.602)	957.16 (92.475)
			5	106974.92256 (4.85002)	1.0615e+05 (597.9)	1.0732e+05 (278.76)	1.0782e+05 (122.77)
			8	103508612.39430 (2468038.12139)	8.4132e+07 (5.091e+06)	8.5452e+07 (5.0593e+06)	8.6741e+07 (5.4619e+06)
	IGD	3	0.07394 (0.05019)	0.096546 (0.086435)	0.094037 (0.087249)	0.08844 (0.0893)	
		5	0.44697 (0.00014)	0.34368 (0.016687)	0.35503 (0.01459)	0.33042 (0.019036)	
		8	5.50966 (0.76683)	5.4261 (0.60367)	5.2804 (0.62199)	5.158 (0.68225)	
	Δ	3	1.16310 (0.02635)	0.73404 (0.15005)	0.71698 (0.15382)	0.74246 (0.15196)	
		5	1.13303 (0.00126)	0.79355 (0.030979)	0.78332 (0.028615)	0.73866 (0.029768)	
		8	0.83376 (0.01582)	0.88308 (0.033268)	0.86755 (0.014537)	0.87251 (0.012343)	
	DTLZ1 ⁻¹	HV	3	8.748E+07 (5045.70)	9.0485e+07 (1.1951e+06)	9.0242e+07 (1.1356e+06)	8.9832e+07 (1.0804e+06)
			5	2.798E+05 (9.857E+04)	2.2245e+11 (9.6254e+09)	2.3931e+11 (1.1688e+10)	2.1398e+11 (1.0467e+10)
			8	1.384E+10 (1.109E+12)	4.1375e+13 (1.9934e+13)	3.6396e+13 (1.8353e+13)	2.5167e+13 (1.327e+13)
IGD		3	434.78700 (0.00372)	453.22 (6.6321)	452.57 (7.684)	444.99 (5.7168)	
		5	266.63600 (0.00100)	420.24 (27.059)	419.11 (26.06)	417.08 (23.371)	
		8	443.06050 (52.62104)	549.26 (19.488)	543.02 (22.084)	537.27 (19.152)	
Δ		3	1.02225 (0.00044)	0.57312 (0.029628)	0.57278 (0.032554)	0.58273 (0.030165)	
		5	1.53902 (0.00128)	0.57664 (0.028421)	0.44584 (0.023578)	0.47488 (0.028099)	
		8	1.83157 (0.02632)	0.61931 (0.040976)	0.40582 (0.049126)	0.4666 (0.046509)	
DTLZ2 ⁻¹		HV	3	345.68182 (0.01621)	13.007 (0.03197)	13.001 (0.036894)	12.991 (0.04125)
			5	13.27768 (0.03843)	6.019 (0.13507)	6.1048 (0.13796)	5.8868 (0.13564)
			8	5.83867 (0.03770)	0.10441 (0.0043088)	0.10919 (0.0041929)	0.10529 (0.0048178)
	IGD	3	4.25000 (0.00000)	1.2747 (0.0032522)	1.2745 (0.0037692)	1.2744 (0.0033636)	
		5	0.97998 (0.0018638)	1.3799 (0.0038957)	1.3796 (0.0029466)	1.3795 (0.0033543)	
		8	1.61965 (0.02195)	1.7698 (0.01167)	1.7693 (0.011566)	1.7662 (0.015265)	
	Δ	3	0.83627 (0.00082)	0.63473 (0.025824)	0.6254 (0.025715)	0.60459 (0.026649)	
		5	0.33956 (0.00071)	0.3511 (0.012767)	0.33606 (0.007956)	0.34362 (0.0097706)	
		8	0.29725 (0.00070)	0.29509 (0.0078941)	0.27297 (0.0061272)	0.27862 (0.0077423)	
	DTLZ7 ⁻¹	HV	3	32.25700 (0.00001)	32.237 (0.006814)	32.238 (0.0064739)	32.21 (0.23183)
			5	53.00283 (0.02129)	53.245 (0.018405)	53.254 (0.013004)	53.271 (0.013165)
			8	2.26150 (0.00149)	2.2707 (0.010357)	2.2769 (0.01647)	2.2908 (0.024697)
IGD		3	0.00813 (0.00000)	0.0053097 (0.00011357)	0.0053002 (0.00019806)	0.0054844 (0.0010869)	
		5	0.32892 (0.00079)	0.27475 (0.0071707)	0.26542 (0.0060731)	0.27164 (0.0067699)	
		8	3.44283 (0.00079)	3.4426 (0.0008096)	3.4279 (0.049665)	3.3959 (0.066686)	
Δ		3	1.79847 (0.00077)	1.5066 (0.033582)	1.4961 (0.036856)	1.5046 (0.033394)	
		5	1.35196 (0.00321)	1.1831 (0.018161)	1.1249 (0.015322)	1.14 (0.012571)	
		8	3.44283 (0.00079)	1.5232 (0.0030224)	1.517 (0.0028961)	1.5195 (0.0033124)	
kite		HV	3	0.72878 (0.00000)	0.71862 (0.0026356)	0.71915 (0.0034063)	0.72107 (0.0026839)
			5	0.42707 (0.02603)	0.47232 (0.019068)	0.46775 (0.02302)	0.46954 (0.019034)
			8	0.57339 (0.00001)	0.57342 (5.5668e-05)	0.57343 (6.6738e-05)	0.57343 (6.4314e-05)
	IGD	3	0.03519 (0.00000)	0.035247 (0.0013434)	0.035562 (0.0015011)	0.036287 (0.0013713)	
		5	0.22498 (0.05153)	0.17262 (0.021334)	0.17736 (0.027115)	0.1763 (0.02058)	
		8	1.50465 (0.08738)	1.1504 (0.15494)	1.1428 (0.15638)	1.1423 (0.15711)	
	Δ	3	0.82997 (0.00007)	0.6484 (0.033291)	0.63431 (0.028898)	0.65406 (0.03306)	
		5	0.94935 (0.03379)	0.47338 (0.051389)	0.40241 (0.081205)	0.42791 (0.066007)	
		8	1.50465 (0.08738)	0.64014 (0.055661)	0.55573 (0.064845)	0.59098 (0.06139)	

Cuadro 5.2: Resultados con Técnicas dinámicas de adaptación de pesos.

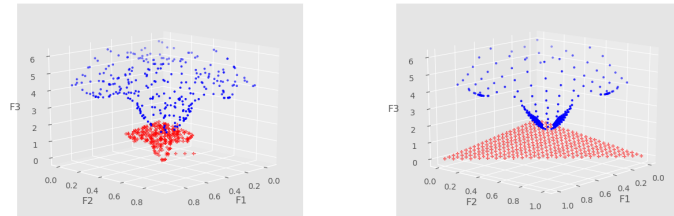


Figura 5.1: Hyb-MOEA/D vs. SLD-MOEA/D para DTLZ7.

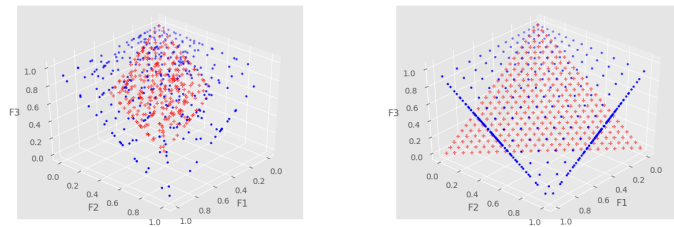
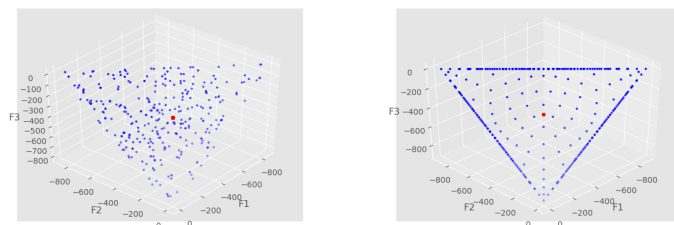


Figura 5.2: Nor-MOEA/D vs. SLD-MOEA/D para función papalote.

Figura 5.3: Lin-MOEA/D vs. SLD-MOEA/D para DTLZ⁻¹.

están casi desiertas, como se muestra en las Figuras 5.2 y 5.3. Por otro lado, los vectores de peso de las técnicas dinámicas se adaptan a la forma del frente de Pareto, como se ilustra claramente en la Figura 5.2 (la escala de la Figura 5.3 no permite observar la misma tendencia para $DTLZ1^{-1}$). Esta adaptación del peso genera una distribución densa, aunque no uniforme, de soluciones dentro del frente. En efecto, para la función Papalote, los indicadores de rendimiento muestran que todas las técnicas dinámicas obtienen hipervolumenes similares o mejores, y siempre los mejores valores de IGD y Δ -diversidad, para todas las dimensiones.

Para $DTLZ1^{-1}$, el indicador HV es mucho mejor para los métodos adaptativos, gracias a los numerosos puntos dentro del frente que tienen una contribución significativa, especialmente con un amplio rango de variación de objetivos. A la inversa, las soluciones concentradas en los límites del frente encontradas con SLD-MOEA/D casi no participan en el HV. El indicador de Δ -diversidad también es globalmente mejor para las técnicas adaptativas. Sin embargo, la distribución uniforme de las soluciones encontradas por SLD-MOEA/D dentro del frente compensa su escaso número, mientras que las técnicas dinámicas no cubren uniformemente la superficie del frente y obtienen valores pobres de IGD.

En la función de prueba $DTLZ2^{-1}$ se muestra que el diseño *simplex lattice* domina en términos de HV para todas las dimensiones. Las técnicas adaptativas mejoran las métricas relativas a la diversidad de las soluciones, destacando las versiones Nor-MOEA/D e Hyb-MOEA/D que presentan resultados prometedores en todas las dimensiones sin lograr una convergencia plena.

Para $DTLZ7^{-1}$ las técnicas adaptativas destacan en sus resultados, en particular la versiones Nor-MOEA/D e Hyb-MOEA/D que presentan resultados prometedores en 5 y 8 dimensiones para las tres métricas utilizadas.

5.3. Resumen

Los experimentos realizados con la estrategia de adaptación de pesos propuesta proporcionaron resultados prometedores. Como se observó anteriormente, una forma de frente de Pareto irregular, como un frente *simplex* invertido, disconexo, degenerado o mal escalado, puede implicar problemas críticos de diversidad: muchos vectores de peso corresponden a la misma solución única en el conjunto de soluciones final o, a la inversa, algunas regiones del frente de Pareto se describen escasamente.

Los experimentos realizados con la estrategia de adaptación de pesos propuesta proporcionaron resultados prometedores. En efecto, la capacidad de esta nueva técnica para modificar la distribución de pesos en función de la superficie expone resultados notables en 3, 5 y 8 dimensiones. La estrategia propuesta merece, pues, una experimentación exhaustiva, con diferentes

funciones de prueba y formas de frentes, por ejemplo, frentes degenerados. Por último, las primeras tendencias observadas en este trabajo permiten identificar algunos puntos débiles, como la pérdida de puntos extremos o límites del frente de Pareto, que deberían corregirse en trabajos futuros.

Conclusiones y trabajo futuro

6.1. Conclusiones

Esta investigación tuvo como objetivo identificar y desarrollar estrategias efectivas de generación estática y dinámica de vectores de peso para Algoritmos Evolutivos Multiobjetivo con un enfoque particular al mantenimiento de la diversidad. En primer lugar, se analizó el rendimiento de dos grupos de técnicas de generación de vectores de pesos estáticos: i) dos métodos basados en diseños de mezcla, y ii) cuatro técnicas basadas en funciones de baja discrepancia. Para la comparación experimental de estas técnicas se adoptaron siete MOPs con diferentes geometrías de frente de Pareto con instancias con 3, 5 y 8 objetivos. Para la evaluación se utilizaron tanto indicadores de convergencia como de diversidad. Después de este análisis, el segundo objetivo consistió en introducir un nuevo enfoque para actualizar los vectores de pesos durante la búsqueda con el fin de hacer frente a una amplia variedad de geometrías de frente de Pareto. La estrategia de adaptación dinámica de vectores de pesos utiliza la distancia de Mahalanobis y puntos tabú para delimitar las regiones donde se encuentran soluciones con base en la información obtenida durante el proceso evolutivo. Con el fin de mantener el número de vectores de pesos constante, aquellos apuntando áreas saturadas del espacio de soluciones son redistribuidos y actualizados para mejorar la búsqueda en regiones no exploradas o poco saturadas del espacio solución.

De la primera comparación de las técnicas de generación estática, se obtuvieron las siguientes observaciones generales. Para los problemas con frentes de Pareto lineales o simétricos convexos, las técnicas basadas en diseño de mezclas (Simplex Lattice y Diseños de dos capas) obtuvieron los mejores resultados tanto en convergencia como en distribución para cualquier número de objetivos. En cuanto a las superficies no convexas o desconectadas, los métodos de diseño basados en las funciones de baja discrepancia (Diseño uniforme, Halton, Hammersley y Sobol) proporcionaron un buen rendimiento para 5 y 8 objetivos. Por último, para los MOP con un frente de Pareto invertido, las técnicas de baja discrepancia mejoran su rendimiento a

medida que aumenta el número de objetivos.

Los experimentos realizados con la estrategia de adaptación de pesos propuesta proporcionaron resultados prometedores. En efecto, la capacidad de esta nueva técnica para modificar la distribución de pesos en función de la superficie expone resultados notables en 3, 5 y 8 dimensiones. Sin embargo y a pesar de los buenos resultados, la estrategia adaptativa presenta puntos débiles que pueden mejorarse en trabajos futuros.

6.2. Trabajo futuro

Las primeras tendencias observadas en este trabajo permiten identificar algunos puntos débiles, como la pérdida de los puntos extremos frente de Pareto durante los procesos de reubicación de puntos o el impacto de la frecuencia de reubicación de puntos no tabú sobre la búsqueda de las fronteras en frentes de Pareto no conexos. Estas fronteras pueden no quedar representadas en el espacio de los vectores de pesos y por ser excluidas en el resultado final de la ejecución del algoritmo, la aplicación de técnicas que permitan preservar estos puntos dentro de los puntos tabú conforme a otras características medibles a partir de la información que se obtiene dentro del mismo proceso evolutivo podría volver mas robusta la técnica de búsqueda.

La estrategia propuesta merece, pues, una experimentación exhaustiva, con diferentes funciones de prueba y formas de frentes, por ejemplo, frentes degenerados. La misma técnica podría integrarse en *NSGA-III* que diferenciándose de su antecesor, propone sustituir el operador *Crowding distance* (que evalúa la densidad de las soluciones en distintas regiones del espacio de los objetivos) por un método de nichos centrados sobre puntos de referencia en el espacio de los objetivos, de tal manera que ayuden a mejorar la diversidad de la población. Este cambio central en *NSGA-III* utilizando puntos de referencia y el mismo proceso evolutivo permiten que se pueda adaptar la estrategia de adaptación dinámica de vectores de peso propuesta para evaluar su robustez con respecto al motor de búsqueda.

La selección de la técnica de reubicación de puntos de referencia dentro de los límites marcados por los puntos tabú puede incluir aproximaciones que permita distinguir regiones que ya presentan una alta densidad de soluciones asociadas. Esto permitiría evitar cálculos innecesarios y movimientos del mismo punto de referencia en distintas fases de reubicación, siempre guiadas por la información obtenida del proceso evolutivo.

Bibliografía

- [1] Ahrari, A., Deb, K., and Preuss, M. Multimodal optimization by covariance matrix self-adaptation evolution strategy with repelling subpopulations. *Evolutionary computation* (2016).
- [2] Asafuddoula, M., Singh, H. K., and Ray, T. An enhanced decomposition-based evolutionary algorithm with adaptive reference vectors. *IEEE transactions on cybernetics* 48, 8 (2018), 2321–2334.
- [3] Cai, X., Mei, Z., and Fan, Z. A decomposition-based many-objective evolutionary algorithm with two types of adjustments for direction vectors. *IEEE transactions on cybernetics* 48, 8 (2018), 2335–2348.
- [4] Cheng, R., Jin, Y., Olhofer, M., and Sendhoff, B. A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Trans. Evolutionary Computation* 20, 5 (2016), 773–791.
- [5] Coello, C. A. An updated survey of ga-based multiobjective optimization techniques. *ACM Computing Surveys (CSUR)* 32, 2 (2000), 109–143.
- [6] Coello Coello, Carlos A. y Lamont, G. B. *Applications of Multi-objective Evolutionary Algorithms*. Advances In Natural Computation. World Scientific Publishing Company, 2004.
- [7] Cornell, J. A. *Experiments with mixtures: designs, models, and the analysis of mixture data*, vol. 403. John Wiley & Sons, 2011.
- [8] Das, I., and Dennis, J. E. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM J. on Optimization* 8, 3 (Mar. 1998), 631–657.
- [9] Deb, K., and Jain, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box

- constraints. *IEEE Transactions on Evolutionary Computation* 18, 4 (Aug 2014), 577–601.
- [10] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Trans. Evol. Comp* 6, 2 (Apr. 2002), 182–197.
- [11] Fang, K., and Lin, D. J. uniform designs and their application in industry. *Handbook on Statistics in Industry. Amsterdam: Elsevier* (2003), 131–170.
- [12] Fang, K.-T. Uniform design: application of number-theoretic methods in experimental design. *Acta Math. Appl. Sin.* 3 (1980), 363–372.
- [13] Fang, K.-T., and Qin, H. A note on construction of nearly uniform designs with large number of runs. *Statistics & probability letters* 61, 2 (2003), 215–224.
- [14] Fang, K.-T., Wang, Y., and Bentler, P. M. Some applications of number-theoretic methods in statistics. *Statistical Science* (1994), 416–428.
- [15] Fleischer, M. The measure of pareto optima applications to multi-objective metaheuristics. In *International Conference on Evolutionary Multi-Criterion Optimization* (2003), Springer, pp. 519–533.
- [16] Gentle, J. E. *Random number generation and Monte Carlo methods*. Springer Science & Business Media, 2006.
- [17] Giagkiozis, I., Purshouse, R. C., and Fleming, P. J. Towards understanding the cost of adaptation in decomposition-based optimization algorithms. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on* (2013), IEEE, pp. 615–620.
- [18] Glover, F. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research* 13, 5 (jan 1986), 533–549.
- [19] Goldberg, D. E., and Holland, J. H. Genetic algorithms and machine learning. *Machine learning* 3, 2 (1988), 95–99.
- [20] Gu, F., and Cheung, Y.-M. Self-organizing map-based weight design for decomposition-based many-objective evolutionary algorithm. *IEEE Transactions on Evolutionary Computation* 22, 2 (2018), 211–225.
- [21] Gu, F.-Q., and Liu, H.-L. A novel weight design in multi-objective evolutionary algorithm. In *Computational Intelligence and Security (CIS), 2010 International Conference on* (2010), IEEE, pp. 137–141.
- [22] Hammersley, J. *Monte carlo methods*. Springer Science & Business Media, 2013.
-

-
- [23] Hansen, N., Müller, S. D., and Koumoutsakos, P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary computation* 11, 1 (2003), 1–18.
- [24] Hofer, R., Kritzer, P., Larcher, G., and Pillichshammer, F. Distribution properties of generalized van der corput–halton sequences and their subsequences. *International Journal of Number Theory* 5, 04 (2009), 719–746.
- [25] Holland, J. H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1992.
- [26] Hua, L.-K., and Wang, Y. *Applications of number theory to numerical analysis*. Springer Science & Business Media, 2012.
- [27] Ishibuchi, H., and Murata, T. A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 28, 3 (1998), 392–403.
- [28] Ishibuchi, H., Setoguchi, Y., Masuda, H., and Nojima, Y. Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes. *IEEE Trans. Evolutionary Computation* 21, 2 (2017), 169–190.
- [29] Jain, H., and Deb, K. An improved adaptive approach for elitist nondominated sorting genetic algorithm for many-objective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization* (2013), Springer, pp. 307–321.
- [30] Jiang, S., Cai, Z., Zhang, J., and Ong, Y.-S. Multiobjective optimization by decomposition with pareto-adaptive weight vectors. In *Natural Computation (ICNC), 2011 Seventh International Conference on* (2011), vol. 3, IEEE, pp. 1260–1264.
- [31] Kukkonen, S., and Deb, K. A fast and effective method for pruning of non-dominated solutions in many-objective problems. In *Parallel Problem Solving from Nature-PPSN IX*. Springer, 2006, pp. 553–562.
- [32] Li, H., Ding, M., Deng, J., and Zhang, Q. On the use of random weights in moea/d. In *Evolutionary Computation (CEC), 2015 IEEE Congress on* (2015), IEEE, pp. 978–985.
- [33] Li, H., and Landa-Silva, D. An adaptive evolutionary multi-objective approach based on simulated annealing. *Evolutionary Computation* 19, 4 (2011), 561–595.
- [34] Li, M., and Yao, X. What weights work for you? adapting weights for any pareto front shape in decomposition-based evolutionary multi-objective optimisation. *arXiv preprint arXiv:1709.02679* (2017).
-

-
- [35] Ma, C., and Fang, K.-T. A new approach to construction of nearly uniform designs. *International Journal of Materials and Product Technology* 20, 1-3 (2004), 115–126.
- [36] Mahalanobis, P. C. On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India, 1936* (1936), 49–55.
- [37] Messac, A., Ismail-Yahaya, A., and Mattson, C. The normalized normal constraint method for generating the pareto frontier. *Structural and Multidisciplinary Optimization* 25, 2 (Jul 2003), 86–98.
- [38] Miettinen, K. *Nonlinear multiobjective optimization*. Kluwer Academic Publishers, Boston, 1999.
- [39] Osyczka, A. 7 - multicriteria optimization for engineering design. In *Design Optimization*, J. S. Gero, Ed. Academic Press, 1985, pp. 193 – 227.
- [40] Qi, Y., et al. MOEA/D with a delaunay triangulation based weight adjustment. In *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation* (2014), ACM, pp. 93–94.
- [41] Qi, Y., et al. Moea/d with adaptive weight adjustment. *Evolutionary Computation* 22, 2 (2014), 231–264. PMID: 23777254.
- [42] Rodríguez Sánchez, A., Ponsich, A., López Jaimes, A., and Zapotecas Martínez, S. A parallel tabu search heuristic to approximate uniform designs for reference set based moeas. In *EMO 2019* (2019), Springer, LNCS, p. In press.
- [43] Roth, K. F. Rational approximations to algebraic numbers. *Mathematika* 2, 1 (1955), 1–20.
- [44] Rudolph, G. Convergence analysis of canonical genetic algorithms. *IEEE transactions on neural networks* 5, 1 (1994), 96–101.
- [45] Sánchez, A. R., Ponsich, A., and Jaimes, A. L. Generation techniques and a novel on-line adaptation strategy for weight vectors within decomposition-based moeas. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (2019), pp. 229–230.
- [46] Scheffé, H. Experiments with mixtures. *Journal of the Royal Statistical Society. Series B (Methodological)* (1958), 344–360.
- [47] Tan, Y.-Y., et al. Moea/d+ uniform design: A new version of moea/d for optimization problems with many objectives. *Computers & Operations Research* 40, 6 (2013), 1648–1660.
-

-
- [48] Wang, Y., and Hickernell, F. J. An historical overview of lattice point sets. In *Monte Carlo and Quasi-Monte Carlo Methods 2000*. Springer, 2002, pp. 158–167.
- [49] Xiang, Y., Zhou, Y., Li, M., and Chen, Z. A vector angle-based evolutionary algorithm for unconstrained many-objective optimization. *IEEE Transactions on Evolutionary Computation* 21, 1 (2017), 131–152.
- [50] Xie, M.-Y., and Fang, K.-T. Admissibility and minimaxity of the uniform design measure in nonparametric regression model. *Journal of statistical planning and inference* 83, 1 (2000), 101–111.
- [51] y Luís Paquete y Manuel López-Ibáñez, C. M. F. An improved dimension-sweep algorithm for the hypervolume indicator. In *Proceedings of the 2006 Congress on Evolutionary Computation (CEC 2006)*. IEEE Press, Piscataway, NJ, July 2006, pp. 1157–1163.
- [52] Zapotecas-Martínez, S., Aguirre, H. E., Tanaka, K., and Coello Coello, C. A. On the low-discrepancy sequences and their use in moea/d for high-dimensional objective spaces. In *Evolutionary Computation (CEC), 2015 IEEE Congress on (2015)*, IEEE, pp. 2835–2842.
- [53] Zapotecas-Martínez, S., Coello, C. A. C., Aguirre, H. E., and Tanaka, K. A review of features and limitations of existing scalable multi-objective test suites. *IEEE Transactions on Evolutionary Computation* (2018), 1–1.
- [54] Zapotecas-Martínez, S., López-Jaimes, A., and García-Nájera, A. Libea: A lebesgue indicator-based evolutionary algorithm for multi-objective optimization. *Swarm and Evolutionary Computation* (2018).
- [55] Zhang, C., Tan, K. C., Lee, L. H., and Gao, L. Adjust weight vectors in moea/d for bi-objective optimization problems with discontinuous pareto fronts. *Soft Computing* 22, 12 (2018), 3997–4012.
- [56] Zhang, Q., et al. Multiobjective optimization test instances for the cec 2009 special session and competition. *University of Essex, Colchester, UK and Nanyang technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, technical report 264* (2008).
- [57] Zhang, Q., and Li, H. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation* 11, 6 (Dec 2007), 712–731.
- [58] Zhang, Q., and Li, H. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation* 11, 6 (December 2007), 712–731.
-

- [59] Zhang, Q., Liu, W., and Li, H. The performance of a new version of moea/d on cec09 unconstrained mop test instances. In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on (2009)*, IEEE, pp. 203–208.
-