



TRABAJO FIN DE GRADO  
GRADO EN INGENIERÍA INFORMÁTICA

**Análisis Multi-Modal y Multi-Objetivo al Problema de  
Clustering con Restricciones**

---

**Autor**

Alfonso García Martínez

**Director**

Salvador García López



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE  
TELECOMUNICACIÓN

—  
Granada, septiembre de 2021



**Análisis Multi-Modal y Multi-Objetivo al Problema de  
Clustering con Restricciones**

**Autor**

Alfonso García Martínez

**Director**

Salvador García López

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA  
ARTIFICIAL



# **Análisis Multi-Modal y Multi-Objetivo al Problema de Clustering con Restricciones**

Alfonso García Martínez

**Palabras clave:** aprendizaje automático, clustering con restricciones, aprendizaje semisupervisado, metaheurísticas, algoritmos evolutivos, optimización multiobjetivo, optimización multimodal

## **Resumen**

Poner aquí el resumen.



# **Multi-Modal and Multi-Objective Analysis for Constrained Clustering**

Alfonso García Martínez

**Keywords:** machine learning, constrained clustering, semi-supervised learning, metaheuristics, evolutionary algorithms, multi-objective optimization, multimodal optimization

## **Abstract**

Write here the abstract in English.





---

---

D. **Salvador García López**, Profesor del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada.

**Informa:**

Que el presente trabajo, titulado *Análisis Multi-Modal y Multi-Objetivo al Problema de Clustering con Restricciones*, ha sido realizado bajo su supervisión por **Alfonso García Martínez**, y autorizo la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expide y firma el presente informe en Granada a X de mes de 2021.

**El director:**

**Salvador García López**



## AGRADECIMIENTOS

---

A mis padres.

A mi director, Salvador.

Y, especialmente, a mi mentor **Germán González Almagro**. Su tiempo, su ayuda, su apoyo y su asesoramiento han sido imprescindibles para poder completar este trabajo.



## ÍNDICE GENERAL

---

1	INTRODUCCIÓN	1
1.1	Estructura y objetivos de este trabajo . . . . .	3
I	CONTEXTO TEÓRICO	
2	INTRODUCCIÓN AL PROBLEMA DEL CLUSTERING	7
2.1	Introducción . . . . .	7
2.2	Ejemplos de uso . . . . .	7
2.3	Definiciones formales . . . . .	8
2.3.1	Notación . . . . .	8
2.3.2	Tipos de algoritmos de clustering . . . . .	9
2.3.3	Índices de validación . . . . .	10
2.3.4	Clustering como problema de optimización . .	12
3	EL PROBLEMA DEL CLUSTERING CON RESTRICCIONES	13
3.1	Introducción al aprendizaje semi-supervisado . . . . .	13
3.2	Clustering con restricciones . . . . .	15
3.2.1	Ventajas e inconvenientes del uso de restricciones en clustering . . . . .	17
3.2.2	Ejemplos de uso . . . . .	19
4	LOS ALGORITMOS EVOLUTIVOS MULTIOBJETIVO	21
4.1	Optimización multiobjetivo . . . . .	21
4.1.1	Problema de optimización de un objetivo . . . .	21
4.1.2	Problema de optimización multiobjetivo . . . .	22
4.2	Algoritmos evolutivos . . . . .	24
4.2.1	Metaheurísticas . . . . .	25
4.2.2	Conceptos básicos de los algoritmos evolutivos	25
4.2.3	Algoritmos evolutivos para clustering y clustering con restricciones . . . . .	27
4.3	Algoritmos evolutivos multiobjetivo . . . . .	30
5	ALGORITMOS EVOLUTIVOS MULTIOBJETIVO BASADOS EN DESCOMPOSICIÓN	33
5.1	Multiobjective Evolutionary Algorithm based on Decomposition . . . . .	33
5.1.1	Funcionamiento general . . . . .	33
5.1.2	Métodos de descomposición . . . . .	35
5.2	El framework ADA . . . . .	39
5.2.1	Algoritmos evolutivos multimodales multiobjetivo basados en descomposición . . . . .	39
5.2.2	ADA: un framework para manejar la multimodalidad . . . . .	39
5.2.3	Operadores de ADA . . . . .	40

II	EXPERIMENTACIÓN	
6	METODOLOGÍA	51
6.1	Esquema de codificación . . . . .	51
6.2	Funciones objetivo a optimizar . . . . .	52
6.3	Variantes de ADA . . . . .	52
6.3.1	Inclusión de población externa . . . . .	53
6.3.2	Métrica de distancia en el espacio de soluciones . . . . .	54
6.4	Parámetros . . . . .	55
6.5	Conjuntos de datos y de restricciones . . . . .	56
6.6	Ejecución y métricas de evaluación . . . . .	58
6.6.1	Métricas a nivel de partición . . . . .	58
6.6.2	Métricas a nivel de frente de Pareto . . . . .	60
6.6.3	Ejecución de los experimentos . . . . .	61
7	IMPLEMENTACIÓN	63
7.1	Librerías utilizadas . . . . .	63
7.2	Estructura del software . . . . .	63
8	RESULTADOS	65
8.1	Tablas de resultados y primeras observaciones . . . . .	65
8.2	Análisis estadístico de los resultados . . . . .	72
8.3	Conclusiones . . . . .	72
8.4	Posibles trabajos futuros . . . . .	72
	BIBLIOGRAFÍA	73

## ÍNDICE DE FIGURAS

Figura 2.1	Ejemplo de tres clusters claramente distinguibles y separables con densidades distintas. . .	8
Figura 3.1	Arriba, un pequeño conjunto de datos con restricciones ML y CL. Abajo, una solución de clustering para ese conjunto de datos que satisface a todas las restricciones. . . . .	16
Figura 3.2	Arriba, un pequeño conjunto de datos con restricciones ML y CL. Abajo, una solución de clustering para ese conjunto de datos después de haber modificado las distancias entre las instancias teniendo en cuenta esas restricciones. . .	17
Figura 4.1	Representación gráfica de un frente de Pareto	24
Figura 4.2	Codificación real. . . . .	26
Figura 4.3	Codificación binaria para $K = 3$ y $K = 5$ clusters. . . . .	28
Figura 4.4	Codificación entera basada en etiquetas. . . . .	29
Figura 4.5	Codificación entera basada en grafos. . . . .	30
Figura 4.6	Codificación real. . . . .	31
Figura 5.1	Diagrama de flujo del MOEA/D . . . . .	34

Figura 5.2	Ilustración geométrica de los métodos BI y PBI para un problema de maximización. . . . .	38
Figura 5.3	Ilustración del concepto de multimodalidad, en la cual soluciones muy dispares en el espacio de soluciones son casi idénticas en el espacio objetivo. . . . .	41
Figura 5.4	Diagrama de flujo del esquema ADA . . . . .	43
Figura 5.5	Ejemplo que ilustra cómo las soluciones asignadas al mismo subproblema/vector de pesos son cercanas entre sí en el espacio objetivo. . .	44

## ÍNDICE DE TABLAS

---

Tabla 6.1	Variantes de ADA comparadas . . . . .	54
Tabla 6.2	Valores escogidos para los parámetros e hiperparámetros de ADA . . . . .	56
Tabla 6.3	Resumen de los conjuntos de datos . . . . .	57
Tabla 6.4	Tamaños de los conjuntos de restricciones . . .	58
Tabla 8.1	Mejores puntuaciones ARI - 10 % restricciones	66
Tabla 8.2	Mejores puntuaciones ARI - 15 % restricciones	66
Tabla 8.3	Mejores puntuaciones ARI - 20 % restricciones	67
Tabla 8.4	Restricciones no satisfechas (UNSAT) - 10 % restricciones . . . . .	67
Tabla 8.5	Restricciones no satisfechas (UNSAT) - 15 % restricciones . . . . .	68
Tabla 8.6	Restricciones no satisfechas (UNSAT) - 20 % restricciones . . . . .	68
Tabla 8.7	Hipervolúmenes - 10 % restricciones . . . . .	69
Tabla 8.8	Hipervolúmenes - 15 % restricciones . . . . .	69
Tabla 8.9	Hipervolúmenes - 20 % restricciones . . . . .	70
Tabla 8.10	Tamaños frentes de Pareto - 10 % restricciones	70
Tabla 8.11	Tamaños frentes de Pareto - 15 % restricciones	71
Tabla 8.12	Tamaños frentes de Pareto - 20 % restricciones	71

## ÍNDICE DE ALGORITMOS

---

Algoritmo 1	MOEA/D . . . . .	45
Algoritmo 2	El framework ADA . . . . .	46
Algoritmo 3	MOEA/D-AD . . . . .	47

## ACRÓNIMOS

---



## INTRODUCCIÓN

---

Vivimos en un mundo de datos. Cada minuto, cada segundo, se generan y almacenan cantidades masivas de datos en internet, en las redes sociales y en bases de datos de todo el mundo. Y no solo eso, sino que además la cantidad de datos que se crean es cada vez mayor: con la aparición del Internet de las Cosas (*Internet of Things*), la generación de nuevos datos ha experimentado una gran aceleración gracias a fuentes de datos tan cotidianos como teléfonos móviles, cámaras, micrófonos, sensores biométricos, etc. Al ritmo actual, se estima que en el año 2025 existirán un total de 163 zettabytes de datos en todo el mundo [1].

Por todo ello, las técnicas de minería de datos han cobrado una especial relevancia en los últimos años, ya que pueden resultar muy útiles para extraer patrones e información valiosa de las bases de datos masivas. Un cierto tipo de algoritmos, los llamados algoritmos de aprendizaje automático, constituyen toda una rama de la inteligencia artificial y tienen la particularidad de que son capaces de aprender a partir de datos que reciben como entrada, por lo que resultan idóneos para aprovechar todos esos datos. La primera vez que se habló del concepto de *aprendizaje automático* (originalmente *machine learning*) se remonta al 1959, año en que Arthur Samuel, por aquel entonces investigador de IBM y pionero de la inteligencia artificial, acuñó el término [2]. Desde entonces, han surgido numerosas definiciones de lo que es el aprendizaje automático. Una definición bastante formal y de las más usadas a la hora de referirse a los algoritmos de aprendizaje quizá sea la siguiente: «Un programa se dice que aprende de la experiencia  $E$  con respecto a una clase de tareas  $T$  y una medida de rendimiento  $P$  si su rendimiento en la tarea  $T$ , medida con  $P$ , mejora con la experiencia  $E$ .»[3]

Tradicionalmente, las técnicas de aprendizaje automático se han agrupado en tres grandes categorías [4]:

- **Aprendizaje supervisado.** En este tipo de aprendizaje, los datos de entrenamiento (es decir, los datos usados para que el modelo *aprenda*) contienen ejemplos explícitos de cuáles serían las etiquetas o valores de salida (*outputs*) correctos. Un buen ejemplo de este enfoque sería la clasificación de imágenes con dígitos escritos a mano, donde es posible asignar a cada imagen una etiqueta para indicarle al modelo a qué dígito del 0 al 9 corresponde. Este tipo de problema se conoce como clasificación, el cual, junto al problema de la regresión, constituye la variante más típica del aprendizaje supervisado.

- **Aprendizaje no supervisado.** En este enfoque, los datos de entrenamiento no contienen ninguna información de salida, tan solo se dispone de datos no etiquetados. Aún sin esa información, es posible extraer información de los datos, encontrando patrones, agrupaciones o estructuras intrínsecos en ellos, por medio de técnicas como el *clustering*, que más adelante se verá con detalle.
- **Aprendizaje por refuerzo.** Al igual que en el aprendizaje no supervisado, en el aprendizaje por refuerzo tampoco se dispone de información explícita que indique cuál es la salida correcta. En su lugar, por cada salida dada por el modelo se dispone de un grado de bondad de esa salida obtenida, de modo que es posible saber cómo de buena ha sido. De esta forma, el modelo es capaz de ajustarse por sí solo sucesivas veces hasta obtener el mejor grado de bondad posible.

Además de estas tres, existe una cuarta categoría adicional llamada **aprendizaje semisupervisado**, que, simplificando en exceso, consiste en un híbrido entre el aprendizaje supervisado y el no supervisado. De esta nueva categoría forma parte lo que se conoce como *clustering con restricciones*, una variante del clustering convencional en la que se introduce algo de conocimiento sobre la solución óptima (es decir, la solución verdadera) en forma de restricciones que se deben satisfacer. Todos estos conceptos se estudiarán con mayor profundidad más adelante.

También se verá que tanto el problema del clustering clásico como el del clustering con restricciones pueden ser enfocados como un problema de optimización convencional. Y de hecho, también pueden formularse como problemas de optimización con más de una función objetivo a minimizar o maximizar, lo que se conoce como *optimización multiobjetivo*.

Como tales, para resolverlos pueden utilizarse los llamados *algoritmos evolutivos*, un tipo de algoritmos de optimización aproximativa inspirados en la biología y la evolución de las especies. De ellos, aquellos que son capaces de lidiar con la optimización multiobjetivo reciben el nombre de *algoritmos evolutivos multiobjetivo*. Una clase concreta de este tipo de algoritmos que han demostrado ser bastante prometedores son los algoritmos evolutivos multiobjetivo basados en descomposición, los cuales funcionan descomponiendo un problema de optimización multiobjetivo en varios subproblemas de optimización de un objetivo.

Más recientemente, han surgido variantes de estos algoritmos que son capaces de manejar la multimodalidad, un fenómeno que se da en los problemas de optimización cuando soluciones muy diferentes ofrecen resultados casi idénticos o de una calidad muy similar. En concreto, el método ADA, propuesto en 2019, es capaz de adaptar un algoritmo evolutivo multiobjetivo basado en descomposición para

que sea capaz de afrontar problemas multimodales, aportando para ello un mecanismo para mantener la diversidad en el espacio de soluciones.

### 1.1 ESTRUCTURA Y OBJETIVOS DE ESTE TRABAJO

Este Trabajo de Fin de Grado tiene como objetivo aplicar distintas versiones de un algoritmo evolutivo multiobjetivo basado en descomposición modificado con ADA para resolver problemas de clustering con restricciones, los cuales presentan un alto grado de multimodalidad. No hay constancia de que se haya empleado ADA para resolver problemas de clustering con restricciones con anterioridad, por lo que, en este sentido, este trabajo será pionero.

La primera parte de este trabajo servirá para situar al lector en el contexto teórico del que forman parte tanto el problema a resolver como los métodos utilizados. Se presentarán los problemas del clustering clásico y el clustering con restricciones. Se explicará en qué consiste la optimización multiobjetivo y cuáles son sus particularidades frente a la optimización de un solo objetivo. Se hablará brevemente de los llamados algoritmos evolutivos y, más concretamente, de los algoritmos evolutivos multiobjetivo, para finalmente describir con detalle los algoritmos evolutivos multiobjetivo basados en descomposición: cuál es el algoritmo más conocido, cómo funciona la idea de *descomposición*, cómo es posible introducir diversidad en el espacio de soluciones para afrontar la multimodalidad, cómo funciona exactamente ADA, etc.

En la segunda parte se describirá la experimentación seguida para este trabajo y se analizarán los resultados obtenidos. Se enumerarán los conjuntos de datos empleados y se describirá sus características. Se describirá la metodología seguida en el experimento, listando las variantes de ADA con las que se experimentará, así como los parámetros e hiperparámetros elegidos. También se hablará de la estructura del software desarrollado y de los detalles de la implementación de los algoritmos. Finalmente, se llevará a cabo un análisis estadístico de los resultados para determinar si las diferencias entre las distintas versiones de ADA usadas son significativas.



Parte I

CONTEXTO TEÓRICO



## INTRODUCCIÓN AL PROBLEMA DEL CLUSTERING

---

Para poder entender adecuadamente los algoritmos que han sido usados en ese trabajo, así como el tipo de problemas a los que se han aplicado, es necesario introducir una serie de conceptos teóricos. Comenzaremos explicando en este capítulo el problema del clustering clásico.

### 2.1 INTRODUCCIÓN

El problema de la agrupación o *clustering* es un problema de aprendizaje no supervisado que consiste, como su propio nombre indica, en agrupar un conjunto de objetos, puntos, datos o ítems según una serie de características que tengan en común, de forma que los objetos que formen parte de un mismo *cluster* sean de alguna forma similares entre sí, y que al mismo tiempo sean distintos de los que formen parte de clusters ajenos. Podemos entender, por tanto, que el clustering no es más que una forma de clasificar objetos.

El clustering y la clasificación en general son útiles para mantener un sumario que organice un conjunto de datos de gran tamaño, lo que permite un entendimiento más fácil y eficiente de la información contenida en él. Los primeros intentos de uso de métodos numéricos para la agrupación sistemática de objetos tuvieron lugar en las ciencias naturales, tales como la biología y la zoología, con el fin de obtener clasificaciones más objetivas y estables que permitiesen prescindir de la subjetividad de las técnicas tradicionales de taxonomía. Estas técnicas numéricas de agrupación han recibido distintos nombres según el campo en el que se han usado: *taxonomía numérica* en biología, *segmentación* en investigación de mercados, a veces *Q-análisis* en psicología... En general, *análisis de cluster* o simplemente *clustering* quizá sean los términos genéricos con mayor aceptación. [5]

### 2.2 EJEMPLOS DE USO

En [5] podemos encontrar algunos ejemplos de aplicación del clustering:

- **Investigación de mercados:** clasificación de compañías según características financieras; clasificación de ciudades según tamaño, renta per cápita, etc [6], clasificación de clientes para encontrar compradores potenciales de un cierto producto [7].

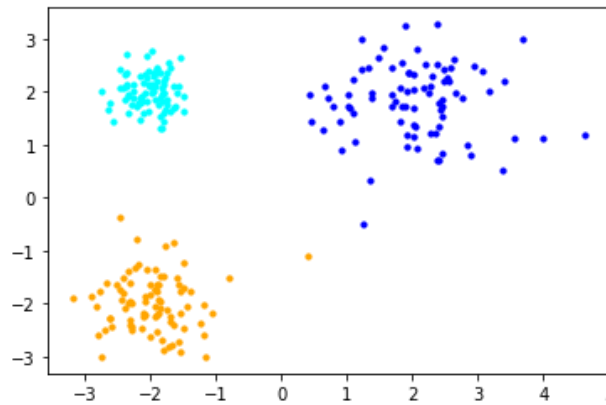


Figura 2.1: Ejemplo de tres clusters claramente distinguibles y separables con densidades distintas.

- **Astronomía:** clasificación de nebulosas según composición química [8], clasificación de estrellas según su velocidad hacia el centro de la galaxia y su velocidad de rotación alrededor de la galaxia [9].
- **Arqueología:** creación de una taxonomía de hachas encontradas en las islas británicas según tamaño y tipo de punta [10], análisis de coprolitos para estudiar el maíz presente en la alimentación [11].
- **Bioinformática y genética:** uso de clustering en datos genéticos para identificar tipos de cáncer asociados a una mayor probabilidad de supervivencia [12].

### 2.3 DEFINICIONES FORMALES

Hasta ahora, se ha dado una interpretación bastante intuitiva acerca de en qué consiste el clustering. Llega el momento de definir formalmente el problema del clustering e introducir su notación. Para tal fin, en toda esta sección nos basaremos en el trabajo realizado en [13].

#### 2.3.1 Notación

Los siguientes son elementos típicos de un problema de clustering:

- Un *objeto*, *patrón* o *instancia* es un dato individual representado por un vector  $\mathbf{x} = (x_1, x_2, \dots, x_D)^T$ , donde  $x_i \in \mathbb{R}$  es una *característica* o atributo, y  $D$  representa la *dimensionalidad* (el número de características que tienen los objetos).



- Un *conjunto de datos* (o *dataset* en inglés) es una matriz  $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$ , donde  $\mathbf{x}_i \in \mathbb{R}^D$  para  $i = 1, \dots, N$ , y  $N$  es el número total de objetos en el conjunto de datos.
- Un *cluster* puede definirse como una región del espacio que contiene objetos.
- Una *solución de clustering*  $\mathbf{C} = \{c_k | k = 1, \dots, K\}$  es un conjunto de clusters disjuntos que particiona al conjunto de datos  $\mathbf{X}$  en  $K$  grupos.
- El número de objetos en el cluster  $c_k$  es el valor expresado como  $n_k = |c_k|$
- El *centroide* o *prototipo* del cluster  $c_k$  (o prototipo de  $c_k$ ) se expresa como  $\bar{\mathbf{c}}_k = \frac{1}{n_k} \sum_{\mathbf{x}_i \in c_k} \mathbf{x}_i$ .
- El centroide del conjunto de datos  $\mathbf{X}$  o *centroide global* de  $\mathbf{X}$  es  $\bar{\mathbf{X}} = \frac{1}{N} \sum_{\mathbf{x}_i \in \mathbf{X}} \mathbf{x}_i$
- Una *medida de distancia* es una métrica usada para cuantificar la proximidad (cercanía o lejanía) entre objetos.
- Un *índice de validez* o CVI (*cluster validity index*) usa una medida de distancia para evaluar cuantitativamente una solución de clustering obtenida.

### 2.3.2 Tipos de algoritmos de clustering

El clustering no es un algoritmo en sí, sino el tipo de problema que se pretende resolver, para el cual pueden usarse diferentes técnicas. Tradicionalmente, se han considerado dos tipos de algoritmos para el problema del clustering:

- **Algoritmos de clustering particional:** estos algoritmos dividen directamente el conjunto de datos en un determinado número de clusters. De esta forma, un conjunto de datos  $\mathbf{X}$  es particionado en  $K$  grupos (o clusters) disjuntos  $\mathbf{C} = \{c_1, c_2, \dots, c_K\}$ , de forma que se cumplan tres condiciones:
  1. No puede haber clusters vacíos:  $c_i \neq \emptyset$ ,  $i = 1, \dots, K$ .
  2. La unión de todos los clusters es igual al conjunto de datos:  $\bigcup_{i=1}^K c_i = \mathbf{X}$ .
  3. Los clusters son disjuntos, es decir, no hay solapamientos:  $c_i \cap c_j = \emptyset$ ,  $i, j = 1, \dots, K$  y además  $i \neq j$ .

El algoritmo de clustering particional más conocido probablemente sea el  $k$ -means [14], que funciona asignando cada objeto al cluster cuyo centroide se encuentre más cerca, para posteriormente recalcular los centroides como los promedios de los

objetos asociados a los respectivos clusters, de forma que las distancias entre los objetos y los centroides de los clusters a los que están asignados se minimizan.

- **Algoritmos de clustering jerárquico:** los algoritmos de este tipo producen como salida una jerarquía de clusters llamada *dendograma*, que representa las agrupaciones anidadas de los objetos. Al crear un dendograma, se generan  $N$  niveles de clustering, de manera que cada nivel se construye a partir del anterior. De esta forma, no es necesario establecer un número de clusters a priori, puesto que en cada nivel se generan distintos números de clusters. Los dos métodos principales de clustering jerárquico son el aglomerativo y el divisivo.

### 2.3.3 Índices de validación

Un índice de validación o CVI (*cluster validity index*) evalúa cómo de buena es una solución (partición) de clustering que ha sido generada por un algoritmo, usando únicamente información inherente al conjunto de datos (es decir, sin tener conocimiento de cuál es la solución óptima, puesto que en aprendizaje no supervisado, en principio, no se sabe cuál es). Un buen índice de validación debería tener un significado intuitivo, debería ser fácil de calcular y debería ser justificable matemáticamente, además de que debería permitir establecer un orden entre las soluciones obtenidas según cómo de buenas son.

Dependiendo del índice que se vaya a emplear para evaluar soluciones, habrá que buscar la solución que maximice o minimice el índice escogido. Por tanto, podemos distinguir entre índices de maximización e índices de minimización.

Son numerosos los índices de validez existentes. A continuación, se listan algunos de los más utilizados:

- **Dispersión intra-grupo** (*within-group scatter*, WGS) [14]. Índice de minimización, también llamado dispersión intra-cluster. Mide la suma de distancias (euclídeas) al cuadrado entre los objetos y los centroides de los clusters a los que pertenecen:

$$\text{WGS}(\mathbf{C}) = \sum_{\mathbf{c}_k \in \mathbf{C}} \sum_{\mathbf{x}_i \in \mathbf{c}_k} d_e^2(\mathbf{x}_i, \bar{\mathbf{c}}_k), \quad (2.1)$$

donde  $d_e(\dots)$  representa la métrica de distancia euclídea entre dos puntos:

$$d_e(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}. \quad (2.2)$$

- **Dispersión entre grupos** (*between-group scatter*, BGS) [15]. Índice de maximización. Mide la suma de distancias entre los centroides de los clusters:

$$\text{BGS}(\mathbf{C}) = \sum_{\mathbf{c}_k, \mathbf{c}_r \in \mathbf{C}} d_e(\bar{\mathbf{c}}_k, \bar{\mathbf{c}}_r). \quad (2.3)$$

- **Índice de conectividad** (*connectivity index*) [16]. Índice de minimización. Mide el grado en el que objetos vecinos han sido asignados al mismo cluster:

$$\text{Conn}(\mathbf{C}) = \sum_{i=1}^N \left( \sum_{j=1}^L a_{i,n_{ij}} \right), \quad (2.4)$$

donde

$$a_{i,n_{ij}} = \begin{cases} \frac{1}{j} & \text{si } \neg \exists \mathbf{c}_k : \mathbf{x}_i \in \mathbf{c}_k \wedge \mathbf{x}_{n_{ij}} \in \mathbf{c}_k, \\ 0 & \text{en caso contrario} \end{cases}, \quad (2.5)$$

donde  $n_{ij}$  hace referencia al  $j$ -ésimo vecino más cercano del  $i$ -ésimo objeto. Por su parte,  $L$  es un parámetro que establece el número de vecinos a tener en cuenta al calcular el índice de conectividad.

Intuitivamente, lo que hace este índice es añadir una penalización cada vez que se detectan objetos vecinos (ceranos) pertenecientes a clusters distintos.

- **Índice Calinski-Harabasz** (CH) [17]. Índice de maximización. Calinski-Harabasz ofrece una proporción entre la separación y la cohesión de los clusters. La cohesión se estima como la suma de las distancias entre los objetos y sus respectivos centroides, y la separación se mide como la distancia entre los centroides de los clusters y el centroide global de todo el conjunto de datos:

$$\text{CH}(\mathbf{C}) = \frac{N - K}{K - 1} \times \frac{\sum_{\mathbf{c}_k \in \mathbf{C}} n_k d_e(\bar{\mathbf{c}}_k, \bar{\mathbf{X}})}{\sum_{\mathbf{c}_k \in \mathbf{C}} \sum_{\mathbf{x}_i \in \mathbf{c}_k} d_e(\mathbf{x}_i, \bar{\mathbf{c}}_k)}. \quad (2.6)$$

- **Índice Davies-Bouldin** (DB) [18]. Índice de minimización. En este caso, la cohesión es estimada con la distancia media entre los objetos y sus respectivos centroides, y la separación se calcula con la distancia entre centroides:

$$S(\mathbf{C}) = \frac{1}{K} \sum_{\mathbf{c}_k \in \mathbf{C}} \max_{\mathbf{c}_r \in \mathbf{C} \setminus \mathbf{c}_k} \left\{ \frac{S(\mathbf{c}_k) + S(\mathbf{c}_r)}{d_e(\bar{\mathbf{c}}_k, \bar{\mathbf{c}}_r)} \right\}, \quad (2.7)$$

donde

$$S(\mathbf{c}_k) = \frac{1}{n_k} \sum_{\mathbf{x}_i \in \mathbf{c}_k} d_e(\mathbf{x}_i, \bar{\mathbf{c}}_k). \quad (2.8)$$

- **Índice Silhouette (SI)** [19]. Índice de maximización. Se trata de una sumatoria normalizada, en la cual la cohesión se calcula como la suma de las distancias entre los puntos pertenecientes a un mismo cluster, y la separación se basa en la distancia entre los puntos más cercanos pertenecientes a clusters distintos:

$$SI(\mathbf{C}) = \frac{1}{N} \sum_{\mathbf{c}_k \in \mathbf{C}} \sum_{\mathbf{x}_i \in \mathbf{c}_k} \frac{b(\mathbf{x}_i, \mathbf{c}_k) - a(\mathbf{x}_i, \mathbf{c}_k)}{\max\{b(\mathbf{x}_i, \mathbf{c}_k), a(\mathbf{x}_i, \mathbf{c}_k)\}}, \quad (2.9)$$

donde:

$$a(\mathbf{x}_i, \mathbf{c}_k) = \frac{1}{n_k} \sum_{\mathbf{x}_j \in \mathbf{c}_k} d_e(\mathbf{x}_i, \mathbf{x}_j), \quad (2.10)$$

$$b(\mathbf{x}_i, \mathbf{c}_k) = \min_{\mathbf{c}_r \in \mathbf{C} \setminus \mathbf{c}_k} \left\{ \sum_{\mathbf{x}_j \in \mathbf{c}_r} d_e(\mathbf{x}_i, \mathbf{x}_j) \right\}. \quad (2.11)$$

#### 2.3.4 Clustering como problema de optimización

El problema del clustering puede formularse como un problema de optimización.

Sea  $\Omega$  el conjunto de todas las posibles soluciones de clustering (particiones) para un conjunto de datos  $\mathbf{X}$  dado, y sea  $f$  una función objetivo, que puede ser cualquiera de los índices de validez presentados en la sección 2.3.3. Suponiendo que se ha optado por utilizar un índice de validez de minimización, entonces el objetivo del problema de clustering  $(\Omega, f)$  consiste en encontrar la solución de clustering óptima  $\mathbf{C}^*$ , para la cual

$$f(\mathbf{C}^*) = \min\{f(\mathbf{C}) | \mathbf{C} \in \Omega\}. \quad (2.12)$$

## EL PROBLEMA DEL CLUSTERING CON RESTRICCIONES

---

El clustering con restricciones es un problema perteneciente al ámbito del aprendizaje semi-supervisado, tal y como se dijo en la introducción. Por ello, es lógico dedicar un apartado que sirva como introducción al aprendizaje semi-supervisado antes de comenzar a definir y explicar con detalle el clustering con restricciones. Para ello, se ha tomado como referencia principal el capítulo introductorio de [20].

### 3.1 INTRODUCCIÓN AL APRENDIZAJE SEMI-SUPERVISADO

El aprendizaje semi-supervisado es una de las cuatro modalidades existentes de aprendizaje automático, quizá la menos conocida de ellas. Podría decirse que es un híbrido entre aprendizaje supervisado y aprendizaje no supervisado.

En aprendizaje no supervisado (ver capítulo anterior), se dispone únicamente del conjunto de datos  $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$ , compuesto por  $N$  muestras o puntos, de forma que el objetivo consiste en encontrar patrones o estructuras en  $\mathbf{X}$  que puedan ser interesantes. Además del clustering o agrupación, existen otras formas de aprendizaje no supervisado, como la detección de *outliers* o las técnicas de reducción de dimensionalidad.

En aprendizaje supervisado, por su parte, se pretende encontrar una aplicación  $\mathbf{x} \mapsto y$  dado un conjunto de entrenamiento compuesto por pares  $(\mathbf{x}_i, y_i)$ . En este caso,  $y_i$  son las etiquetas de las muestras  $\mathbf{x}_i$ , de forma que, además de  $\mathbf{X}$ , se dispone también del vector de etiquetas  $\mathbf{y} = (y_1, \dots, y_N)^T$ . Nótese que para cada muestra siempre se dispone de una etiqueta. Si las etiquetas adoptan valores continuos ( $y_i \in \mathbb{R}$  o  $y_i \in \mathbb{R}^p$ ), la tarea se conoce como regresión, mientras que si adoptan valores discretos (es decir, valores de conjuntos finitos), se conoce como clasificación. Las dos grandes categorías de algoritmos de aprendizaje supervisado son los algoritmos generativos, los cuales pretenden calcular la distribución de probabilidad condicionada  $P(\mathbf{x}|y)$  para obtener a partir de ella la distribución  $P(y|\mathbf{x})$ , y los algoritmos discriminativos, que en general se limitan a obtener  $P(y|\mathbf{x})$ .

El aprendizaje semi-supervisado es una modalidad a medio camino entre el supervisado y el no supervisado. Efectivamente, se dispone del conjunto de datos sin etiquetas  $\mathbf{X}$ , pero además también se dispone de una cierta cantidad de información supervisada, aunque no para todas las muestras.

En muchas ocasiones, la información supervisada disponible son las etiquetas de algunos puntos de  $\mathbf{X}$ . En tales casos, el conjunto de datos  $\mathbf{X}$  puede dividirse en dos subconjuntos: por un lado, el conjunto  $\mathbf{X}_e = (\mathbf{x}_0, \dots, \mathbf{x}_e)$  para el cual se disponen de las etiquetas  $\mathbf{y}_e = (y_1, \dots, y_e)$ , y por otro lado el conjunto sin etiquetas  $\mathbf{X}_s = (\mathbf{x}_{e+1}, \dots, \mathbf{x}_{e+s})$ .

En otros casos, la información supervisada se da en forma de restricciones. Dichas restricciones indican si algunos ciertos pares de muestras tienen (o no) la misma etiqueta, o si forman parte o no del mismo cluster. A este tipo de aprendizaje semi-supervisado pertenece precisamente el problema que nos ocupa: el clustering con restricciones.

¿Cuándo tiene sentido usar técnicas semi-supervisadas, pudiendo emplear directamente aprendizaje supervisado? En los casos en los que obtener un etiquetado de los datos resulte demasiado costoso, cosa que suele ocurrir a menudo. Por ejemplo: generar conjuntos de datos con grabaciones de voz para un sistema de reconocimiento de voz es una tarea fácil, ya que pueden o bien grabarse expresamente para ello o bien recopilarse de grabaciones previas. Sin embargo, etiquetar ese conjunto de datos con grabaciones no es tan sencillo, pues requiere de alguien que escuche todas esas grabaciones y las transcriba. Otros ejemplos en los que esto también se cumple pueden ser la clasificación de documentos o páginas web, que necesitan que sean leídas por personas, o la clasificación de imágenes, pues igualmente hacen falta personas que vean esas imágenes e indiquen cuál es su contenido.

Dicho esto, ¿podemos esperar un buen rendimiento de las técnicas semi-supervisadas en comparación con las supervisadas? Lo cierto es que sí, pero es necesario asumir algunas hipótesis. Una de ellas, con respecto a la continuidad de la distribución de los datos, consiste en asumir que si dos puntos  $\mathbf{x}_1$  y  $\mathbf{x}_2$  en una región con alta densidad de puntos están cerca el uno del otro, entonces sus respectivos valores de salida  $y_1$  e  $y_2$  también deberían ser cercanos entre ellos. En el caso concreto del clustering, esto se traduce a que, si dos puntos pertenecen a la misma agrupación o cluster (región de alta densidad), entonces lo lógico sería pensar que pertenecen a la misma clase, o que poseen la misma etiqueta. De esta observación se puede inferir que las fronteras de decisión, o sea, los límites entre clusters distintos, deberían transcurrir por regiones de poca densidad, ya que si, de lo contrario, una frontera se situase en una región de alta densidad, un cluster quedaría cortado por dicha frontera. [20]

### 3.2 CLUSTERING CON RESTRICCIONES

Esta sección, centrada en describir y formular el problema del clustering con restricciones, tomará como referencia el análisis realizado por Davidson y Basu en [21].

El clustering con restricciones permite incorporar conocimiento experto sobre el problema en cuestión en forma de restricciones. Existen dos tipos de restricciones a nivel de instancia, definidas por primera vez en [22]:

- **Restricciones *Must-Link* (ML)**, denotadas  $c_=(\mathbf{x}, \mathbf{y})$ . Indican que los puntos  $\mathbf{x}$  e  $\mathbf{y}$  deben pertenecer a un mismo cluster.
- **Restricciones *Cannot-Link* (CL)**, denotadas  $c_\neq(\mathbf{x}, \mathbf{y})$ . Indican que los puntos  $\mathbf{x}$  e  $\mathbf{y}$  no pueden pertenecer a un mismo cluster.

En particular, las restricciones ML resultan ser una relación de equivalencia entre puntos, por lo que cumplen las propiedades reflexiva, simétrica y transitiva. Esta última es especialmente relevante, ya que permite generar más restricciones ML a partir de las que ya hubiese disponibles:

**Observación 3.1 Las Restricciones *Must-Link* Son Transitivas.** Sean  $CC_i$  y  $CC_j$  componentes conexas, es decir, subgrafos totalmente conectados con restricciones ML. Sean  $x$  e  $y$  dos puntos o instancias en  $CC_i$  y  $CC_j$  respectivamente. Si  $c_=(x, y)$  tal que  $x \in CC_i$ ,  $y \in CC_j$ , entonces  $c_=(a, b) \forall a, b : a \in CC_i, b \in CC_j$  [21] [23].

Análogamente, aún sin constituir una relación de equivalencia, también pueden inferirse nuevas restricciones CL a partir de otras:

**Observación 3.2 Las Restricciones *Cannot-Link* Pueden Conllevar Más Restricciones *Cannot-Link*.** Sean  $CC_i$  y  $CC_j$  componentes conexas, es decir, subgrafos totalmente conectados con restricciones ML. Sean  $x$  e  $y$  dos puntos o instancias en  $CC_i$  y  $CC_j$  respectivamente. Si  $c_\neq(x, y)$  donde  $x \in CC_i, y \in CC_j$ , entonces  $c_\neq(a, b) \forall a, b : a \in CC_i, b \in CC_j$  [21] [23].

Como puede verse, las restricciones ML y CL albergan mucho potencial a pesar de su simpleza. Con ellas es posible encadenar nuevas restricciones, permitiendo aumentar la colección de restricciones de la que se dispone inicialmente.

Además de estas dos propiedades, también existen otras formas de generar nuevas restricciones ML y CL basadas en la distancia:

- **Restricciones Delta ( $\delta$ )**. Para cada punto  $\mathbf{x}$ , se establece una restricción Must-Link con cada punto  $\mathbf{y}$  cuya distancia con  $\mathbf{x}$  sea menor que un cierto valor  $\delta$ . Con esto, es posible garantizar que los puntos pertenecientes a distintos clusters estén separados por una distancia mayor o igual a  $\delta$ .

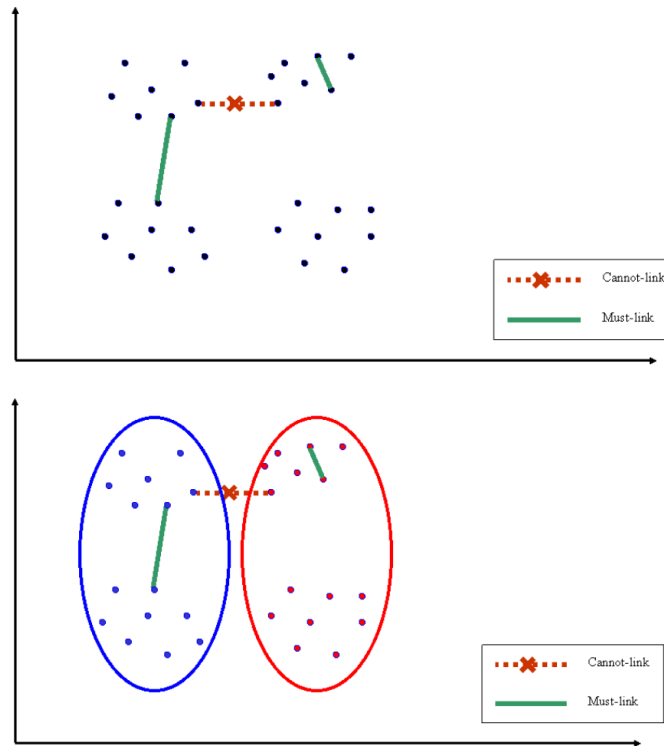


Figura 3.1: Arriba, un pequeño conjunto de datos con restricciones ML y CL. Abajo, una solución de clustering para ese conjunto de datos que satisface a todas las restricciones. [21]

- **Restricciones Épsilon ( $\epsilon$ ).** Para cada punto  $x$ , se establece una restricción Must-Link con al menos un punto  $y$  cuya distancia con  $x$  sea menor o igual a un cierto valor  $\epsilon$ . Esto permite forzar a que todos los puntos tengan un vecindario de diámetro  $\epsilon$ .

Los métodos que hacen uso de este tipo restricciones que afectan a parejas de instancias (Must-Link, Cannot-Link) se conocen como **métodos basados en restricciones**. Éstos métodos modifican al algoritmo de clustering en sí para que las restricciones establecidas introduzcan un sesgo en el proceso de búsqueda para poder encontrar la solución correcta. En esencia, existen dos tipos de métodos basados en restricciones:

1. Aquellos que obligan a que se cumplan las restricciones, de forma que se pretende encontrar la mejor solución que satisfaga a todas ellas [24] [25].
2. Aquellos que obligan parcialmente a que se cumplan permitiendo que no todas sean satisfechas, de forma que se busca, por un lado, encontrar la mejor solución de clustering posible, y por otro, maximizar el número de restricciones satisfechas [26] [27] [28] [29].



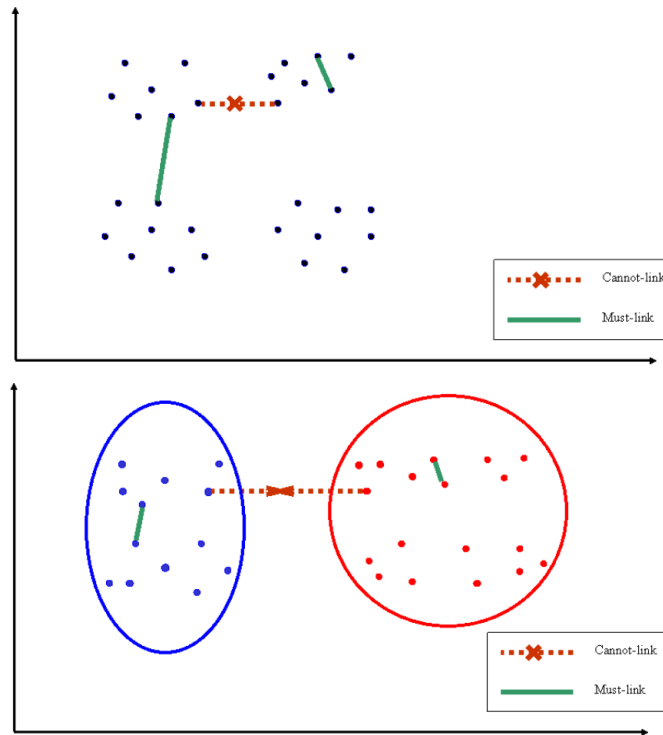


Figura 3.2: Arriba, un pequeño conjunto de datos con restricciones ML y CL. Abajo, una solución de clustering para ese conjunto de datos después de haber modificado las distancias entre las instancias teniendo en cuenta esas restricciones. [21]

La otra gran categoría de técnicas de clustering con restricciones está conformada por los llamados **métodos basados en la distancia**. En ellos, el algoritmo de clustering emplea una medida de distancia *entrenada* o adaptada para que se puedan satisfacer todas las restricciones, en lugar de la distancia euclídea convencional. En este sentido, el espacio es modificado para que se puedan satisfacer las restricciones, de forma que las instancias sujetas a restricciones Must-Link quedan cerca unas de otras, y las restricciones Cannot-Link terminan alejadas entre ellas.

### 3.2.1 Ventajas e inconvenientes del uso de restricciones en clustering

Como todo, el clustering con restricciones tiene una serie de beneficios así como de problemas comparado con el clustering básico. En los dos siguientes apartados veremos de cuáles se tratan.

#### 3.2.1.1 Ventajas

Existen dos beneficios principales de la aplicación de restricciones al problema del clustering. La primera, y quizá la más relevante, es la siguiente:

**Observación 3.3** *Las Restricciones Mejoran la Exactitud en el Caso Promedio.* Al hacer el promedio con muchos conjuntos de restricciones distintos, el rendimiento de la predicción de las etiquetas correctas tenderá a aumentar con respecto a no usar restricciones en absoluto. [21].

Se han llevado a cabo numerosos estudios en los que se ha comprobado que esto se cumple. Por poner un ejemplo, en [22] se logró una mejora del 6 %, 10 % y del 20 % para los conjuntos de datos de la UCI Soybean, Pos y Mushroom respectivamente, con tan solo un conjunto de 40 restricciones a nivel de instancia.

El segundo beneficio del uso de restricciones es la posibilidad de *amoldar* la forma geométrica de los clusters resultantes. En [24], por ejemplo, donde se emplean datos de rutas de coches vía GPS, los clusters que deberían representar las vías no salen con la forma alargada que cabría esperar. Por medio de la implantación de restricciones Cannot-Link entre instancias situadas a más de 4 metros de distancia entre ellas en dirección perpendicular a las carreteras, fue posible corregir los clusters generados para que adoptasen la forma esperada. De esta forma, o con la formación de restricciones  $\delta$  y restricciones  $\epsilon$ , es posible modificar las propiedades geométricas de la forma que mejor convenga.

### 3.2.1.2 Desventajas

A pesar de haberse demostrado los beneficios del uso de restricciones, existen dos importantes limitaciones que se han de tener en cuenta.

El primer problema es la **factibilidad**, es decir, el hecho de que sea factible satisfacer todas las restricciones del conjunto de restricciones dado. Si no se emplea un buen método para generar las restricciones, estas pueden contradecirse las unas a las otras, por lo que puede que no exista ninguna solución de clustering que las satisfaga a todas. Por ejemplo, si tenemos dos objetos  $x_a$  y  $x_b$ , bastaría con definir las restricciones  $c_=(x_a, x_b)$  y  $c_=(x_a, x_b)$  para tener una contradicción. Una definición más formal de este problema es la siguiente:

**Definición 3.1 Problema de la factibilidad:** Dado un conjunto de datos  $X$ , un conjunto de restricciones  $R$ , una cota inferior  $K_l$  y una cota superior  $K_u$ , ¿existe una partición de  $X$  en  $k$  clusters tal que  $K_l \leq k \leq K_u$  y que todas las restricciones en  $R$  sean satisfechas? [21] [28].

El problema de la factibilidad es NP-Completo cuando se emplea una conjunción de restricciones Cannot-Link [21], por lo que el problema de encontrar la mejor solución que satisfaga todas las restricciones puede volverse inviable. Determinar si una única solución satisface todas las restricciones ya es complejo de por sí, pero obtener la mejor solución que las satisfaga a todas lo es aún más.

El otro gran problema del uso de restricciones para el problema del clustering es que **no todos los conjuntos de restricciones son útiles**, aunque sí sean satisfacibles. Normalmente, se asume que las restricciones son *pistas* que ayudan durante el proceso de búsqueda, puesto que introducen información sobre la solución óptima o verdadera y permiten que, en teoría, sea más fácil obtener una solución similar a la verdadera.

Pensar eso no es descabellado teniendo en cuenta la observación 3.3. Sin embargo, hay que tener en cuenta que las restricciones mejoran los resultados en el caso promedio, y no todos los conjuntos de restricciones provocarán una mejora. Por ejemplo, en [30] se demuestra que aún empleando restricciones sin ruido, sin errores y obtenidas directamente de la solución verdadera, es posible que algunos conjuntos de restricciones concretos afecten negativamente al rendimiento. Esto parece ir en contra de lo que suele deducir de otros experimentos, pero el motivo es, de nuevo, que en ellos se promedian los resultados, y en tales casos la observación 3.3 vuelve a cumplirse:

**Observación 3.4 Conjuntos de Restricciones Concretos Pueden Tener Efectos Adversos:** *Algunos conjuntos de restricciones generados a partir de las etiquetas de la solución verdadera, con la que se evalúan los clusters obtenidos, pueden disminuir la exactitud de la predicción de esas mismas etiquetas [21].*

### 3.2.2 Ejemplos de uso

En [21] se enumeran algunos casos reales de aplicación del clustering con restricciones:

- **Imágenes:** clustering de píxeles para identificación de objetos en la navegación del robot Aibo, mediante el uso de restricciones espaciales (restricciones  $\epsilon$  y  $\delta$ , por ejemplo) [cita]
- **Vídeos:** inclusión de restricciones ML en grupos de píxeles que representan el mismo objeto a lo largo de sucesivos fotogramas [Yan et al. 2004]
- **Biología:** clustering de genes, con restricciones ML para los genes y proteínas que participan en los mismos procesos celulares [Xenarios et al. 2001] [Segal et al. 2003].
- **Audio:** análisis de audio, empleando restricciones para distinguir si dos locutores corresponden a la misma persona o si son diferentes [Bar-Hillel et al. 2003].
- **GPS:** detección de vías y carreteras, empleando restricciones ML en posiciones por las que ha pasado el mismo coche por una carretera, y restricciones CL entre coches situados a más de

4 metros de distancia entre ellas en dirección perpendicular a la de la circulación [24].

## LOS ALGORITMOS EVOLUTIVOS MULTI OBJETIVO

---

El tipo de algoritmos al que pertenecen aquellos que serán estudiados e implementados en este trabajo son los llamados algoritmos evolutivos multiobjetivo. Para comprenderlos bien, primero será necesario estudiar los dos conceptos clave en los que se basan: la optimización multiobjetivo y los algoritmos evolutivos.

### 4.1 OPTIMIZACIÓN MULTI OBJETIVO

Antes de comenzar a definir la optimización multiobjetivo, para situar al lector sería lógico hablar brevemente sobre la optimización clásica, es decir, aquella cuyo objetivo consiste en optimizar una única función. Eso es justamente lo que se hará a continuación.

#### 4.1.1 Problema de optimización de un objetivo

En los problemas de optimización convencionales, el objetivo consiste en encontrar una solución, dentro del espacio de soluciones posibles, que optimice una única función objetivo. Ya se vio un ejemplo en la sección 2.3.4, donde se formula sin mucho rigor el problema del clustering como un problema de optimización. Una definición general y más formal de los problemas de optimización sería la siguiente:

**Definición 4.1 Problema General de Optimización de un Único Objetivo:** *Un problema de optimización de un único objetivo consiste en minimizar (o maximizar) una función  $f(\mathbf{x})$  sujeta a  $g_i(\mathbf{x}) \leq 0$ ,  $i = \{1, \dots, r\}$  y  $h_j(\mathbf{x})$ ,  $j = \{1, \dots, p\}$ . Una solución  $\mathbf{x}$  minimiza (o maximiza) el escalar  $f(\mathbf{x})$ , donde  $\mathbf{x} \in \Omega$  es un vector  $n$ -dimensional de variables de decisión  $\mathbf{x} = (x_1, \dots, x_n)$  de un universo  $\Omega$  [31].*

Como se puede ver,  $g_i(\mathbf{x}) \leq 0$  y  $h_j(\mathbf{x}) = 0$  representan restricciones que deben cumplirse al optimizar  $f(\mathbf{x})$ . Por su parte el universo  $\Omega$  es el conjunto que contiene todas las posibles soluciones  $\mathbf{x}$  que pueden usarse para tanto la función objetivo  $f(\mathbf{x})$  como las restricciones.

El método por el cual se encuentra el óptimo global de una función cualquiera se llama *optimización global*:

**Definición 4.2 Optimización del Mínimo Global:** *Dada una función  $f : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $\Omega \neq \emptyset$ , para  $\mathbf{x} \in \Omega$  el valor  $f^* \equiv f(\mathbf{x}^*)$  es el mínimo global si y solo si*

$$\forall \mathbf{x} \in \Omega : f(\mathbf{x}^*) \leq f(\mathbf{x}), \quad (4.1)$$

donde  $\mathbf{x}^*$  es la solución mínima global,  $f$  es la función objetivo, y el conjunto  $\Omega$  es la región factible de  $\mathbf{x}$ . El objetivo de encontrar la solución o soluciones mínimas globales se llama **problema de optimización global** [31].

#### 4.1.2 Problema de optimización multiobjetivo

En un principio, puede parecer que la única novedad existente en la optimización multiobjetivo con respecto a la optimización convencional consiste, como su nombre indica, en la optimización de varias una funciones objetivo. Sin embargo, esta diferencia tiene implicaciones que cambian notablemente la forma en la que se trabaja en los problemas de optimización.

La optimización multiobjetivo en general puede definirse de la siguiente forma:

**Definición 4.3 Problema General de Optimización Multiobjetivo :** Un problema de optimización multiobjetivo consiste en minimizar (o maximizar)  $F(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$  sujeto a  $g_i(\mathbf{x}) \leq 0, i = \{1, \dots, r\}$  y  $h_j(\mathbf{x}), j = \{1, \dots, p\}$ . Un problema de optimización minimiza (o maximiza) los componentes de un vector  $F(\mathbf{x}) \in \mathbb{R}^m$  donde  $\mathbf{x} \in \Omega$  es un vector de  $n$  variables de decisión  $\mathbf{x} = (x_1, \dots, x_n)$  de un universo  $\Omega$  [31].

De nuevo,  $g_i(\mathbf{x}) \leq 0$  y  $h_j(\mathbf{x}) = 0$  representan las restricciones, y  $\Omega$  representa el espacio de soluciones o región factible.

Al optimizar más de una función objetivo, surge un nuevo problema que no se daba cuando se trabaja con una única función objetivo: que no existe una definición consensuada de qué se considera una solución óptima, por lo que resulta difícil comparar unas soluciones con otras [32].

Al ser las funciones objetivo independientes entre ellas, existiendo en ocasiones incluso contradicciones entre ellas, no es posible encontrar siempre una única solución que sea óptima para todas las funciones objetivo a la vez, y por tanto no existe un orden total en el espacio objetivo, sino tan solo un orden parcial [33]. Por ejemplo, para un espacio objetivo en  $\mathbb{R}^2$ , podríamos decir que el punto (1,2) es menor que el punto (3,6). Sin embargo, los puntos (1,3) y (2,1) no se podrían comparar tan fácilmente.

Es necesario establecer una forma de determinar qué soluciones son, de alguna forma, mejores que otras. Para ello, introduciremos la terminología de Pareto, que será la que nos permita comparar vectores de valores objetivo  $F(\mathbf{x})$  y determinar qué soluciones  $\mathbf{x}$  obtenidas pueden ser útiles, interesantes o válidas para el problema a resolver.

El primer concepto a definir es el de *dominancia de Pareto* o *Pareto-dominancia*, que nos permitirá definir una relación de orden (parcial) entre vectores de valores objetivo:

**Definición 4.4 Pareto-dominancia:** un vector  $\mathbf{u}$  domina al vector  $\mathbf{v}$  (denotado  $\mathbf{u} \preceq \mathbf{v}$ ) si, en caso de estar ante un problema de minimización,  $\mathbf{u}$  es parcialmente menor que  $\mathbf{v}$  [13][31]:

$$\mathbf{u} \preceq \mathbf{v} \iff \forall i \in \{1, \dots, m\} : u_i \leq v_i \wedge \exists j \in \{1, \dots, m\} : u_j < v_j. \quad (4.2)$$

Dicho en un lenguaje más natural: el vector  $\mathbf{u}$  domina al vector  $\mathbf{v}$  si todos los elementos de  $\mathbf{u}$  son menores o iguales a los respectivos elementos de  $\mathbf{v}$ , y al menos uno de los elementos de  $\mathbf{u}$  es estrictamente menor que el elemento correspondiente de  $\mathbf{v}$ . Nótese que esta definición asume que el problema es de minimización. Para los problemas de maximización, bastaría con cambiar los menores o iguales y los menores estrictos, por mayores o iguales y mayores estrictos, respectivamente.

A partir de la dominancia de Pareto, se definen los conceptos de *optimalidad de Pareto* y *conjunto de Pareto*:

**Definición 4.5 Optimalidad de Pareto :** Una solución  $\mathbf{x} \in \Omega$  se dice que es Pareto-óptima con respecto a  $\Omega$  si y solo si no existe ninguna otra solución  $\mathbf{x}'$  tal que  $F(\mathbf{x}') \preceq F(\mathbf{x})$  [31] [33].

**Definición 4.6 Conjunto Óptimo de Pareto:** : Dadas unas funciones objetivo  $F(\mathbf{x})$ , el Conjunto Óptimo de Pareto  $\mathcal{P}^*$  se compone de todas las soluciones Pareto-óptimas, es decir, todas las soluciones no dominadas del espacio de soluciones [31]:

$$\mathcal{P}^* := \{\mathbf{x} \in \Omega \mid \neg \exists \mathbf{x}' \in \Omega : F(\mathbf{x}') \preceq F(\mathbf{x})\}. \quad (4.3)$$

El conjunto óptimo de Pareto está formado, en teoría, por las mejores soluciones posibles. Nótese que es todo un *conjunto* de soluciones, y no una única solución óptima. Bien es cierto que, en optimización de un objetivo, pueden obtenerse más de una solución óptima para un problema determinado, pero eso significaría que sus correspondientes valores objetivos  $f(\mathbf{x})$  serían el mismo. En cambio, en el ámbito de la optimización multiobjetivo se da la particularidad de que se obtienen diversas soluciones óptimas (o en este caso, no dominadas) cuyos respectivos vectores de valores objetivo  $F(\mathbf{x})$  pueden ser distintos entre ellos, siempre y cuando ninguno de esos vectores domine a otro. Esto pone de manifiesto la complejidad adicional que supone trabajar con más de una función objetivo.

El último concepto relativo a la optimización multiobjetivo que introduciremos es el de *frente de Pareto*:

**Definición 4.7 Frente de Pareto** [31]: dados unas funciones objetivo  $F(\mathbf{x})$  y un Conjunto Óptimo de Pareto  $\mathcal{P}^*$ , el Frente de Pareto  $\mathcal{PF}^*$  se define como:

$$\mathcal{PF}^* := \{F(\mathbf{x}) \mid \mathbf{x} \in \mathcal{P}^*\}. \quad (4.4)$$

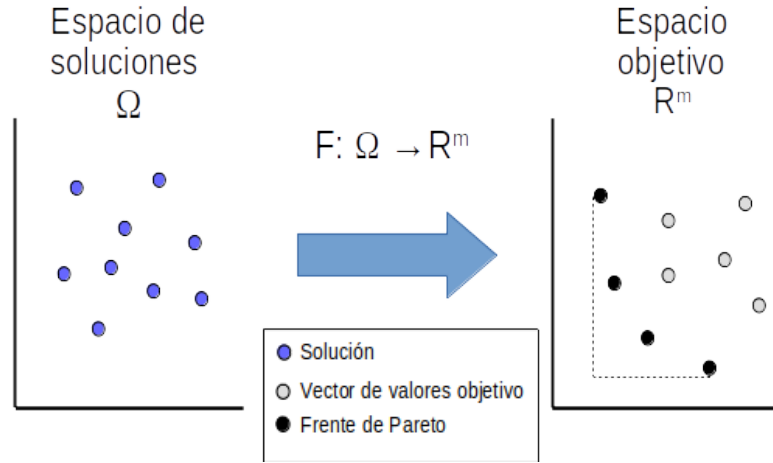


Figura 4.1: Representación gráfica de un frente de Pareto

El frente de Pareto de un conjunto óptimo  $\mathcal{P}^*$  no es sino la imagen del conjunto  $\mathcal{P}^*$  en el espacio de valores objetivo, o lo que es lo mismo, es el conjunto de correspondientes vectores objetivo de las soluciones que componen  $\mathcal{P}^*$ .

¿Cómo puede obtenerse el conjunto óptimo  $\mathcal{P}^*$ ? Un posible método consistiría en generar todas las posibles soluciones de  $\Omega$ , obtener los vectores de valores objetivos de estas y comprobar cuáles de ellas no son dominadas. Este método, no obstante, no es viable debido al alto número de soluciones del que normalmente se compone  $\Omega$  (de hecho, incluso puede tratarse de un conjunto infinito). Como tampoco es fácil, por no decir imposible, obtener una expresión analítica para el frente de Pareto, el procedimiento normal para generarlo consiste en obtener un subconjunto de soluciones de  $\Omega$  lo suficientemente grande, para posteriormente obtener sus vectores de valores objetivo y comprobar cuáles de ellos no son dominados [31].

¿Por qué se le llama *frente* al frente de Pareto? Si se dibujan en una gráfica los vectores de valores objetivo  $F(\mathbf{x})$  de las soluciones  $\mathbf{x}$  obtenidas, se puede observar que aquellos puntos  $F(\mathbf{x})$  correspondientes a soluciones no dominadas describen una curva que se sitúa en el límite o frontera del conjunto de puntos obtenidos, y es precisamente en esa frontera donde, en teoría, se encuentran todas las soluciones óptimas, a pesar de que a priori solo se haya generado un subconjunto de ellas. La figura 4.1 ilustra el concepto de frente de Pareto por medio de un sencillo ejemplo.

#### 4.2 ALGORITMOS EVOLUTIVOS

Antes de hablar sobre los algoritmos evolutivos multiobjetivo, es lógico dedicar un espacio a los algoritmos evolutivos en general, así como a la clase de algoritmos a los que pertenecen: las metaheurísticas.



#### 4.2.1 Metaheurísticas

Las *metaheurísticas* son una familia de algoritmos de optimización aproximada, muy útiles para resolver problemas de gran dificultad y complejidad puesto que ofrecen soluciones *aceptables* o que pueden considerarse suficientemente buenas en un tiempo razonable. A diferencia de otros algoritmos de optimización, las metaheurísticas no garantizan la optimalidad de las soluciones encontradas [34].

Las técnicas metaheurísticas pueden dividirse entre dos grupos principales [13] [34]:

- **Metaheurísticas de una única solución.** En este tipo de metaheurísticas, una única solución se va mejorando de acuerdo a una función objetivo, de forma que esta solución representa, en muchos casos, un punto que va siguiendo una trayectoria en el espacio de soluciones. Ejemplos de esta clase de algoritmos son el enfriamiento simulado (*Simulated Annealing*) [35] y la búsqueda tabú (*Tabu Search*) [36].
- **Metaheurísticas basadas en poblaciones.** Estos algoritmos, como su propio nombre indica, trabajan con toda una población de soluciones. Dicha población es inicializada aleatoriamente, e iterativamente se van generando nuevas poblaciones que son de alguna forma integradas en la población principal usando un cierto criterio de selección. La ventaja de trabajar con toda una población de soluciones en vez de con solo una de ellas es que permite diversificar el proceso de búsqueda en el espacio de soluciones. Ejemplos de metaheurísticas basadas en poblaciones son los algoritmos genéticos o GA (*Genetic Algorithms*) [37], la evolución diferencial o DE (*Differential Evolution*) [38], la optimización por nube/enjambre de partículas o PSO (*Particle Swarm Optimization*) [39] y la optimización por colonia de hormigas o ACO (*Ant Colony Optimization*) [40].

A su vez, dentro de las metaheurísticas basadas en poblaciones se encuentran los llamados *algoritmos evolutivos*, al que pertenecen, entre otros, los algoritmos genéticos y las técnicas de evolución diferencial.

#### 4.2.2 Conceptos básicos de los algoritmos evolutivos

Los algoritmos evolutivos o EAs (*Evolutionary Algorithms*) tienen en común el hecho de que se basan en la evolución biológica de las especies, de forma que a los individuos de la población se les aplican operadores evolutivos para generar nuevos individuos (llamados *descendientes*) a partir de ellos.

Un *individuo* representa una solución codificada al problema que se pretende optimizar, y suele estar representado por cadenas o secuencias de valores, que corresponden al *genotipo* en términos biológicos.

El genotipo de un individuo se compone de un cromosoma (o más), y a su vez un *cromosoma* se compone de genes que pueden adoptar determinados valores (*alelos*). De esta forma, cada individuo codifica una serie de parámetros usados como valores de entrada de la función que se desea optimizar. Por último, un conjunto de cromosomas se denomina *población*. [31]

Con el fin de generar soluciones cada vez mejores, sobre la población se aplican una serie de operadores inspirados en la biología y la selección natural, llamados *operadores evolutivos*. Los tres operadores usados en los algoritmos evolutivos son: el operador de *cruce* o *recombinación*, el operador de *mutación* y el operador de *selección* [31]. A grandes rasgos, el operador de cruce permite generar una o varias soluciones nuevas (descendientes) a partir de una pareja de soluciones de la población (padres). El operador de mutación, por su parte, permite introducir pequeñas alteraciones en las soluciones para, por ejemplo, hacerlas ligeramente diferentes de sus soluciones padres. Finalmente, el operador de selección se utiliza para dar, en general, mayor probabilidad de reproducirse (es decir, de usar el operador de cruce) a los individuos mejores, mientras que análogamente se da menor probabilidad a los peores individuos.

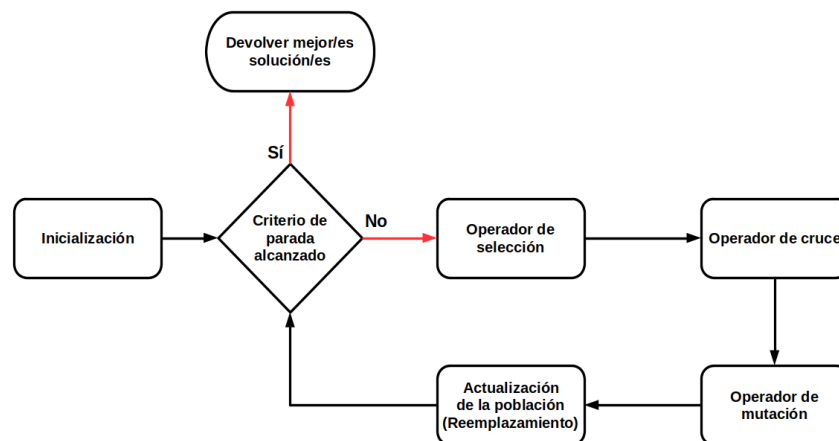


Figura 4.2: Diagrama de flujo de un algoritmo genético, un tipo de algoritmo evolutivo.

Como ya se sabe, al ser los algoritmos evolutivos un tipo de algoritmos de optimización, estos trabajan con una función objetivo que se pretende optimizar, también llamada en estos casos función de *fitness* [31]. La función de *fitness* asignan un valor numérico a las soluciones de conforman la población indicando cómo de buena es cada una de ellas, y es en este valor de bondad en el que se basa el operador de selección para asignar a cada solución su probabilidad de reproducirse.

#### 4.2.3 Algoritmos evolutivos para clustering y clustering con restricciones

Como ya se vió en la sección 2.3.4, el clustering puede formularse como un problema de optimización, y como tal puede ser resuelto con algoritmos evolutivos. Para ello es necesario establecer dos elementos fundamentales: una función objetivo y un esquema de representación o codificación (es decir, la forma en que se codifican los cromosomas que corresponden a las distintas soluciones de la población).

En el caso del clustering clásico, cualquiera de los índices de validación expuestos en la sección 2.3.3 puede usarse como función objetivo, mientras que para el clustering con restricciones podría (o debería) usarse un índice o función que tenga en cuenta de alguna forma las restricciones que han sido satisfechas o no. Lo que sí pueden usarse de la misma forma para ambos problemas son los esquemas de representación. Los esquemas de representación para el clustering (y el clustering con restricciones) pueden agruparse en tres clases [13]:

- **Codificación binaria:** Una solución se representa con una secuencia de  $N$  dígitos binarios, donde  $N = |\mathbf{X}|$  es el número de objetos en el conjunto de datos, de forma que cada posición corresponde a un objeto del conjunto de datos. El valor de la  $i$ -ésima posición será igual a 1 si el  $i$ -ésimo objeto del conjunto de datos corresponde a un centroide o prototipo de cluster, mientras que será igual a 0 en caso contrario.
- **Codificación entera:** De forma similar a la codificación binaria, las soluciones se representan con secuencias de  $N$  números enteros que codifican todos los objetos del dataset. A su vez, existen dos tipos de esquemas de representación con números enteros:
  - **Representación basada en etiquetas:** En cada posición se guarda un valor entero del conjunto  $\{1, \dots, K_{max}\}$  (o del conjunto  $\{0, \dots, K_{max} - 1\}$  si se empieza desde el 0), donde  $K_{max}$  es el número máximo de clusters que pueden formarse. De esta forma, si en la posición  $i$  del cromosoma aparece el valor  $k$ , significa que el  $i$ -ésimo objeto del conjunto de datos  $x_i$  pertenece o ha sido asignado al cluster  $c_k$ .
  - **Representación basada en grafos:** En este esquema el conjunto de datos es tratado como un grafo dirigido. Cada posición de la secuencia puede almacenar valores enteros del conjunto  $\{1, \dots, N\}$  (o  $\{0, \dots, N - 1\}$ ), de forma que si en la posición  $i$  aparece el valor entero  $j$ , esto quiere decir que el objeto  $x_i$  está conectado al objeto  $x_j$ . De esta forma, la solución de clustering se construye identificando las distintas partes conexas del grafo, de forma que cada parte conexa del grafo corresponde a un cluster.

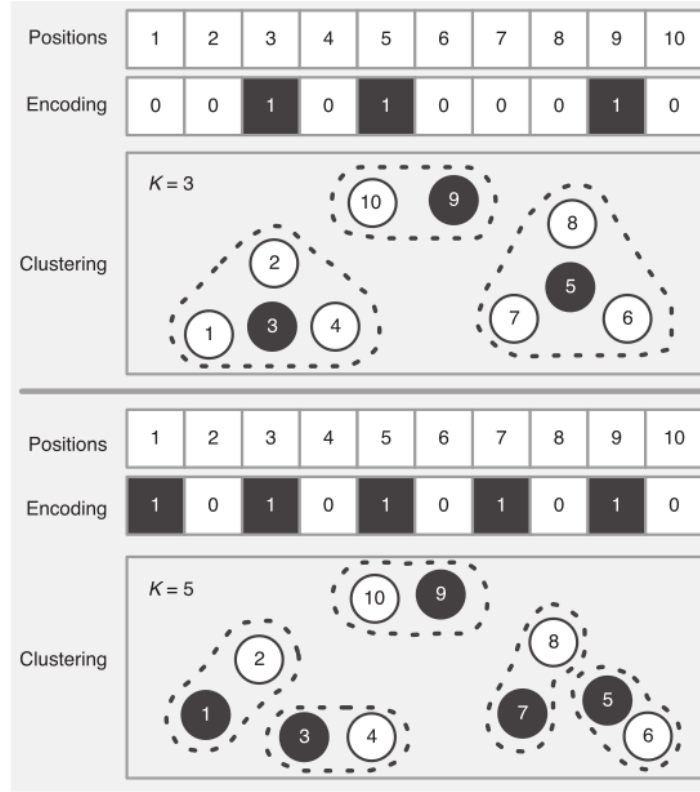


Figura 4.3: Codificación binaria para  $K = 3$  y  $K = 5$  clusters [13].

- Codificación real:** En este tipo de codificación, los cromosomas representan las localizaciones o coordenadas de los centroides de los clusters en el espacio de características de  $D$  dimensiones. Si en el problema a resolver existen  $N$  objetos en un espacio  $D$ -dimensional, los cuales deben ser agrupados en  $K$  clusters, entonces los centroides de una solución determinada pueden codificarse con matrices de  $D \times K$  valores reales. La codificación real puede ser de tamaño fijo o tamaño variable:
  - Codificación de longitud fija:** Las soluciones se codifican asumiendo un predeterminado número máximo de clusters  $K_{max}$ , de forma que todos los cromosomas de la población mantienen la misma longitud de  $D \times K_{max}$ . Junto a los cromosomas suelen usarse vectores con umbrales de activación para determinar cuáles de esos  $K_{max}$  clusters están participando realmente en el agrupamiento.
  - Codificación de longitud variable:** Si se tiene un conjunto de  $P$  soluciones  $\mathcal{C} = \{\mathbf{C}_i \mid i = 1, \dots, P\}$ , la  $i$ -ésima solución codifica un cierto número de centroides  $K_i$ . Por tanto, la longitud del cromosoma será variable dependiendo de la solución, teniendo un tamaño de  $DK_i$ , aspecto que debe ser tenido en cuenta al hacer uso de los operadores evolutivos.

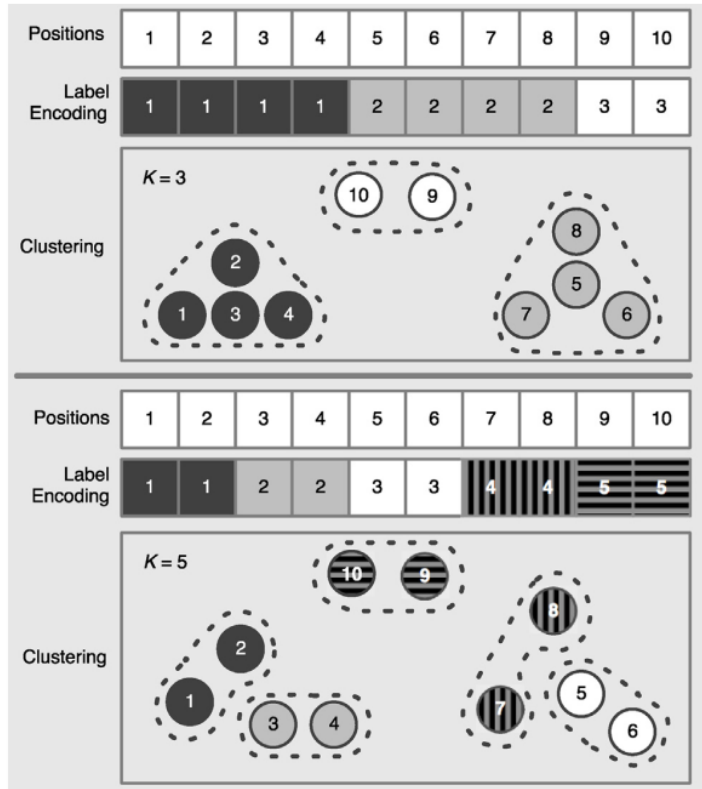


Figura 4.4: Codificación entera basada en etiquetas [13].

Existen diversos ejemplos del uso de algoritmos evolutivos para el problema del clustering y del clustering con restricciones. En [41], por ejemplo, se realiza un análisis sobre el uso de técnicas de clustering basadas en algoritmos genéticos. Por otra parte, en [42] se propone un algoritmo evolutivo con búsqueda local, al que llaman DILS (*Dual Iterative Local Search*), en cuya función de *fitness* se suma, por un lado, la sumatoria de distancias al cuadrado entre elementos del mismo cluster, y por otro, un factor de penalización que aumenta cuantas más restricciones son violadas (o sea, no satisfechas). Se trata, como es lógico, de una función de minimización, puesto que una solución es mejor cuanto más cercanos sean los elementos de un mismo cluster y cuantas más restricciones satisfaga.

En definitiva, el uso de algoritmos evolutivos en problemas de clustering, con o sin restricciones, es idóneo dados, por un lado, el hecho de que encontrar la partición óptima en un problema del clustering es de una alta complejidad, y por otro, el hecho de que los algoritmos evolutivos ofrezcan soluciones lo suficientemente buenas en un tiempo aceptable.

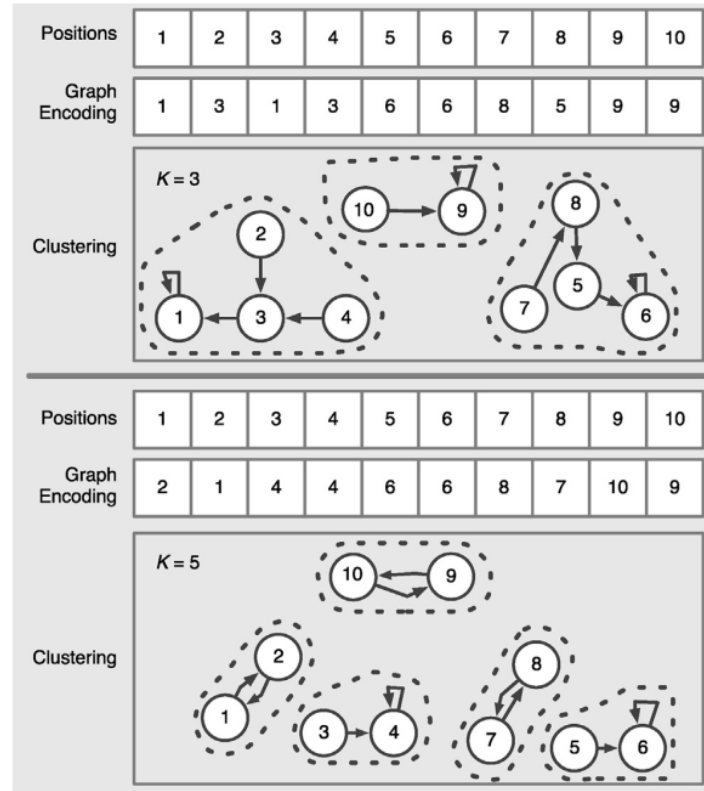


Figura 4.5: Codificación entera basada en grafos [13].

#### 4.3 ALGORITMOS EVOLUTIVOS MULTIOBJETIVO

Los algoritmos evolutivos multiobjetivo o MOEAs (*Multi-Objective Evolutionary Algorithms*) son especialmente adecuados para los problemas de optimización multiobjetivo, ya que al ser algoritmos basados en poblaciones están diseñados para manejar múltiples soluciones durante el proceso de búsqueda [32].

Una característica relativa a la terminología de Pareto que distingue a los MOEAs de los EAs es el uso de poblaciones adicionales de individuos que no son Pareto-dominados. Muchos MOEAs manejan una población secundaria de soluciones, que al principio está vacía, en la que se van almacenando todas las soluciones no dominadas que se han encontrado hasta el momento. Como la Pareto-dominancia de un vector depende del contexto en el que es evaluado, cada vez que se añade una solución o conjunto de soluciones no dominadas a esa población, las soluciones ya incluidas deben ser comprobadas para asegurarse de que siguen siendo no dominadas. De lo contrario, dichas soluciones son eliminadas de la población secundaria [31].

Tipos de MOEAs

- A Priori
- Progresivos

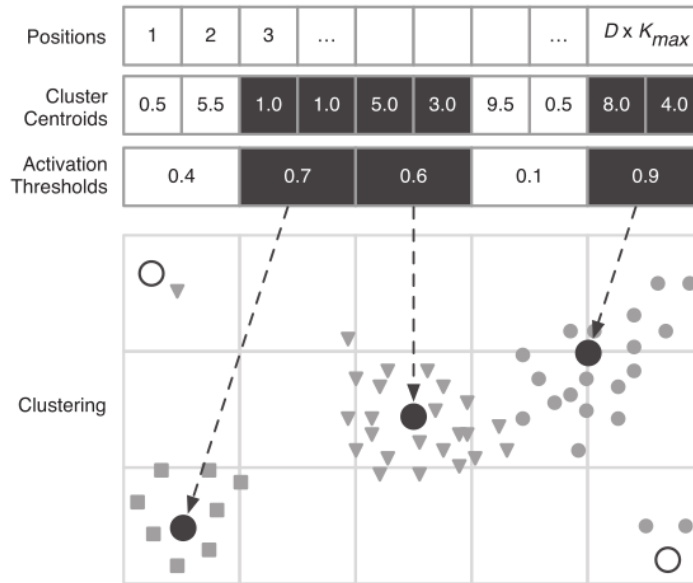


Figura 4.6: Codificación real [13].

#### ■ Posteriori

(Ejemplos de MOEAs)

Uso de MOEAs para clustering (¿clustering con restricciones?)

La ventaja de usar MOEAs para resolver problemas de clustering clásico, al ser multiobjetivos, permiten usar múltiples índices de validación. Pueden usarse al mismo tiempo tanto índices que prioricen la cohesión intra-cluster (p.ej.: minimizar distancias entre las instancias y los centroides de los clusters a los que pertenecen, minimizar distancias entre instancias del mismo cluster, etc.) como índices que se enfoquen en la distancia entre clusters (p.ej.: maximizar distancias entre centroides de los clusters, maximizar distancias entre instancias de clusters distintos), sin que se establezca un orden de prioridad entre ellas.

Además, los MOEAs también son especialmente idóneos para ser usados con problemas de clustering con restricciones, ya que permiten usar como funciones distintas e independientes, por un lado, los índices de validación para clustering clásico para evaluar la calidad de la agrupación obtenida, y por otro, alguna función de insatisfacción para controlar en qué grado se respetan las restricciones impuestas.





## ALGORITMOS EVOLUTIVOS MULTIOBJETIVO BASADOS EN DESCOMPOSICIÓN

---

En este capítulo, será introducida una nueva clase de algoritmos evolutivos que afrontan la optimización objetivo de una forma diferente a los demás: los algoritmos basados en descomposición. Comenzaremos con uno de los más conocidos: el MOEA/D.

### 5.1 MULTIOBJECTIVE EVOLUTIONARY ALGORITHM BASED ON DECOMPOSITION

*Multiobjective Evolutionary Algorithm based on Decomposition*, o por sus siglas MOEA/D, [43] es un algoritmo evolutivo que emplea el concepto de descomposición para resolver problemas de optimización multiobjetivo. La *descomposición* consiste en asociar cada solución de un conjunto óptimo de Pareto a un problema de optimización escalar<sup>1</sup>, de forma que un frente de Pareto pueda ser *descompuesto* en un cierto número de subproblemas de optimización escalar. Para ello, pueden usarse diferentes funciones de agregación que transformen cada vector de valores objetivo en un valor escalar, las cuales se expondrán con detalle más adelante.

La idea de usar descomposición en un MOEA viene dada por el hecho de que la dominancia de Pareto, como ya se vio previamente en la sección 4.1.2, no define un orden total entre las soluciones en el espacio objetivo, por lo que resulta más difícil en tal caso seleccionar (o descartar) las soluciones que se generan a lo largo del proceso de búsqueda.

#### 5.1.1 Funcionamiento general

MOEA/D necesita descomponer el problema de optimización multiobjetivo que se pretende resolver, y para ello necesita una función de agregación  $g(\mathbf{x})$  que tome como entrada una solución de nuestro problema (o sea, un vector  $\mathbf{x}$ ) y devuelva como salida un valor escalar. De esta forma se puede convertir o *descomponer* el problema de optimización multiobjetivo en un cierto número de subproblemas de optimización escalar.

Para tal fin, la función de agregación necesita de vectores de pesos. Cada uno de ellos consiste en un vector de valores no negativos  $\lambda = (\lambda_1, \dots, \lambda_m)^T$  tal que  $\sum_{i=1}^m \lambda_i = 1$ , de modo que un subproblema

<sup>1</sup> Por «problema de optimización escalar», entiéndase «problema de optimización con una única función objetivo»

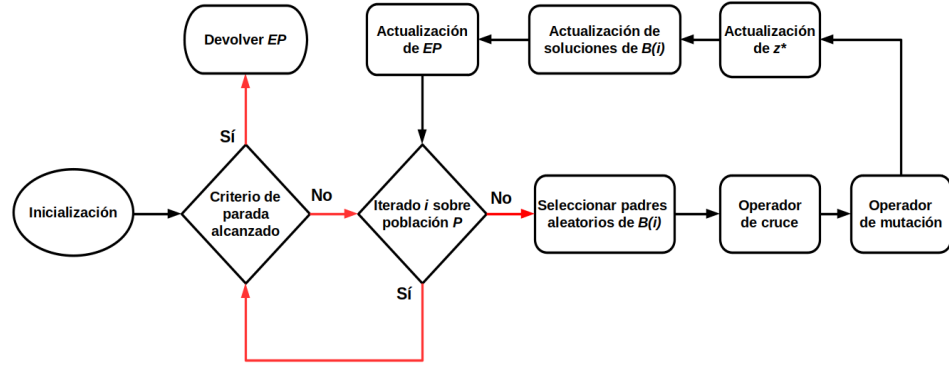


Figura 5.1: Diagrama de flujo del MOEA/D

de optimización escalar consistiría en encontrar la solución  $\mathbf{x}$  que optimice la función de agregación  $g(\mathbf{x}|\lambda)$  para un  $\lambda$  dado. De esta forma, por cada vector de pesos, existirá un subproblema de optimización escalar, puesto que para cada vector de pesos  $\lambda$  habrá que encontrar la solución  $\mathbf{x}^*$  que lo optimice. La notación  $g(\mathbf{x}|\lambda)$  hace énfasis en el hecho de que  $\lambda$  es un vector de coeficientes, mientras que  $\mathbf{x}$  es el vector con las variables que se han de optimizar.

En otras palabras, podría decirse que cada  $\lambda$  representa un subproblema, y que por tanto habrá tantos subproblemas escalares como vectores de pesos. Además, lo ideal es que todos ellos estén lo más uniformemente distribuidos para que todo el frente de Pareto quede bien representado al descomponerse.

En MOEA/D se asume que la función de agregación  $g(\mathbf{x}|\lambda)$  es continua, y que si dos vectores de pesos  $\lambda^i$  y  $\lambda^j$  son cercanos, sus respectivas soluciones óptimas de  $g(\mathbf{x}|\lambda^i)$  y  $g(\mathbf{x}|\lambda^j)$  también serán cercanas. Este hecho es importante, porque ayuda a encontrar el  $\mathbf{x}^*$  óptimo de un determinado vector  $\lambda^i$  a partir de los  $\mathbf{x}^*$  óptimos de sus vectores más cercanos. Para sacar provecho de ello, se define para cada  $\lambda^i$  un vecindario  $B(i) = \{i_1, \dots, i_T\}$ , de forma que el conjunto conformado por los vectores  $\{\lambda^{i_1}, \dots, \lambda^{i_T}\}$  representa los  $T$  vectores de pesos más cercanos a  $\lambda^i$ . De esta forma, como la población se compone de la mejor solución encontrada hasta el momento para cada subproblema, solo las soluciones actuales son aprovechadas para optimizar cada subproblema escalar usando estos vectores de pesos.

Por último, cabe destacar que, como la población se compone de la mejor solución encontrada hasta el momento para cada subproblema, tendremos tantos individuos  $\mathbf{x}$  como subproblemas. Dicho de otra forma: a cada  $\mathbf{x}$  de la población le corresponde un vector de pesos  $\lambda$  que lo optimice.

### 5.1.2 Métodos de descomposición

Las siguientes son algunos métodos de descomposición que pueden emplearse con MOEA/D para generar, por medio de diferentes funciones de agregación, problemas de optimización escalares:

#### 1. Método de Suma Ponderada (*Weighted Sum*) .

Sea  $\lambda = \{\lambda_1, \dots, \lambda_m\}$  un vector de pesos no negativos tal que  $\sum_{i=1}^m \lambda_i = 1$ . La solución óptima para el siguiente problema de optimización:

$$\max_{\mathbf{x} \in \Omega} g^{ws}(\mathbf{x}|\lambda) = \sum_{i=1}^m \lambda_i f_i(\mathbf{x}) \quad (5.1)$$

es una solución Pareto-óptima del problema de optimización multiobjetivo a resolver. El conjunto de vectores  $\lambda$  que se utilicen determinará el conjunto de Pareto que se obtenga. Este sencillo enfoque puede resultar útil si el frente de Pareto es convexo (o cóncavo si el problema es de maximización) [33].

#### 2. Método de Tchebycheff.

En este enfoque el problema escalar de optimización es el siguiente:

$$\min_{\mathbf{x} \in \Omega} g^{te}(\mathbf{x}|\lambda, \mathbf{z}^*) = \max_{1 \leq i \leq m} \{\lambda_i |f_i(\mathbf{x}) - z_i^*|\} \quad (5.2)$$

donde  $\mathbf{z}^* = (z_1^*, \dots, z_m^*)$  es el punto de referencia o punto ideal, de forma que si nuestro problema es de minimización, entonces  $z_i^* = \min_{\mathbf{x} \in \Omega} f_i(\mathbf{x})$ . Nótese que se asume que todas las funciones objetivo están acotadas. En caso de no conocer cuál es el punto, puede usarse como aproximación el mínimo de los  $f_i(\mathbf{x})$  encontrados hasta el momento en el proceso de búsqueda.

Puede demostrarse que el problema de optimización descrito en (5.2) siempre tiene al menos una solución óptima, y si esa solución es única, entonces será Pareto-óptima. También se cumple que, para cualquier solución Pareto-óptima  $\mathbf{x}^*$ , siempre existirá un vector de pesos mayores que 0 para el cual  $\mathbf{x}^*$  será una solución óptima de (5.2) [33].

#### 3. Método de Intersección de Frontera (*Boundary Intersection*).

Normalmente, el frente de Pareto de un problema de optimización objetivo se sitúa en el límite inferior izquierdo o en el límite superior derecho, dependiendo de si el problema es de minimización o maximización respectivamente. Si se dispone de un conjunto de rectas uniformemente distribuidas, los puntos de intersección entre el límite definido por el frente de Pareto y ese

conjunto de rectas conforman una buena aproximación del frente de Pareto. Es posible usar este método en MOEA/D usando el punto de referencia  $\mathbf{z}^*$  y los vectores de pesos  $\lambda$  para construir dichas rectas, de forma que todas las rectas pasen por  $\mathbf{z}^*$  y tengan como dirección la que determine cada vector  $\lambda$ . A partir de ahí, puede generarse el siguiente problema de optimización escalar:

$$\begin{aligned} \min_{\mathbf{x} \in \Omega} \quad & g^{bi}(\mathbf{x}|\lambda, \mathbf{z}^*) = d \\ \text{sujeto a} \quad & F(\mathbf{x}) - \mathbf{z}^* = d\lambda \end{aligned} \quad (5.3)$$

Es preciso aclarar que aquí se asume que el problema de optimización es de minimización. Si, por el contrario, el problema es de maximización, la restricción de igualdad definida en (5.3) se ha de cambiar por lo siguiente:

$$\mathbf{z}^* - F(\mathbf{x}) = d\lambda. \quad (5.4)$$

En cualquier caso, esta restricción de igualdad asegura que el vector de valores objetivo  $F(\mathbf{x})$  siempre se sitúe sobre  $L$ , la recta con dirección  $\lambda$  (el vector de pesos) y que pasa por el punto  $\mathbf{z}^*$  (el punto ideal en el espacio objetivo). Esta restricción, sin embargo, resulta difícil de manejar. Para solventar esto, puede utilizarse otro método cuya función de agregación introduzca una penalización, para evitar así la inclusión de restricciones:

$$\min_{\mathbf{x} \in \Omega} \quad g^{bi}(\mathbf{x}|\lambda, \mathbf{z}^*) = d_1 + \theta d_2 \quad (5.5)$$

donde, en caso de minimización,

$$d_1 = \frac{\|(F(\mathbf{x}) - \mathbf{z}^*)^T \lambda\|}{\|\lambda\|} \quad (5.6)$$

$$d_2 = \|F(\mathbf{x}) - (\mathbf{z}^* + d_1 \lambda)\| \quad (5.7)$$

o, si se trata de maximización,

$$d_1 = \frac{\|(\mathbf{z}^* - F(\mathbf{x}))^T \lambda\|}{\|\lambda\|} \quad (5.8)$$

$$d_2 = \|F(\mathbf{x}) - (\mathbf{z}^* - d_1 \lambda)\| \quad (5.9)$$

Geométricamente,  $d_1$  representa la distancia entre  $\mathbf{z}^*$  y la proyección de  $F(\mathbf{x})$  sobre la recta  $L$ , mientras que  $d_2$  es la distancia entre  $F(\mathbf{x})$  y la recta  $L$ , es decir, la distancia entre  $F(\mathbf{x})$  y su proyección en  $L$ . Nótese que aquí  $d_1$  es el equivalente a  $d$  en

la expresión 5.3, y  $d_2$  es el valor que representa. Por su parte,  $\theta > 0$  es un parámetro de penalización que determina cómo de importante es que se minimice  $d_2$ . Si dicho parámetro  $\theta$  es ajustado adecuadamente, las soluciones de ambos métodos (5.3) y (5.5) serán muy similares.

En [43], este método recibe el nombre de Método de Intersección de Frontera Basado en Penalización o PBI (*Penalty-based Boundary Intersection Method*). Su principal ventaja frente al enfoque de Tchebycheff es que se obtienen soluciones más uniformemente distribuidas, para el mismo conjunto de vectores de pesos. Además, para dos soluciones  $\mathbf{x}$  e  $\mathbf{y}$  tales que  $\mathbf{x} \preceq \mathbf{y}$ , con el método de Tchebycheff es posible que ocurra que  $g^{te}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{z}^*) = g^{te}(\mathbf{y}|\boldsymbol{\lambda}, \mathbf{z}^*)$ , mientras que es difícil que pase lo mismo con PBI.

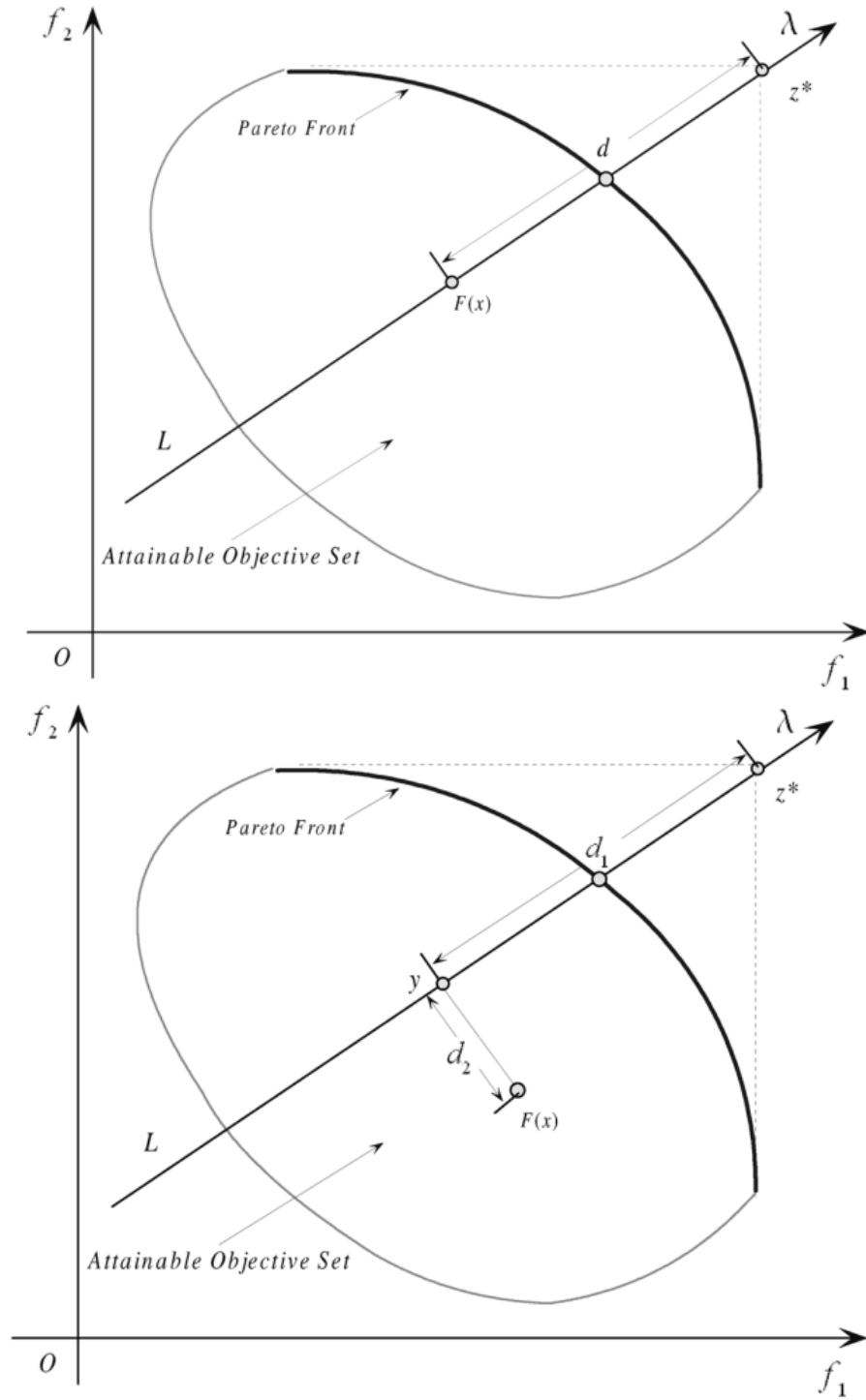


Figura 5.2: Ilustración geométrica de los métodos BI (arriba) y PBI (abajo) para un problema de maximización [43].

Sin embargo, el principal inconveniente del método PBI es el hecho de que  $\theta$  sea un parámetro, y como tal debe ser ajustado adecuadamente. Un valor demasiado bajo o demasiado alto provocará que los resultados empeoren.

## 5.2 EL FRAMEWORK ADA

En este apartado se introducirá el esquema usado en este trabajo que introduce el manejo de la multimodalidad en los algoritmos evolutivos multiobjetivo basados en descomposición. Antes de eso, para poner al lector en situación, se hablará brevemente sobre el concepto de multimodalidad y su relación con los MOEAs.

### 5.2.1 Algoritmos evolutivos multimodales multiobjetivo basados en descomposición

En muchos MOEAs se asume que únicamente la distribución de las soluciones en el espacio objetivo es relevante, dejando en un segundo plano la diversidad en el espacio de soluciones. Sin ir más lejos, un buen ejemplo de esto es el propio MOEA/D, con el cual se puede mantener una buena diversidad en el espacio objetivo si se escoge un método de descomposición adecuado y, sobretodo, si se emplea un conjunto de vectores de pesos que estén uniformemente bien distribuidos [43]. Sin embargo, no existe en MOEA/D ningún mecanismo que mantenga la diversidad en el espacio de soluciones.

Pueden darse situaciones en las que existan diversas soluciones que, aún siendo muy distantes en el espacio de soluciones, son muy próximas o similares en el espacio objetivo, como puede apreciarse en la figura 5.3, que ilustra a la perfección el concepto de *multimodalidad*. El hecho de que en un problema existan formas muy distintas de llegar a (casi) la misma solución es algo muy relevante, pues permite elegir entre distintas soluciones de una calidad similar en función de un cierto criterio o preferencia (por ejemplo, que alguna solución sea más fácil de obtener, que una de las soluciones se vuelva inalcanzable o totalmente inviable, etc). Por ello, en ciertos problemas es importante mantener la diversidad en el espacio de soluciones.

Los llamados algoritmos evolutivos multimodales multiobjetivo son MOEAs pensados para problemas en los que se dan precisamente estas circunstancias, los llamados problemas de optimización multiobjetivo multimodales. Su objetivo consiste en encontrar el mayor número de soluciones Pareto-óptimas equivalentes, para lo cual se sirven de mecanismos para manejar múltiples soluciones equivalentes durante su ejecución. Algunos ejemplos de este tipo de algoritmos son el  $P_{Q,\epsilon}$ -MOEA, el Omni-optimizer y el DIOP, entre otros.

### 5.2.2 ADA: un framework para manejar la multimodalidad

ADA [44] es un método propuesto en 2019 para manejar la multimodalidad en los MOEAs basados en descomposición. No se trata de un algoritmo en sí, sino que es un *framework*, es decir, un esquema o marco general que puede aplicarse a distintos algoritmos.

ADA es una generalización de MOEA/D-AD [45], una versión del MOEA/D capaz de manejar la multimodalidad propuesta por los mismos autores de ADA, con un mecanismo para mantener la diversidad en el espacio basado en el concepto de vecindario de soluciones. Lo cierto es que MOEA/D-AD no es una adaptación trivial del MOEA/D básico, sino que cambia por completo su diseño y funcionamiento.

### 5.2.3 Operadores de ADA

MOEA/D-AD posee distintos componentes y operadores, cada uno de los cuales cumple un papel fundamental. De hecho, el nombre de MOEA/D-AD viene de MOEA/D *Addition-Deletion*, haciendo alusión a dos de sus componentes: sus operadores de adición y eliminación, respectivamente. ADA posee esos mismos operadores, con la diferencia de que los operadores de MOEA/D-AD son de una forma determinada, mientras que en ADA pueden adoptar distintas formas y variantes, dependiendo de sobre qué algoritmo se esté aplicando.

#### 5.2.3.1 Operadores genéticos y de selección

Como todo algoritmo evolutivo, ADA hace uso de operadores de cruce y mutación para generar nuevas soluciones a partir de las ya existentes. Al igual que en MOEA/D, con ADA puede usarse cualquier operador de cruce clásico, como el operador uniforme o el operador de punto fijo, en caso de usar un esquema de codificación binaria o entera para los cromosomas/individuos de la población, u operadores de cruce como el BLX- $\alpha$  [46] o el SBX [47], si se usa codificación real. Por supuesto, también pueden usarse operadores de cruce para problemas específicos.

Por su parte, el método de selección de padres de ADA y MOEA/D-AD es de lo más sencillo: mientras que en el MOEA/D original se seleccionaban dos individuos cuyos respectivos vectores de pesos estuviesen en el vecindario del individuo actual  $\mathbf{x}^i$  (ver líneas 6 y 7 del algoritmo 1), en ADA y MOEA/D-AD simplemente se seleccionan dos padres aleatorios de toda la población  $P$ .

#### 5.2.3.2 Operador de asignación

En MOEA/D, desde el principio habrá uno y solo un individuo asignado a cada vector de pesos  $\lambda$  (cada uno de los cuales representaba un subproblema), de forma que al individuo  $\mathbf{x}^i$  le corresponde el subproblema dado por  $\lambda^i$ . A lo largo de toda la ejecución, tanto el número de individuos en la población como el número de vectores de pesos se mantiene fijo en  $N$ , optimizando cada individuo de acuerdo a su correspondiente vector de pesos.



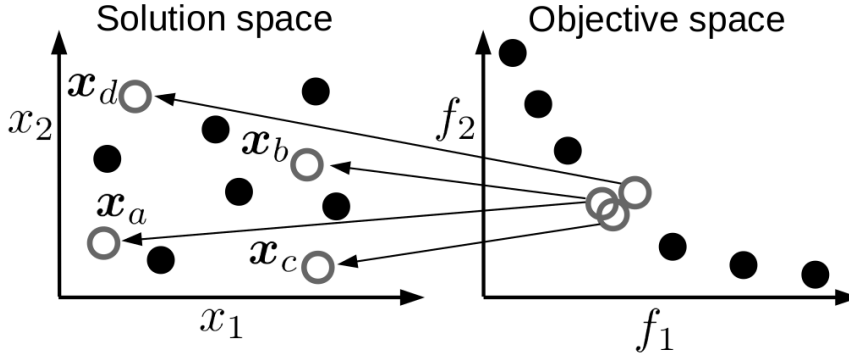


Figura 5.3: Ilustración del concepto de multimodalidad, en la cual soluciones muy dispares en el espacio de soluciones son casi idénticas en el espacio objetivo.[44]

En ADA y MOEA/D-AD no ocurre lo mismo. Mientras que sí se mantienen siempre los mismos vectores de pesos, el tamaño de la población de individuos va cambiando a lo largo del proceso de búsqueda. Además, la forma en la que se asignan las soluciones a los vectores de pesos es diferente: no tiene por qué asignarse el  $i$ -ésimo individuo  $\mathbf{x}^i$  al  $i$ -ésimo vector  $\lambda^i$ , si no que ahora  $\mathbf{x}^i$  se asigna al  $j$ -ésimo vector  $\lambda^j$ , siendo  $\lambda^j$  el vector de pesos que mejor se adapta a  $\mathbf{x}_i$  bajo un cierto criterio.

Es necesario mantener una estructura que permita saber a qué vector de pesos se ha asignado la solución  $\mathbf{x}^i$ . Así pues, se define el vector  $\mathbf{a} = (a_1, \dots, a_\mu)$ , donde  $\mu = |P|$  (recordemos que en ADA y en MOEA/D-AD el tamaño de la población es variable), de forma que si  $a_i = j$  entonces al individuo  $\mathbf{x}^i$  le corresponde el vector  $\lambda^j$ , y por tanto habrá sido asignado al  $j$ -ésimo subproblema escalar.

El criterio para asignar individuos a vectores de pesos puede basarse en uno de los métodos de descomposición descritos en la sección 5.1.2. Si, por ejemplo, se usase el enfoque de Tchebycheff, una solución  $\mathbf{x}$  se asignaría al  $j$ -ésimo subproblema, donde:

$$j = \arg \min_{k \in \{1, \dots, N\}} \{g^{te}(\mathbf{x} | \lambda^k)\} \quad (5.10)$$

En MOEA/D-AD, por su parte, se emplea la distancia perpendicular entre el vector normalizado de valores objetivos  $F(\mathbf{x})$  de una solución  $\mathbf{x}$ , y un vector de pesos  $\lambda$ , que no es sino la distancia entre  $F(\mathbf{x})$  normalizado y su proyección en la recta que pasa por el punto referencia  $\mathbf{z}$  con la dirección determinada por  $\lambda$  (véase el valor  $d_2$  de la fórmula 5.5).

### 5.2.3.3 Criterio de vecindario

Tanto en MOEA/D como en MOEA/D-AD y las variantes de ADA, la diversidad en el espacio objetivo se mantiene mediante la asigna-

ción de individuos (soluciones) a los diferentes subproblemas, descrita justo en el apartado anterior. Sin embargo, en lo que se refiere al espacio de soluciones, ningún mecanismo para mantener la diversidad en el espacio de soluciones se emplea en MOEA/D, mientras que en MOEA/D-AD y ADA sí.

La forma de controlar la diversidad en el espacio de soluciones consiste en un criterio de vecindad. Para ello, en [44] y [45] se propone usar la distancia Euclídea entre soluciones, habiendo normalizado antes las variables de decisión (cromosomas) si en ellas se emplean distintas escalas de valores. De esta forma, puede determinarse para cada solución un vecindario conformado por las  $L$  soluciones más cercanas.

Además de este sencillo criterio basado en distancias, con ADA puede usarse cualquier otro criterio de vecindad o de nichos, pero en el estudio preliminar llevado a cabo en [44] se vió que otros criterios más sofisticados no funcionarion bien.

#### 5.2.3.4 Operador de eliminación

En ADA, el criterio según el cual se descartan las soluciones de la población  $P$  es clave, pues ha sido ideado teniendo en mente la diversidad tanto del espacio de soluciones como la del espacio objetivo.

Sea  $\mathbf{u}$  una nueva solución generada a partir de dos soluciones padres mediante los operadores genéticos. Se construyen dos conjuntos de soluciones a partir de  $\mathbf{u}$ , a los que llamaremos  $Y$  y  $Z$ , definidos de la siguiente forma:

$$\begin{aligned} Y &= \{\mathbf{x} \in P \mid \mathbf{x} \text{ es vecino de } \mathbf{u} \text{ en el espacio de soluciones}\} \\ Z &= \{\mathbf{x} \in P \mid \mathbf{x} \text{ está asignado al mismo subproblema que } \mathbf{u}\} \end{aligned} \quad (5.11)$$

El conjunto  $Y$  se basa en el criterio de vecindario que se haya escogido, del cual se habla en el anterior apartado, mientras que  $Z$  se basa en el operador de asignación. A partir de estos dos conjuntos, se construye uno tercero, al que llamamos  $X$ , compuesto por las soluciones que cumplen tanto la condición de  $Y$  como la de  $Z$ :

$$X = Y \cap Z \quad (5.12)$$

Posteriormente, se compara la nueva solución  $\mathbf{u}$  con cada una de las soluciones de  $X$ . Aquellas soluciones de  $X$  que sean peores que  $\mathbf{u}$  según un cierto criterio serán eliminadas de la población  $P$ .

¿Por qué se comparan con  $\mathbf{u}$  solo las soluciones de  $X$ , y no con las de  $Y$  y luego con las de  $Z$  por separado? Supongamos que, en un momento dado durante la ejecución de ADA, existe una solución  $\mathbf{x}'$  tal que  $\mathbf{x}' \in Z$  y  $\mathbf{x}' \notin Y$ . Si resultase que  $\mathbf{x}'$  fuese peor que  $\mathbf{u}$  y se eliminase de la población  $P$ , estaríamos perdiendo una solución que aporta diversidad al espacio de soluciones, ya que  $\mathbf{u}$  no pertenece al

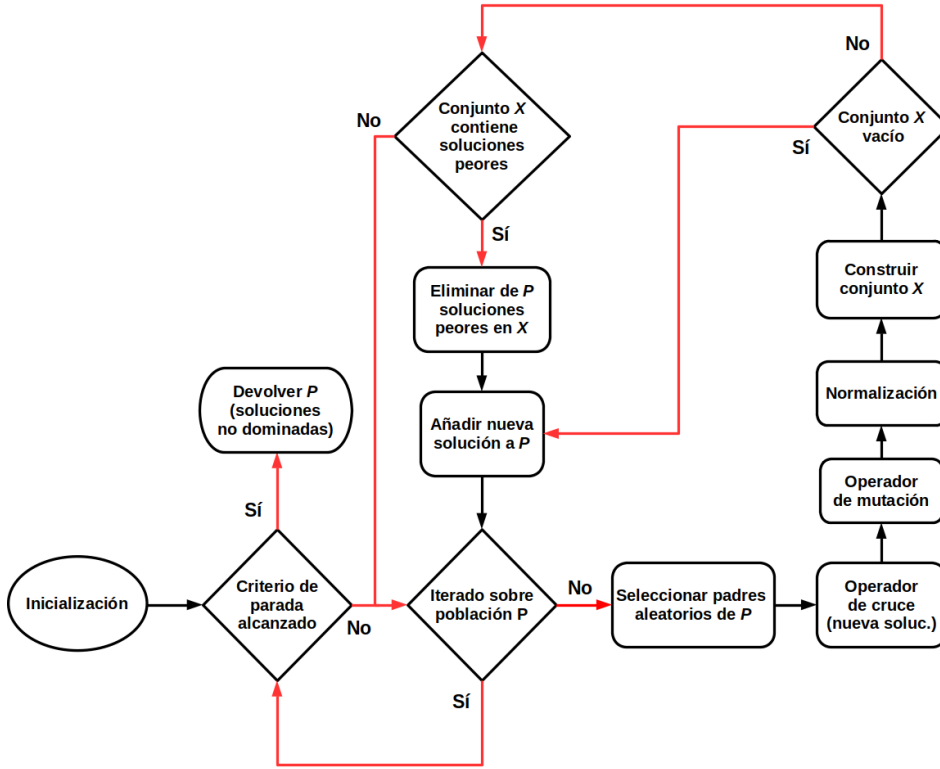


Figura 5.4: Diagrama de flujo del esquema ADA

vecindario de  $\mathbf{u}$ . De forma análoga, si  $\mathbf{x}' \in Y$  pero  $\mathbf{x}' \notin Z$ , eliminar a  $\mathbf{x}'$  tampoco tendría sentido aunque fuese, de alguna forma, peor que  $\mathbf{u}$ , porque ni siquiera está asignado al mismo subproblema que  $\mathbf{u}$ , por lo que se perdería una solución que aporta diversidad al espacio objetivo. Por lo tanto, solo eliminando soluciones que pertenezcan a  $X$  es posible asegurarse de que no se pierde diversidad ni en el espacio de soluciones ni en el espacio objetivo, porque así se eliminan únicamente soluciones que sean vecinas de  $\mathbf{u}$  (cercanas a  $\mathbf{u}$  en espacio de soluciones) y que al mismo tiempo estén asignadas al mismo subproblema que  $\mathbf{u}$  (cercanas a  $\mathbf{u}$  en espacio objetivo).

En ADA pueden usarse distintos criterios para determinar si una solución es peor que otra. Si se aplica ADA en algún algoritmo basado en MOEA/D, como el MOEA/D-AGR [48], puede usarse cualquiera de los métodos de descomposición listados en la sección 5.1.2. Si, por ejemplo, se optase por usar el enfoque de Tchebycheff, entonces una solución  $\mathbf{x}$  sería peor que otra solución  $\mathbf{y}$  si y solo si  $g^{te}(\mathbf{y}|\boldsymbol{\lambda}, \mathbf{z}^*) \leq g^{te}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{z}^*)$  para el mismo vector de pesos  $\boldsymbol{\lambda}$ .

#### 5.2.3.5 Operador de adición

Una vez comparada la nueva solución  $\mathbf{u}$  con todas las soluciones del conjunto  $X$  (si las hubiese), y eliminadas aquellas que fuesen peores, llega el momento de decidir qué hacer con  $\mathbf{u}$ : si añadirla a la

población  $P$  o descartarla. De esto se encargará el operador de adición, el cual añadirá  $\mathbf{u}$  a la población  $P$  si se cumple una de estas dos condiciones:

1. Si el conjunto  $X$  generado a partir de  $\mathbf{u}$  no tiene elementos ( $X = \emptyset$ ).
2. Si  $X$  sí tiene elementos, pero al menos uno de ellos es peor que  $\mathbf{u}$ .

¿Por qué se añade  $\mathbf{u}$  si  $X = \emptyset$ ? Si  $X$  no tiene elementos, entonces no existen soluciones en  $P$  que sean vecinas de  $\mathbf{u}$  en el espacio de soluciones y que al mismo tiempo estén asignadas al mismo subproblema (puede que existan soluciones que cumplan una condición u otra, pero no ambas a la vez, ya que en tal caso formarían parte de  $X$ ). Por tanto,  $\mathbf{u}$  es una solución que aporta diversidad, ya sea al espacio de soluciones, al espacio objetivo, o a ambos, por lo que tiene sentido añadirla a la población  $P$  aunque esta ya contenga soluciones que sean mejores que  $\mathbf{u}$ .

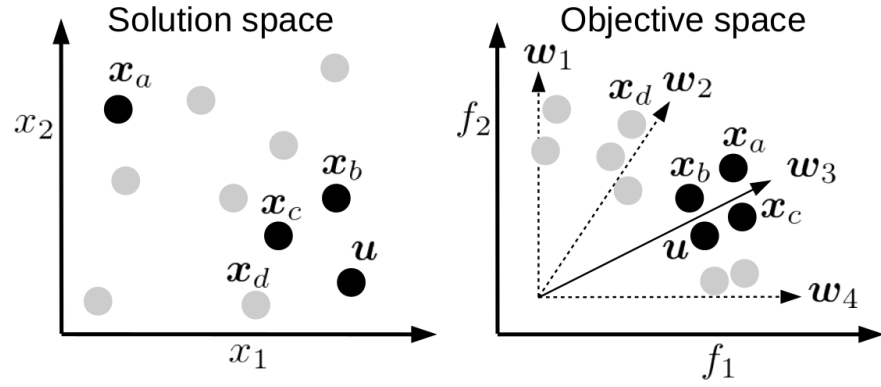


Figura 5.5: Ejemplo que ilustra cómo las soluciones asignadas al mismo subproblema/vector de pesos son cercanas entre sí en el espacio objetivo. [44]

¿Por qué, si  $X \neq \emptyset$ ,  $\mathbf{u}$  se añade solo si alguna de las soluciones de  $X$  es peor que  $\mathbf{u}$ ? Porque al ser vecinas de  $\mathbf{u}$  y estar asignadas al mismo subproblema, no se perderá diversidad si son sustituidas por  $\mathbf{u}$ . Por tanto, si se han eliminado de  $P$  soluciones que formaban parte de  $X$  por ser peores que  $\mathbf{u}$ , lo lógico es añadir a la propia  $\mathbf{u}$  a  $P$  para reemplazarlas.

---

**Algoritmo 1:** MOEA/D

---

**Entrada:** Número de subproblemas  $N$ , tamaño del vecindario de los vectores de pesos  $T$ , opcionalmente conjunto de vectores de pesos uniformemente distribuidos  $\{\lambda_1, \dots, \lambda_N\}$

```

/* Inicialización */
1 Inicializar población externa  $EP = \emptyset$ 
2 Generar conjunto de vectores de pesos uniformemente
  distribuidos  $\{\lambda_1, \dots, \lambda_N\}$  (si no se ha pasado como entrada)
3 Calcular distancia euclídea entre todos los pares de vectores
  de pesos. Hacer  $B(i) = \{i_1, \dots, i_T\}$  para todo  $i \in \{1, \dots, N\}$ ,
  de forma que  $\lambda^{i_1}, \dots, \lambda^{i_T}$  son los  $T$  vectores más cercanos a
   $\lambda^i$ 
4 Inicializar, aleatoriamente o de otra forma, la población  $P$ 
5 Obtener valores objetivos de la población  $FV$ :  $FV^i = F(\mathbf{x}^i)$ 
  para todo  $i \in \{1, \dots, N\}$ 

/* Actualización */
6 while criterio de parada no se cumple do
7   for  $i \in \{1, \dots, N\}$  do
8     Elegir aleatoriamente dos valores  $l, k \in B(i)$ 
9     Generar un descendiente  $\mathbf{y}$  aplicando el operador de
      cruce sobre  $\mathbf{x}^l$  y  $\mathbf{x}^k$ 
10    Aplicar sobre  $\mathbf{y}$  el operador de mutación. Aplicar
      opcionalmente mejora o reparación.

    /* Actualizar punto ideal  $z$  */
11    for  $j \in \{1, \dots, m\}$  do
12      if  $f_j(\mathbf{y}) \leq z_j$  then
13         $z_j = f_j(\mathbf{y})$ 
14      end
15    end

    /* Actualizar soluciones vecinas */
16    for  $j \in B(i)$  do
17      if  $g(\mathbf{y}|\lambda^j) \leq g(\mathbf{x}^j|\lambda^j)$  then
18         $\mathbf{x}^j = \mathbf{y}$ ,  $FV_j = F(\mathbf{y})$ 
19      end
20    end

    /* Actualizar población externa */
21    Eliminar de  $EP$  todas las soluciones dominadas por
       $F(\mathbf{y})$ 
22    Añadir  $F(\mathbf{y})$  a  $EP$  si ningún vector de  $EP$  domina a
       $F(\mathbf{y})$ 
23  end
24 end
25 return  $EP$ 

```

---

---

**Algoritmo 2:** El framework ADA
 

---

**Entrada:** Número de subproblemas  $N$ , proporción de vecindario  $\tau$

```

/* Inicialización */
1 Inicializar población externa  $EP = \emptyset$ 
2 Generar conjunto de vectores de pesos  $\{\lambda_1, \dots, \lambda_N\}$ 
3 Inicializar, aleatoriamente o de otra forma, la población  $P$ 
4 Obtener  $FV$ :  $FV^i = F(\mathbf{x}^i)$  para todo  $i \in \{1, \dots, N\}$ 
5 Inicializar problemas asignados  $\mathbf{a} = (a_1, \dots, a_N)$ 
/* Op. Asignación */
6 for  $i \in \{1, \dots, N\}$  do
7   Encontrar  $j$ -ésimo subproblema que mejor se adapte a  $\mathbf{x}^i$ 
8    $a_i = j$ 
9 end
10 while criterio de parada no se cumple do
    /* Tamaño de población y vecindario */
11    $\mu = |P|$ ,  $T = \mu \cdot \tau$ 

    /* Selección de padres y op. genéticos */
12   Seleccionar dos valores aleatorios  $a, b \in \{1, \dots, \mu\}$ 
13   Generar un descendiente  $\mathbf{u}$  aplicando el operador de
       cruce sobre  $\mathbf{x}^a$  y  $\mathbf{x}^b$ 

14   Aplicar sobre  $\mathbf{u}$  el operador de mutación. Aplicar
       opcionalmente mejora o reparación.
15   Normalizar los vectores de valores objetivos  $F(\mathbf{u})$  y  $FV$ 
16   Encontrar  $j$ -ésimo subproblema que mejor se adapte a  $\mathbf{u}$ 
17   Generar vecindario de  $\mathbf{u}$  ( $T$  soluciones más cercanas)
        $X = \{\mathbf{x} \in P \mid \mathbf{x} \text{ está asignado al subproblema } j \wedge$ 
        $\mathbf{x} \text{ está en el vecindario de } \mathbf{u}\}$ 

    /* Op. Eliminación */
18    $b_{\text{ganador}} = FALSE$ 
19    $b_{\text{explorador}} = FALSE$ 
20   if  $X = \emptyset$  then
21      $b_{\text{explorador}} = TRUE$ 
22   end
23   for  $\mathbf{x} \in X$  do
24     if  $\mathbf{x}$  es peor que  $\mathbf{u}$  then
25        $P = P \setminus \{\mathbf{x}\}$ 
26        $b_{\text{ganador}} = TRUE$ 
27     end
28   end

    /* Op. Adición */
29   if  $b_{\text{explorador}} \vee b_{\text{ganador}}$  then
30      $P = P \cup \{\mathbf{u}\}$ 
31     Añadir  $j$  al final de  $\mathbf{a}$ 
32   end
33 end
34 return  $P$ 
  
```

---

---

**Algoritmo 3:** MOEA/D-AD
 

---

**Entrada:** Número de subproblemas  $N$ , tamaño del vecindario de los vectores de pesos  $T$ , opcionalmente conjunto de vectores de pesos uniformemente distribuidos  $\{\lambda_1, \dots, \lambda_N\}$

**Salida:** Población externa  $EP$

- <sub>1</sub> Adicion
  - <sub>2</sub> Eliminacion
  - <sub>3</sub> **return**  $EP$
-





Parte II

EXPERIMENTACIÓN



## METODOLOGÍA

---

Como se anticipó en el capítulo introductorio, la segunda parte de este trabajo gira en torno a la experimentación realizada. En este capítulo en concreto se explicarán todos los datos, parámetros, variantes y medidas de rendimiento elegidos para los experimentos.

### 6.1 ESQUEMA DE CODIFICACIÓN

Para aplicar ADA al problema del clustering con restricciones, se empleará un esquema de **representación entera basada en etiquetas**. Existen dos principales motivos por los que la representación basada en etiquetas es una buena opción para el clustering y el clustering con restricciones:

1. La representación basada en etiquetas permite la **formación de clusters con cualquier forma**. En general, los esquemas de codificación entera, tanto la basada en etiquetas como la basada en grafos, son capaces de generar clusters no convexos, a diferencia de los esquemas de representación binarios y de representación real, los cuales, al basarse en la cercanía de cada instancia a los centroides, únicamente forman clusters convexos.
2. La representación basada en etiquetas es la **forma de representación más directa**. Este esquema de representación es la única que no requiere de un proceso de *decodificación* a posteriori para averiguar a qué cluster pertenece cada punto o instancia, puesto que esa información ya queda representada de forma explícita. Por tanto, será más rápido evaluar una solución (es decir, obtener sus valores objetivo).

En el caso de la representación entera basada en grafos, habría que decodificar los grafos dirigidos que representan las soluciones, viendo en cada caso a qué otras instancias está conectada cada una para identificar los componentes conexos, cada uno de los cuales representaría un cluster distinto.

Por su parte, en la representación binaria y la representación real, para saber a qué cluster pertenece cada instancia, hace falta comprobar cuál de los  $k$  centroides es el más cercano a ella.

En definitiva, al decodificar una solución de alguno de los esquemas de representación que así lo requieren, el resultado obtenido es un vector con las etiquetas que indican a qué cluster pertenece cada objeto o instancia, por lo que si se opta desde

el principio por la representación basada en etiquetas no haría falta aplicar dicho proceso de decodificación a las soluciones.

## 6.2 FUNCIONES OBJETIVO A OPTIMIZAR

Como ya se dijo, en este trabajo se pretende enfocar el problema de clustering con restricciones como un problema de optimización. Así pues, se emplearán tres funciones objetivo: dos de ellas corresponden a índices de validación de los vistos en la sección 2.3.3, y una tercera mide el grado en que se satisfacen (o violan) las restricciones.

Los dos índices de validación elegidos son el **índice Davies-Bouldin** y el **índice de conectividad**, definidos según las expresiones (2.7) y (2.4), respectivamente. El índice Davies-Bouldin consiste en un valor que tiene en cuenta tanto la cohesión de cada cluster como la separación de cada cluster. Por su parte, el índice de conectividad penaliza a las instancias cuyas instancias vecinas (o sea, las más cercanas) pertenezcan a clusters distintos, favoreciendo que los elementos de distintos clusters se mantengan a una cierta distancia y contribuyendo, de esta forma, a la separación entre clusters. En definitiva, estos dos índices juntos constituyen una buena forma de estimar (y mejorar) la calidad de una solución de clustering.

Por su parte, la tercera función objetivo, que es la que tiene en cuenta las restricciones, consiste simplemente en el **número de restricciones insatisfechas**, es decir, la suma de restricciones, tanto Must-Link como Cannot-Link, que no han sido satisfechas:

$$Insatis(\mathbf{C}, C_-, C_{\neq}) = Insatis(\mathbf{C}, C_-) + Insatis(\mathbf{C}, C_{\neq}), \quad (6.1)$$

donde  $\mathbf{C}$  es una solución de clustering, y  $C_-$  y  $C_{\neq}$  son los conjuntos de restricciones ML y CL, respectivamente.

Las tres funciones objetivo son de minimización, y además están acotadas inferiormente ya que el valor mínimo de todas ellas es de 0. Esto último será especialmente relevante a la hora de definir el vector de referencia  $\mathbf{z}^*$ , usado en los métodos de descomposición de Tchebycheff, BI y PBI.

## 6.3 VARIANTES DE ADA

Este trabajo tiene como fin comprobar cómo funciona ADA con el problema del clustering con restricciones. Para ello, se aplicará ADA al algoritmo MOEA/D y se introducirán pequeñas modificaciones hasta crear un total de 8 variantes de ADA distintas, las cuales serán comparadas entre ellas. Las variantes con las que se experimentará son las siguientes:

- **MOEA/D-ADA**: la versión más básica que se probará. Tal y como ocurre en el algoritmo MOEA/D-AGR-ADA, la forma de

comprobar si una solución es peor que otra, así como la asignación de las soluciones a subproblemas, se hará con el método de Tchebycheff. No se aplicará normalización a los valores objetivo de las soluciones.

- **MOEA/D-ADA<sub>Norm</sub>**: esta versión es exactamente igual a la anterior, con la única diferencia de que los valores objetivo se normalizan, tal y como se especifica en el planteamiento original de ADA, con el fin de que las distintas en las que se mueven las funciones objetivo no influyan. La normalización consiste en un escalado simple que utiliza el punto ideal  $\mathbf{z}^*$  para que los valores objetivo pasen al rango  $[0, 1]$ . Suponiendo minimización:

$$f'_i(\mathbf{x}) = \frac{f_i(\mathbf{x}) - \mathbf{z}_{\text{peor}}^*}{\mathbf{z}^* - \mathbf{z}_{\text{peor}}^*}, \quad (6.2)$$

donde  $\mathbf{z}_{\text{peor}}^*$  es el punto contrario a  $\mathbf{z}^*$ , es decir, el punto con los máximos valores objetivo encontrados hasta el momento.

- **MOEA/D-ADA-PD**: partiendo de la versión básica, MOEA/D-ADA, en la asignación de subproblemas se empleará, en lugar del método de Tchebycheff, la distancia perpendicular entre la solución y el vector de pesos, tomando la idea del algoritmo MOEA/D-AD.
- **MOEA/D-ADA-PD<sub>Norm A</sub>**: idéntica a **MOEA/D-ADA-PD**, pero con normalización de valores objetivo
- **MOEA/D-ADA-PD<sub>Norm B</sub>**:
- **MOEA/D-ADA-PD-SSPS**:
- **MOEA/D-ADA-KM**: versión idéntica a MOEA/D-ADA, pero aplicando el algoritmo de  $k$ -medias al 20 % de la población inicial. Esto puede ser beneficioso si en los conjuntos de datos con los que se vayan a usar estos algoritmos predominan los clusters convexos, pero puede no ser el caso. De hecho, este es uno de los motivos por los que se utilizan restricciones: para evitar la tendencia a generar clusters convexos o hiperesféricos, corrigiendo su forma según la información disponible sobre la solución óptima.
- **MOEA/D-ADA-PD-KM**: de forma análoga a la versión anterior, esta versión es como MOEA/D-ADA-PD pero aplicando  $k$ -medias al 20 % de la población tras la inicialización.

Tabla 6.1: Variantes de ADA comparadas

Variante	Normalización	Op. selección	Op. asignación	Inic. con $k$ -medias
MOEA/D-ADA	No	Padres aleatorios	Tchebycheff	No
MOEA/D-ADA <sub>Norm</sub>	Sí	Padres aleatorios	Tchebycheff	No
MOEA/D-ADA-PD	No	Padres aleatorios	Dist. perpendicular	No
MOEA/D-ADA-PD <sub>Norm A</sub>	Sí	Padres aleatorios	Dist. perpendicular	No
MOEA/D-ADA-PD <sub>Norm B</sub>	Solo en op. asign.	Padres aleatorios de $P$	Dist. perpendicular	No
MOEA/D-ADA-PD-SSPS	No	Padres aleat. asignados al mismo subprob.	Tchebycheff	No
MOEA/D-ADA-KM	No	Padres aleatorios	Tchebycheff	20 % población
MOEA/D-ADA-PD-KM	No	Padres aleatorios de	Dist. perpendicular	20 % población

### 6.3.1 Inclusión de población externa

En todas las variantes se ha incluido un mecanismo de población externa ( $EP$ ) que no existía en ADA. Obtener soluciones no dominadas de  $P$  da como resultado demasiado pocas soluciones, con lo cual es menos probable terminar con soluciones tan buenas que como si se tuviese una población externa que guarde todas las soluciones no dominadas obtenidas hasta el momento a lo largo de la ejecución del algoritmo. Puede que se pierda algo de diversidad (en espacio objetivo y espacio de soluciones) porque puede que existan soluciones que durante el proceso de búsqueda de ADA fuesen eliminadas por ser peores que una solución que fuese mejor según el método de Tchebycheff, y que además estas estuviesen en su vecindario y asignadas al mismo subproblema. Sin embargo, tras los estudios preliminares, se puede decir que se consiguen mejores resultados en general manteniendo las soluciones no dominadas en una población externa.

### 6.3.2 Métrica de distancia en el espacio de soluciones

Algo a tener en cuenta a la hora de aplicar ADA a un problema de optimización es el criterio de vecindario de soluciones de ADA (ver sección 5.2.3.3). En todas las variantes que se han usado aquí, el criterio de vecindario usado para construir el conjunto  $X$  se basa en la **distancia euclídea**. Bien es cierto que en [44] y en [45] se emplea la distancia euclídea normalizada, es decir, la distancia euclídea entre dos puntos habiendo normalizado antes ambos puntos para que sus valores/características/cromosomas adopten únicamente valores entre 0 y 1. Sin embargo, en nuestro caso no es necesario normalizar las soluciones, puesto que, al haber optado por un esquema de codificación entera basado en etiquetas, todas las variables/características tienen el mismo rango de valores, esto es, valores enteros entre el 0 y  $K$ , siendo  $K$  el número de clusters para ese conjunto de datos.

No obstante, en el problema que nos ocupa (clustering con restricciones) y el esquema de representación de soluciones escogido (codificación entera basada en etiquetas), existe un problema a la hora de usar la distancia euclídea entre soluciones. Tal y como se dijo en

la sección 5.2.3.3, la idea del criterio de vecindario se emplea para determinar si una solución aporta o no diversidad al espacio de soluciones en función de si se considera una solución vecina, o lo que es lo mismo, una solución cercana. Para ello, el criterio de vecindario se sirve de una métrica de distancia dada para saber qué soluciones son más similares entre ellas, de forma que, como es lógico, cuanto más cercanas sean dos soluciones en el espacio de soluciones, más similares serán. Así pues, la distancia euclídea podría decirse que es la opción estándar para determinar la cercanía o similitud entre dos puntos en un espacio  $D$ -dimensional. No obstante, cuando las soluciones codifican la etiqueta asignada a cada instancia en un problema de clustering (o de clustering con restricciones), la distancia euclídea puede que no sea la mejor de las opciones. A continuación se verá por qué.

Supongamos que se pretende resolver un problema de clustering con  $k = 2$  clusters o etiquetas para un conjunto de datos de 4 instancias. Supongamos también que, en la población de soluciones  $P$ , existen dos soluciones  $\mathbf{x}_a$  y  $\mathbf{x}_b$  tales que  $\mathbf{x}_a = (0, 0, 1, 1)$  y  $\mathbf{x}_b = (1, 1, 0, 0)$ . La distancia euclídea entre ambos puntos será de  $\sqrt{1^2 + 1^2 + 1^2 + 1^2} = \sqrt{4} = 2$ . Sin embargo, lo cierto es que tanto  $\mathbf{x}_a$  como  $\mathbf{x}_b$  **representan exactamente la misma solución**: los dos primeros puntos o instancias del conjunto de datos pertenecen a un cluster, y los dos últimos al otro. Es cierto que, con la distancia euclídea, la distancia desde  $\mathbf{x}_a$  a sí misma es igual a 0, y lo mismo con  $\mathbf{x}_b$ , pero la distancia entre  $\mathbf{x}_a$  y  $\mathbf{x}_b$  no es igual a 0 a pesar de codificar lo mismo.

#### 6.4 PARÁMETROS

Para hacer la comparativa lo más justa posible, se emplearán los mismos parámetros en todas las variantes de ADA:

- El **operador de cruce** será el **uniforme**. Se trata de un operador de propósito general ampliamente usado, que puede utilizarse en cualquier problema que admita un esquema de codificación entera.
- El **método de descomposición** usado para descomponer el frente de Pareto en subproblemas de optimización escalares será el de **Tchebycheff**. Es cierto que PBI puede ofrecer mejores resultados, pero para ello habría que invertir bastante tiempo en ajustar de la forma óptima el parámetro  $\theta$  (ver ecuación (5.5)).
- La **probabilidad de heredar** del primer padre en el operador de cruce será del 0,5. De esta forma existirá la misma probabilidad para ambos padres.
- La **probabilidad de mutar** un gen será de 0,1. En los pequeños estudios preliminares, este ha demostrado ser un valor que da

un buen equilibrio entre la introducción de indeterminismo y nueva información (exploración), y la mantención de las características heredadas de los padres (explotación).

- El **número de subproblemas**  $N$ , así como el valor inicial de  $\mu$  (el tamaño de la población  $P$ ), será de 100, el mismo usado en la experimentación realizada en [45].
- El **tamaño de vecindario**  $\lfloor \tau \cdot \mu \rfloor$ , dado por el valor de proporción de vecindario  $\tau$ , será de  $\lfloor 0,1 \cdot \mu \rfloor$ . En [44] y [45] ese es el valor que mejores resultados parece ofrecer.
- El **punto de referencia**, usado en el método de Tchebycheff, quedará fijado en  $\mathbf{z}^* = (0,0,0)^T$ . Teniendo en cuenta que las tres funciones objetivo son de minimización, que todas están acotadas inferiormente, y que además el ínfimo de todas es 0, establecer  $\mathbf{z}^*$  en ese valor es lo más lógico.
- El **máximo número de evaluaciones** se establecerá en unas 300000 evaluaciones. Se trata una cantidad adecuada, porque da el suficiente tiempo a los algoritmos a que obtengan soluciones lo suficientemente buenas pero sin que se emplee demasiado tiempo en cada ejecución.

Tabla 6.2: Valores escogidos para los parámetros e hiperparámetros de ADA

Parámetro	Valor
Operador de cruce	Uniforme
Método de descomposición	Tchebycheff
Probabilidad de cruce	0,5
Probabilidad de mutación	0,1
Número de subproblemas $N$	100
Tamaño de vecindario $\lfloor \tau \cdot \mu \rfloor$	$\lfloor 0,1 \cdot \mu \rfloor$
Punto ideal/de referencia $\mathbf{z}^*$	$(0,0,0)^T$
Nº máximo evaluaciones	300000

## 6.5 CONJUNTOS DE DATOS Y DE RESTRICCIONES

Los experimentos se han realizado con un total de 20 conjuntos de datos distintos. Todos ellos son bastante conocidos y ampliamente usados como *benchmarks* para probar distintos algoritmos, tanto ya existentes como nuevas propuestas de mejora del estado del arte. Todos los conjuntos de datos están disponibles en el repositorio de datasets de KEEL, la herramienta de minería de datos basada



en algoritmos evolutivos desarrollada en la Universidad de Granada (ref), y pueden ser descargados desde el siguiente enlace: <https://sci2s.ugr.es/keel/category.php?cat=clas>. En la tabla 6.3 se expone en un formato resumido la información más relevante de estos 20 conjuntos de datos.

Tabla 6.3: Resumen de los conjuntos de datos

Dataset	Nº Instancias	Dimensiones (Nº características)	Nº clusters (sol. verdadera)
Appendicitis	106	7	2
Balance	625	4	3
Banana <sup>1</sup>	1590	2	2
Bupa	345	5	16
Ecoli	336	7	8
Glass	214	9	6
Haberman	306	3	2
Hayes Roth	160	4	3
Heart	270	13	2
Iris	150	4	3
Led7Digit	500	7	10
Monk2	432	6	2
Newthyroid	215	5	3
Pima	768	8	2
Saheart	462	9	2
Soybean	47	35	4
Tae	151	5	3
Titanic <sup>1</sup>	661	3	2
Wine	178	13	3
Zoo	101	16	7

A su vez, para cada conjunto de datos se dispone de tres conjuntos de restricciones a nivel de instancia, que quedan en seis si separamos cada conjunto en restricciones Must-Link y restricciones Cannot-Link. Las restricciones han sido obtenidas directamente de las etiquetas de las soluciones verdaderas, de forma aleatoria y evitando cualquier tipo de sesgo. En concreto, el número de restricciones que contendrán cada uno de los tres conjuntos de restricciones para cada dataset afectarán al 10 %, 15 % y del 20 % de objetos o instancias del correspondiente conjunto de datos, respectivamente. De esta forma, se podrá comprobar en qué medida afecta a los resultados distintas cantidades de restricciones en un mismo conjunto de datos.

Se puede apreciar en la tabla 6.3 la diversidad presente en las características de los conjuntos de datos: hay conjuntos de datos con muchas instancias y pocas dimensiones, así como conjuntos de po-

Tabla 6.4: Tamaños de los conjuntos de restricciones

Dataset	Restric. 10 %		Restric. 15 %		Restric. 20 %	
	ML	CL	ML	CL	ML	CL
Appendicitis	37	18	76	44	154	77
Balance	841	1112	1846	2525	3324	4426
Banana (muestra)	6368	6193	14311	14130	25509	24894
Bupa	91	504	217	1109	374	1972
Ecoli	147	414	352	923	644	1634
Glass	58	173	138	390	233	670
Haberman	273	192	631	404	1173	718
Hayes Roth	47	73	86	190	177	319
Heart	173	178	436	384	747	684
Iris	28	77	92	161	132	303
Led7Digit	133	1092	258	2517	492	4458
Monk2	484	462	1064	1016	1835	1906
Newthyroid	125	106	273	255	488	415
Pima	1572	1354	3601	3069	6443	53338
Saheart	613	468	1281	1134	2368	1910
Soybean	1	9	11	17	5	40
Tae	31	89	88	165	164	301
Titanic (muestra)	1249	962	2721	2229	4908	3870
Wine	57	96	105	246	210	420
Zoo	13	42	27	93	53	157

cas instancias pero con una dimensionalidad considerable. También puede verse en la tabla 6.4 que, para algunos conjuntos de datos, las cantidades restricciones ML y CL están más o menos balanceadas, mientras que en otros tienen un peso mucho mayor las restricciones CL.

## 6.6 EJECUCIÓN Y MÉTRICAS DE EVALUACIÓN

Para medir los resultados obtenidos con cada variante de ADA, se emplearán 4 medidas distintas, las cuales podemos agrupar en dos categorías: las métricas a nivel de partición o solución, y las métricas a nivel de frente de Pareto.

### 6.6.1 Métricas a nivel de partición

Las métricas a nivel de partición miden la calidad de una solución individual obtenida por un determinado algoritmo de clustering (con o sin restricciones). En el caso de la optimización multiobjetivo, pue-

den usarse para determinar la mejor solución de un frente de Pareto, lo que a su vez permite tener una idea de la calidad general de todo el frente.

La primera métrica que se usará es el **Índice de Rand Ajustado** o **ARI** (*Adjusted Rand Index*). ARI es la versión corregida del Índice de Rand o RI (*Rand Index*), que a su vez es un método para comparar dos particiones (soluciones de clustering), calculando para ello el grado de similitud entre las dos particiones dadas (cite RI). Sean  $C_1$  y  $C_2$  dos soluciones de clustering, esto es, dos particiones distintas de un mismo conjunto de datos  $X$ . Cualquier partición puede entenderse como una colección de  $\binom{n}{2}$  valores correspondientes a pares  $(i, j)$  (siendo  $n = |X|$  el número de instancias en el conjunto de datos), de forma que a cada par  $(i, j)$  le correspondería un valor que indicase si los elementos  $x_i \in X$  y  $x_j \in X$  pertenecen a un mismo cluster o no. Por ejemplo, si al par  $(i, j)$  le corresponde el valor 1, significaría que  $x_i$  y  $x_j$  sí pertenecen al mismo cluster, y si fuese 0 entonces pertenecen a distintos clusters. Así pues, sea  $s$  el número de pares en los que ambos elementos pertenecen al mismo cluster tanto en la partición  $C_1$  como en  $C_2$ , y sea  $d$  el número de pares que pertenecen a clusters distintos en ambas particiones. El índice de Rand se calcula según la siguiente expresión (cite MEMOEA/D):

$$RI(C_1, C_2) = \frac{s + d}{\binom{n}{2}} = \frac{s + d}{n(n-1)/2}. \quad (6.3)$$

$RI$  puede adoptar valores dentro del intervalo  $[0, 1]$ , de forma que cuanto más cercano a 1 sea el valor, más parecidas serán las particiones.

La anterior expresión puede ser corregida para lidiar con particiones aleatorias mediante el uso de un valor esperado del índice de Rand y un valor máximo, dando como resultado lo que se conoce como Índice de Rand Ajustado (cite ARI):

$$ARI(C_1, C_2) = \frac{RI(C_1, C_2) - RI \text{ Esperado}}{RI \text{ Máximo} - RI \text{ Esperado}}, \quad (6.4)$$

donde  $RI$  Esperado es el valor esperado del índice de Rand para una partición aleatoria, y  $RI$  Máximo se asume que es igual a 1.  $ARI$  puede adoptar valores en el intervalo  $[-1, 1]$ . Las particiones aleatorias tenderán a tener un índice de Rand cercano al  $RI$  esperado cuando se comparen con otra solución, por lo que su  $ARI$  será cercano a 0. De esta forma, las agrupaciones generadas de forma aleatoria difícilmente adoptarán valores cercanos a 1.

Tanto  $RI$  como  $ARI$  son usados para medir la calidad de una solución al compararla con la solución verdadera. A diferencia de los índices de valoración presentados en la sección 2.3.3, ARI y el índice Rand hacen uso de la solución óptima para evaluar soluciones de clustering. Después de todo, ambos son métodos para comparar cómo de

similares son dos particiones, y comparar una solución cualquiera directamente con la verdadera (óptima) permite saber cómo de buena es dicha solución.

La segunda medida basada en particiones consiste en la **proporción de restricciones que no han sido satisfechas**, que en (cite MEMOEA/D) recibe el nombre de **UNSAT**. Se calcula como el número de restricciones no satisfechas entre el número total de restricciones.

$$UNSAT(\mathbf{C}) = \frac{Insatis(\mathbf{C}, C_-, C_+)}{|C_-| + |C_+|} \quad (6.5)$$

Como ya se ha dicho, se trata de una proporción, por lo que siempre adopta valores en el intervalo  $[0, 1]$ , y lo deseable en cualquier caso es obtener valores lo más cercanos a 0 posibles, ya que esto significaría que apenas se están violando restricciones.

Es importante recordar que estas dos medidas evalúan particiones, o sea, soluciones de clustering. Se tratan, por tanto, de medidas específicas del problema que estamos tratando, por lo que los índices de validación que se escojan como funciones objetivo, las cuales también son específicas del problema, repercutirán en los valores resultantes de estas medidas.

#### 6.6.2 Métricas a nivel de frente de Pareto

Las métricas de frente de Pareto, como su propio nombre indica, hacen referencia a las características del frente de Pareto entero, en lugar de enfocarse en una solución concreta. Estas medidas son propias de la optimización multiobjetivo en general, por lo que son independientes del problema a resolver. Por tanto, con este tipo de métricas no se entra a valorar la idoneidad de las funciones objetivo, específicamente elegidas para el problema en cuestión. En [49] se enumeran diversos métodos de evaluación para la optimización multiobjetivo.

La primera métrica que se usará, la cual es mencionada en [49], es el **hipervolumen** del frente de Pareto. Propuesto por primera vez en [50], y analizado en [51] para estudiar formas eficientes de calcularlo, el hipervolumen puede definirse de la siguiente forma:

**Definición 6.1 Hipervolumen:** Dado un conjunto finito de puntos  $\mathcal{P}$  en el ortante positivo  $\mathbb{R}_{\geq 0}^d$ , el indicador de hipervolumen se define como el volumen  $d$ -dimensional del politopo ortogonal sin agujeros

$$\Pi^d = \{\mathbf{x} \in \mathbb{R}_{\geq 0}^d \mid \mathbf{p} \preceq \mathbf{x} \wedge \mathbf{x} \preceq \mathbf{r} \ \forall \mathbf{p} \in \mathcal{P}\}, \quad (6.6)$$

donde  $\mathbf{r}$  es un punto de referencia usado para calcular el hipervolumen, el cual es dominado por todos los puntos de  $\mathcal{P}$ . El politopo  $\Pi^d$  corresponde a todo el espacio de  $\mathbb{R}_{\geq 0}^d$  dominado por todos los puntos de  $\mathcal{P}$  y que, a su vez, domina a  $\mathbf{r}$  [51] (cite MEMOEA/D).

La métrica del hipervolumen nos permite saber cómo de compacto es el frente de Pareto obtenido, puesto que si este está compuesto de muchos puntos muy cercanos entre sí, habrá menos *huecos* en el polítopo, y el hipervolumen será mayor. Además, si se utiliza siempre el mismo punto de referencia  $r$  en cada conjunto de datos, podremos saber qué algoritmos generan un frente de Pareto más cercano al origen de coordenadas.

La segunda métrica es sencilla: se trata del **tamaño del frente de Pareto**. Con esto, se podrá conocer cuántas soluciones no dominadas produce un cierto algoritmo o variante. Sin embargo, esta medida no aporta información demasiado útil por sí sola, sino que es necesario tenerla en cuenta en su contexto, y verla junto con el resto de métricas de calidad.

### 6.6.3 Ejecución de los experimentos

Todos los experimentos han sido ejecutados en el servidor de cómputo Hércules de la Universidad de Granada. En el momento de realizar los experimentos, Hércules consta de 19 nodos de cómputo, cada uno de los cuales dispone de 2 procesadores Intel Xeon Silver 4214 a 2.2 GHz (12 núcleos), 56 GB de memoria RAM DDR4, 1 disco duro SATA de 6 TB, 1 disco SSD NVME de 512 GB y el sistema operativo Ubuntu 20.04 LTS.

Cada variante de ADA será ejecutada unas 5 veces con cada conjunto de datos y cada conjunto de restricciones. Esto es: 5 ejecuciones por 20 conjuntos de datos por 3 conjuntos de restricciones, lo que hacen un total de  $5 \times 20 \times 3 = 300$  ejecuciones con cada variante de ADA.

En cada ejecución de una variante de ADA se obtienen como salida, entre otras cosas, los frentes de Pareto (valores objetivo) y las correspondientes soluciones (cromosomas), con los cuales se calculan las métricas de rendimiento que previamente se han descrito. Posteriormente, se promedian (obteniendo tanto la media como la desviación típica) los resultados de las 5 ejecuciones, con lo cual, para cada variante de ADA, obtendremos los correspondientes valores de las 4 métricas de rendimiento para cada conjunto de datos y cada conjunto de restricciones.

<sup>1</sup> Para conjuntos de datos Banana y Titanic se ha tomado una muestra de los originales. Los tamaños de los conjuntos de datos originales son de 5300 y 2201 instancias, respectivamente.



## IMPLEMENTACIÓN

---

En este capítulo se abordarán los aspectos más técnicos de este trabajo. Se enumerarán las librerías y herramientas empleadas, y se describirá la estructura que sigue el software implementado, el cual ha sido pensado para reutilizar la mayor cantidad de código posible.

Antes de nada, hay que tener en cuenta que no se trata de un proyecto de software convencional, como podría serlo el desarrollo de una aplicación o de un servicio web. Este es un trabajo de experimentación con algoritmos, con un fuerte componente teórico en lo que se refiere a ciencia de datos y algoritmos de optimización. Por ello, no se ha seguido con rigurosidad los principios de ingeniería del software durante el diseño y desarrollo de este trabajo. Aún así se dedicará un espacio para hablar, aunque sea a un nivel un tanto superficial, de la implementación y la estructura del código.

### 7.1 LIBRERÍAS UTILIZADAS

Numpy  
Scikit-Learn  
Parallel (Multiprocessing?)  
jMetal

### 7.2 ESTRUCTURA DEL SOFTWARE





## RESULTADOS

---

En este apartado se expondrán todos los resultados obtenidos . También se elaborará un análisis estadístico haciendo uso de Estadística Bayesiana No Paramétrica (NPBS).

### 8.1 TABLAS DE RESULTADOS Y PRIMERAS OBSERVACIONES

Tal y como se vió en el capítulo 6, tendremos 4 métricas de rendimiento en total, 2 a nivel de partición de clustering y 2 a nivel de frente de Pareto. Tendremos una tabla por cada métrica y, a su vez, por cada conjunto de restricciones (10 %, 15 % y 20 %). Por tanto, tendremos en total 12 tablas de resultados, más alguna tabla de resumen con los promedios.

Tabla 8.1: Mejores puntuaciones ARI - 10 % restricciones

Dataset	MOEAD-ADA	MOEAD-ADA <sub>Norm</sub>	MOEAD-ADA-PD	MOEAD-ADA-PD-SSFS	MOEAD-ADA-PD <sub>NormA</sub>	MOEAD-ADA-PD <sub>NormB</sub>	MOEAD-ADA-KM	MOEAD-ADA-PD-KM
Appendicitis	0.322 ± 0.069	0.194 ± 0.046	0.413 ± 0.172	0.332 ± 0.212	0.353 ± 0.054	0.346 ± 0.081	0.355 ± 0.052	0.641 ± 0.032
Balance	0.057 ± 0.020	0.018 ± 0.005	0.154 ± 0.086	0.151 ± 0.070	0.061 ± 0.065	0.051 ± 0.061	0.179 ± 0.015	0.182 ± 0.020
Banana (muestra)	0.016 ± 0.008	0.042 ± 0.059	0.062 ± 0.025	0.061 ± 0.034	0.118 ± 0.064	0.130 ± 0.052	0.047 ± 0.014	0.168 ± 0.029
Bupa	0.033 ± 0.012	0.011 ± 0.001	0.025 ± 0.006	0.028 ± 0.016	0.012 ± 0.004	0.010 ± 0.002	0.024 ± 0.003	0.026 ± 0.006
Ecoli	0.175 ± 0.121	0.028 ± 0.008	0.122 ± 0.030	0.168 ± 0.064	0.039 ± 0.009	0.031 ± 0.018	0.323 ± 0.014	0.338 ± 0.025
Glass	0.135 ± 0.051	0.051 ± 0.009	0.125 ± 0.046	0.140 ± 0.041	0.068 ± 0.027	0.062 ± 0.020	0.225 ± 0.008	0.229 ± 0.009
Haberman	0.088 ± 0.016	0.130 ± 0.133	0.182 ± 0.107	0.069 ± 0.046	0.193 ± 0.128	0.296 ± 0.087	0.030 ± 0.021	0.011 ± 0.002
Hayes Roth	0.077 ± 0.025	0.051 ± 0.017	0.034 ± 0.022	0.034 ± 0.018	0.051 ± 0.013	0.047 ± 0.020	0.064 ± 0.023	0.071 ± 0.010
Heart	0.473 ± 0.173	0.092 ± 0.023	0.282 ± 0.083	0.195 ± 0.160	0.077 ± 0.018	0.073 ± 0.012	0.446 ± 0.053	0.463 ± 0.019
Iris	0.105 ± 0.025	0.082 ± 0.023	0.211 ± 0.118	0.188 ± 0.070	0.280 ± 0.135	0.214 ± 0.092	0.673 ± 0.017	0.690 ± 0.009
Led7Digit	0.035 ± 0.026	0.022 ± 0.009	0.092 ± 0.002	0.088 ± 0.006	0.017 ± 0.004	0.021 ± 0.004	0.422 ± 0.008	0.418 ± 0.004
Monk2	0.202 ± 0.213	0.039 ± 0.012	0.215 ± 0.091	0.241 ± 0.084	0.067 ± 0.037	0.054 ± 0.029	0.020 ± 0.024	0.018 ± 0.009
Newthyroid	0.345 ± 0.182	0.089 ± 0.042	0.315 ± 0.136	0.168 ± 0.066	0.236 ± 0.137	0.208 ± 0.070	0.613 ± 0.014	0.618 ± 0.018
Pima	0.205 ± 0.130	0.023 ± 0.006	0.329 ± 0.010	0.310 ± 0.041	0.028 ± 0.009	0.035 ± 0.027	0.180 ± 0.037	0.324 ± 0.037
Saheart	0.202 ± 0.156	0.029 ± 0.013	0.310 ± 0.049	0.270 ± 0.080	0.056 ± 0.014	0.055 ± 0.015	0.190 ± 0.044	0.251 ± 0.046
Soybean	0.702 ± 0.042	0.361 ± 0.026	0.383 ± 0.086	0.525 ± 0.173	0.219 ± 0.038	0.244 ± 0.048	0.991 ± 0.019	1.000 ± 0.000
Tae	0.074 ± 0.022	0.052 ± 0.011	0.091 ± 0.034	0.050 ± 0.011	0.069 ± 0.024	0.062 ± 0.032	0.115 ± 0.020	0.105 ± 0.014
Titanic (muestra)	0.054 ± 0.079	0.026 ± 0.013	0.234 ± 0.105	0.283 ± 0.038	0.180 ± 0.055	0.254 ± 0.041	0.280 ± 0.030	0.299 ± 0.016
Wine	0.511 ± 0.120	0.104 ± 0.017	0.413 ± 0.193	0.354 ± 0.141	0.081 ± 0.033	0.068 ± 0.010	0.804 ± 0.005	0.817 ± 0.016
Zoo	0.480 ± 0.108	0.268 ± 0.093	0.512 ± 0.141	0.507 ± 0.158	0.168 ± 0.053	0.177 ± 0.062	0.736 ± 0.070	0.753 ± 0.038
Promedio	0.215 ± 0.190	0.086 ± 0.089	0.225 ± 0.137	0.208 ± 0.142	0.119 ± 0.094	0.122 ± 0.101	0.336 ± 0.283	0.371 ± 0.286

Tabla 8.2: Mejores puntuaciones ARI - 15 % restricciones

Dataset	MOEAD-ADA	MOEAD-ADA <sub>Norm</sub>	MOEAD-ADA-PD	MOEAD-ADA-PD-SSFS	MOEAD-ADA-PD <sub>NormA</sub>	MOEAD-ADA-PD <sub>NormB</sub>	MOEAD-ADA-KM	MOEAD-ADA-PD-KM
Appendicitis	0.278 ± 0.069	0.196 ± 0.030	0.433 ± 0.201	0.491 ± 0.183	0.502 ± 0.046	0.424 ± 0.079	0.443 ± 0.083	0.675 ± 0.044
Balance	0.326 ± 0.051	0.019 ± 0.006	0.313 ± 0.054	0.325 ± 0.035	0.103 ± 0.086	0.089 ± 0.073	0.188 ± 0.015	0.274 ± 0.040
Banana (muestra)	0.070 ± 0.054	0.021 ± 0.010	0.232 ± 0.030	0.225 ± 0.040	0.168 ± 0.026	0.155 ± 0.053	0.057 ± 0.012	0.227 ± 0.038
Bupa	0.026 ± 0.003	0.015 ± 0.001	0.029 ± 0.012	0.026 ± 0.014	0.014 ± 0.004	0.012 ± 0.001	0.025 ± 0.010	0.027 ± 0.005
Ecoli	0.168 ± 0.121	0.037 ± 0.018	0.194 ± 0.097	0.219 ± 0.100	0.035 ± 0.010	0.025 ± 0.006	0.331 ± 0.014	0.338 ± 0.011
Glass	0.143 ± 0.052	0.079 ± 0.022	0.165 ± 0.028	0.148 ± 0.034	0.079 ± 0.026	0.092 ± 0.050	0.228 ± 0.005	0.227 ± 0.008
Haberman	0.190 ± 0.133	0.216 ± 0.154	0.393 ± 0.139	0.498 ± 0.181	0.367 ± 0.069	0.371 ± 0.077	0.060 ± 0.026	0.516 ± 0.018
Hayes Roth	0.084 ± 0.036	0.062 ± 0.010	0.131 ± 0.037	0.131 ± 0.028	0.088 ± 0.021	0.091 ± 0.014	0.084 ± 0.052	0.090 ± 0.016
Heart	0.609 ± 0.248	0.126 ± 0.034	0.625 ± 0.019	0.626 ± 0.060	0.134 ± 0.043	0.105 ± 0.016	0.440 ± 0.051	0.630 ± 0.029
Iris	0.090 ± 0.008	0.105 ± 0.015	0.380 ± 0.050	0.352 ± 0.147	0.376 ± 0.101	0.341 ± 0.144	0.694 ± 0.014	0.698 ± 0.017
Led7Digit	0.033 ± 0.024	0.019 ± 0.006	0.093 ± 0.004	0.090 ± 0.005	0.021 ± 0.005	0.020 ± 0.006	0.412 ± 0.010	0.421 ± 0.003
Monk2	0.398 ± 0.277	0.050 ± 0.016	0.540 ± 0.034	0.555 ± 0.056	0.132 ± 0.042	0.113 ± 0.033	0.086 ± 0.102	0.559 ± 0.028
Newthyroid	0.450 ± 0.224	0.116 ± 0.029	0.609 ± 0.026	0.594 ± 0.085	0.187 ± 0.068	0.269 ± 0.106	0.633 ± 0.010	0.651 ± 0.019
Pima	0.298 ± 0.191	0.035 ± 0.014	0.413 ± 0.032	0.434 ± 0.044	0.052 ± 0.012	0.038 ± 0.016	0.233 ± 0.089	0.407 ± 0.045
Saheart	0.318 ± 0.151	0.052 ± 0.008	0.526 ± 0.040	0.567 ± 0.055	0.110 ± 0.055	0.099 ± 0.038	0.222 ± 0.092	0.491 ± 0.072
Soybean	0.837 ± 0.093	0.504 ± 0.056	0.637 ± 0.206	0.511 ± 0.161	0.299 ± 0.024	0.287 ± 0.048	0.981 ± 0.023	1.000 ± 0.000
Tae	0.142 ± 0.071	0.063 ± 0.012	0.087 ± 0.024	0.098 ± 0.040	0.053 ± 0.015	0.054 ± 0.012	0.096 ± 0.012	0.092 ± 0.013
Titanic (muestra)	0.181 ± 0.145	0.035 ± 0.015	0.445 ± 0.021	0.447 ± 0.019	0.212 ± 0.111	0.251 ± 0.081	0.256 ± 0.028	0.418 ± 0.036
Wine	0.595 ± 0.126	0.127 ± 0.026	0.604 ± 0.074	0.508 ± 0.067	0.113 ± 0.016	0.102 ± 0.026	0.815 ± 0.014	0.819 ± 0.011
Zoo	0.666 ± 0.058	0.509 ± 0.053	0.579 ± 0.101	0.517 ± 0.154	0.191 ± 0.029	0.257 ± 0.042	0.749 ± 0.051	0.765 ± 0.029
Promedio	0.295 ± 0.226	0.119 ± 0.140	0.372 ± 0.198	0.368 ± 0.189	0.162 ± 0.129	0.160 ± 0.124	0.352 ± 0.279	0.466 ± 0.257

Tabla 8.3: Mejores puntuaciones ARI - 20 % restricciones

Dataset	MOEAD-ADA	MOEAD-ADA <sub>Norm</sub>	MOEAD-ADA-PD	MOEAD-ADA-PD-SSFS	MOEAD-ADA-PD <sub>NormA</sub>	MOEAD-ADA-PD <sub>NormB</sub>	MOEAD-ADA-KM	MOEAD-ADA-PD-KM
Appendicitis	0.408 ± 0.081	0.323 ± 0.086	0.807 ± 0.055	0.812 ± 0.110	0.522 ± 0.055	0.530 ± 0.073	0.529 ± 0.055	0.771 ± 0.090
Balance	0.163 ± 0.143	0.021 ± 0.001	0.374 ± 0.016	0.395 ± 0.009	0.118 ± 0.077	0.133 ± 0.064	0.192 ± 0.010	0.310 ± 0.035
Banana (muestra)	0.015 ± 0.004	0.073 ± 0.077	0.267 ± 0.016	0.283 ± 0.011	0.174 ± 0.028	0.160 ± 0.032	0.046 ± 0.014	0.244 ± 0.010
Bupa	0.031 ± 0.007	0.014 ± 0.002	0.034 ± 0.011	0.030 ± 0.012	0.013 ± 0.003	0.016 ± 0.003	0.070 ± 0.087	0.027 ± 0.003
Ecoli	0.232 ± 0.143	0.035 ± 0.008	0.413 ± 0.057	0.410 ± 0.058	0.028 ± 0.005	0.027 ± 0.005	0.337 ± 0.018	0.333 ± 0.021
Glass	0.244 ± 0.093	0.076 ± 0.017	0.230 ± 0.054	0.228 ± 0.085	0.093 ± 0.030	0.094 ± 0.045	0.233 ± 0.011	0.221 ± 0.006
Haberman	0.338 ± 0.178	0.209 ± 0.149	0.639 ± 0.019	0.709 ± 0.029	0.429 ± 0.111	0.343 ± 0.090	0.115 ± 0.051	0.623 ± 0.089
Hayes Roth	0.214 ± 0.078	0.102 ± 0.034	0.235 ± 0.070	0.234 ± 0.035	0.133 ± 0.026	0.125 ± 0.032	0.111 ± 0.023	0.170 ± 0.039
Heart	0.565 ± 0.250	0.132 ± 0.013	0.743 ± 0.015	0.756 ± 0.036	0.159 ± 0.037	0.160 ± 0.034	0.508 ± 0.063	0.625 ± 0.086
Iris	0.208 ± 0.177	0.155 ± 0.107	0.657 ± 0.108	0.654 ± 0.140	0.393 ± 0.135	0.367 ± 0.119	0.695 ± 0.014	0.730 ± 0.028
Led7Digit	0.035 ± 0.034	0.020 ± 0.005	0.088 ± 0.008	0.096 ± 0.012	0.020 ± 0.003	0.017 ± 0.003	0.416 ± 0.010	0.419 ± 0.008
Monk2	0.523 ± 0.183	0.065 ± 0.029	0.630 ± 0.061	0.640 ± 0.050	0.173 ± 0.015	0.209 ± 0.041	0.254 ± 0.227	0.579 ± 0.051
Newthyroid	0.451 ± 0.217	0.156 ± 0.050	0.673 ± 0.020	0.715 ± 0.012	0.313 ± 0.153	0.293 ± 0.112	0.669 ± 0.018	0.678 ± 0.012
Pima	0.176 ± 0.098	0.055 ± 0.004	0.498 ± 0.011	0.503 ± 0.016	0.076 ± 0.026	0.059 ± 0.026	0.190 ± 0.038	0.438 ± 0.006
Saheart	0.278 ± 0.189	0.069 ± 0.012	0.591 ± 0.057	0.597 ± 0.050	0.157 ± 0.027	0.176 ± 0.104	0.190 ± 0.045	0.571 ± 0.039
Soybean	0.929 ± 0.024	0.596 ± 0.080	0.633 ± 0.117	0.652 ± 0.118	0.304 ± 0.060	0.351 ± 0.075	0.978 ± 0.027	1.000 ± 0.000
Tae	0.316 ± 0.099	0.089 ± 0.028	0.326 ± 0.052	0.343 ± 0.020	0.146 ± 0.084	0.146 ± 0.083	0.185 ± 0.080	0.296 ± 0.051
Titanic (muestra)	0.447 ± 0.135	0.039 ± 0.015	0.538 ± 0.013	0.546 ± 0.012	0.242 ± 0.117	0.244 ± 0.097	0.268 ± 0.036	0.444 ± 0.061
Wine	0.614 ± 0.170	0.159 ± 0.083	0.716 ± 0.027	0.696 ± 0.084	0.125 ± 0.021	0.099 ± 0.018	0.828 ± 0.026	0.810 ± 0.015
Zoo	0.684 ± 0.065	0.533 ± 0.056	0.514 ± 0.144	0.478 ± 0.194	0.270 ± 0.102	0.197 ± 0.032	0.765 ± 0.046	0.765 ± 0.023
Promedio	0.344 ± 0.231	0.146 ± 0.158	0.480 ± 0.218	0.489 ± 0.222	0.194 ± 0.136	0.187 ± 0.131	0.379 ± 0.272	0.503 ± 0.246

Tabla 8.4: Restricciones no satisfechas (UNSAT) - 10 % restricciones

Dataset	MOEAD-ADA	MOEAD-ADA <sub>Norm</sub>	MOEAD-ADA-PD	MOEAD-ADA-PD-SSFS	MOEAD-ADA-PD <sub>NormA</sub>	MOEAD-ADA-PD <sub>NormB</sub>	MOEAD-ADA-KM	MOEAD-ADA-PD-KM
Appendicitis	0.091 ± 0.026	0.113 ± 0.014	0.120 ± 0.050	0.142 ± 0.040	0.127 ± 0.030	0.131 ± 0.014	0.105 ± 0.007	0.084 ± 0.019
Balance	0.401 ± 0.020	0.421 ± 0.005	0.368 ± 0.031	0.371 ± 0.030	0.406 ± 0.025	0.411 ± 0.022	0.354 ± 0.006	0.351 ± 0.006
Banana (muestra)	0.473 ± 0.008	0.460 ± 0.029	0.449 ± 0.010	0.451 ± 0.012	0.426 ± 0.029	0.423 ± 0.028	0.457 ± 0.009	0.397 ± 0.014
Bupa	0.112 ± 0.020	0.146 ± 0.004	0.161 ± 0.001	0.156 ± 0.003	0.152 ± 0.002	0.152 ± 0.003	0.141 ± 0.014	0.154 ± 0.002
Ecoli	0.182 ± 0.017	0.246 ± 0.005	0.255 ± 0.011	0.237 ± 0.021	0.257 ± 0.003	0.259 ± 0.002	0.190 ± 0.009	0.208 ± 0.004
Glass	0.119 ± 0.017	0.202 ± 0.007	0.209 ± 0.025	0.208 ± 0.016	0.211 ± 0.003	0.214 ± 0.004	0.158 ± 0.010	0.210 ± 0.009
Haberman	0.358 ± 0.009	0.328 ± 0.077	0.305 ± 0.041	0.366 ± 0.018	0.292 ± 0.056	0.249 ± 0.030	0.363 ± 0.014	0.342 ± 0.009
Hayes Roth	0.197 ± 0.019	0.223 ± 0.016	0.250 ± 0.005	0.262 ± 0.014	0.233 ± 0.019	0.235 ± 0.026	0.192 ± 0.020	0.223 ± 0.021
Heart	0.170 ± 0.092	0.338 ± 0.009	0.265 ± 0.032	0.280 ± 0.036	0.357 ± 0.008	0.355 ± 0.006	0.251 ± 0.060	0.203 ± 0.011
Iris	0.149 ± 0.013	0.166 ± 0.020	0.208 ± 0.020	0.210 ± 0.022	0.145 ± 0.034	0.168 ± 0.030	0.086 ± 0.006	0.067 ± 0.010
Led7Digit	0.111 ± 0.008	0.140 ± 0.002	0.144 ± 0.004	0.138 ± 0.002	0.148 ± 0.001	0.148 ± 0.002	0.084 ± 0.003	0.089 ± 0.002
Monk2	0.326 ± 0.103	0.410 ± 0.007	0.321 ± 0.026	0.311 ± 0.032	0.404 ± 0.013	0.408 ± 0.012	0.393 ± 0.018	0.374 ± 0.007
Newthyroid	0.190 ± 0.038	0.328 ± 0.003	0.257 ± 0.046	0.314 ± 0.040	0.298 ± 0.055	0.299 ± 0.023	0.139 ± 0.020	0.130 ± 0.008
Pima	0.346 ± 0.065	0.447 ± 0.004	0.302 ± 0.005	0.302 ± 0.013	0.440 ± 0.007	0.445 ± 0.008	0.374 ± 0.024	0.305 ± 0.016
Saheart	0.337 ± 0.070	0.410 ± 0.003	0.292 ± 0.018	0.295 ± 0.030	0.409 ± 0.010	0.409 ± 0.009	0.339 ± 0.037	0.312 ± 0.020
Soybean	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
Tae	0.138 ± 0.019	0.193 ± 0.006	0.202 ± 0.048	0.213 ± 0.044	0.217 ± 0.012	0.208 ± 0.011	0.165 ± 0.030	0.193 ± 0.016
Titanic (muestra)	0.424 ± 0.040	0.445 ± 0.008	0.335 ± 0.042	0.315 ± 0.019	0.371 ± 0.027	0.336 ± 0.027	0.316 ± 0.021	0.311 ± 0.011
Wine	0.094 ± 0.015	0.226 ± 0.016	0.142 ± 0.042	0.157 ± 0.053	0.278 ± 0.011	0.268 ± 0.011	0.043 ± 0.011	0.051 ± 0.005
Zoo	0.000 ± 0.000	0.040 ± 0.014	0.051 ± 0.029	0.058 ± 0.027	0.084 ± 0.019	0.095 ± 0.014	0.004 ± 0.007	0.015 ± 0.007
Promedio	0.211 ± 0.137	0.264 ± 0.138	0.232 ± 0.106	0.239 ± 0.107	0.263 ± 0.124	0.261 ± 0.121	0.208 ± 0.136	0.201 ± 0.122

Tabla 8.5: Restricciones no satisfechas (UNSAT) - 15 % restricciones

Dataset	MOEAD-ADA	MOEAD-ADA <sub>Norm</sub>	MOEAD-ADA-PD	MOEAD-ADA-PD-SSFS	MOEAD-ADA-PD <sub>NormA</sub>	MOEAD-ADA-PD <sub>NormB</sub>	MOEAD-ADA-KM	MOEAD-ADA-PD-KM
Appendicitis	0.207 ± 0.024	0.243 ± 0.022	0.148 ± 0.064	0.128 ± 0.065	0.143 ± 0.031	0.168 ± 0.024	0.173 ± 0.021	0.073 ± 0.018
Balance	0.293 ± 0.030	0.437 ± 0.003	0.298 ± 0.019	0.288 ± 0.015	0.397 ± 0.035	0.414 ± 0.029	0.368 ± 0.012	0.320 ± 0.020
Banana (muestra)	0.457 ± 0.027	0.478 ± 0.005	0.375 ± 0.016	0.381 ± 0.021	0.409 ± 0.012	0.417 ± 0.025	0.464 ± 0.007	0.379 ± 0.019
Bupa	0.142 ± 0.019	0.171 ± 0.002	0.180 ± 0.002	0.180 ± 0.002	0.175 ± 0.001	0.175 ± 0.003	0.153 ± 0.020	0.177 ± 0.001
Ecoli	0.244 ± 0.035	0.279 ± 0.004	0.255 ± 0.030	0.252 ± 0.034	0.287 ± 0.001	0.287 ± 0.003	0.207 ± 0.002	0.208 ± 0.004
Glass	0.206 ± 0.062	0.250 ± 0.010	0.229 ± 0.009	0.241 ± 0.007	0.259 ± 0.008	0.256 ± 0.010	0.207 ± 0.011	0.222 ± 0.011
Haberman	0.345 ± 0.058	0.346 ± 0.076	0.251 ± 0.071	0.203 ± 0.088	0.259 ± 0.033	0.262 ± 0.034	0.401 ± 0.013	0.191 ± 0.014
Hayes Roth	0.260 ± 0.016	0.284 ± 0.008	0.241 ± 0.016	0.225 ± 0.018	0.278 ± 0.009	0.272 ± 0.017	0.255 ± 0.033	0.264 ± 0.009
Heart	0.169 ± 0.109	0.389 ± 0.010	0.161 ± 0.007	0.157 ± 0.025	0.385 ± 0.015	0.389 ± 0.017	0.264 ± 0.036	0.155 ± 0.014
Iris	0.269 ± 0.016	0.278 ± 0.012	0.194 ± 0.022	0.215 ± 0.055	0.223 ± 0.028	0.217 ± 0.062	0.123 ± 0.008	0.110 ± 0.008
Led7Digit	0.121 ± 0.003	0.143 ± 0.003	0.144 ± 0.002	0.143 ± 0.002	0.148 ± 0.001	0.147 ± 0.001	0.092 ± 0.003	0.093 ± 0.001
Monk2	0.276 ± 0.126	0.437 ± 0.010	0.207 ± 0.017	0.186 ± 0.021	0.400 ± 0.021	0.411 ± 0.015	0.413 ± 0.040	0.194 ± 0.014
Newthyroid	0.214 ± 0.120	0.370 ± 0.012	0.164 ± 0.016	0.161 ± 0.046	0.352 ± 0.028	0.334 ± 0.041	0.156 ± 0.005	0.149 ± 0.005
Pima	0.332 ± 0.094	0.461 ± 0.003	0.275 ± 0.015	0.268 ± 0.021	0.450 ± 0.006	0.460 ± 0.008	0.362 ± 0.043	0.277 ± 0.020
Saheart	0.307 ± 0.075	0.437 ± 0.004	0.205 ± 0.015	0.188 ± 0.023	0.413 ± 0.026	0.419 ± 0.019	0.359 ± 0.054	0.227 ± 0.033
Soybean	0.000 ± 0.000	0.000 ± 0.000	0.014 ± 0.017	0.029 ± 0.014	0.050 ± 0.029	0.050 ± 0.017	0.000 ± 0.000	0.000 ± 0.000
Tae	0.223 ± 0.020	0.278 ± 0.011	0.247 ± 0.014	0.251 ± 0.020	0.290 ± 0.011	0.283 ± 0.020	0.247 ± 0.022	0.254 ± 0.012
Titanic (muestra)	0.386 ± 0.070	0.461 ± 0.008	0.258 ± 0.009	0.258 ± 0.012	0.370 ± 0.051	0.353 ± 0.041	0.346 ± 0.011	0.273 ± 0.021
Wine	0.112 ± 0.040	0.286 ± 0.013	0.130 ± 0.032	0.161 ± 0.022	0.308 ± 0.008	0.309 ± 0.002	0.056 ± 0.005	0.058 ± 0.006
Zoo	0.017 ± 0.007	0.063 ± 0.014	0.070 ± 0.026	0.102 ± 0.058	0.150 ± 0.009	0.143 ± 0.014	0.022 ± 0.007	0.040 ± 0.011
Promedio	0.229 ± 0.112	0.305 ± 0.132	0.202 ± 0.079	0.201 ± 0.074	0.287 ± 0.109	0.288 ± 0.110	0.233 ± 0.134	0.183 ± 0.097

Tabla 8.6: Restricciones no satisfechas (UNSAT) - 20 % restricciones

Dataset	MOEAD-ADA	MOEAD-ADA <sub>Norm</sub>	MOEAD-ADA-PD	MOEAD-ADA-PD-SSFS	MOEAD-ADA-PD <sub>NormA</sub>	MOEAD-ADA-PD <sub>NormB</sub>	MOEAD-ADA-KM	MOEAD-ADA-PD-KM
Appendicitis	0.203 ± 0.043	0.252 ± 0.046	0.060 ± 0.019	0.059 ± 0.041	0.180 ± 0.028	0.174 ± 0.029	0.166 ± 0.039	0.070 ± 0.029
Balance	0.388 ± 0.070	0.448 ± 0.003	0.283 ± 0.005	0.277 ± 0.005	0.403 ± 0.033	0.405 ± 0.027	0.380 ± 0.008	0.313 ± 0.012
Banana (muestra)	0.488 ± 0.002	0.459 ± 0.038	0.361 ± 0.006	0.354 ± 0.005	0.409 ± 0.014	0.417 ± 0.017	0.472 ± 0.008	0.373 ± 0.005
Bupa	0.155 ± 0.014	0.175 ± 0.003	0.180 ± 0.004	0.182 ± 0.003	0.179 ± 0.001	0.180 ± 0.000	0.162 ± 0.015	0.175 ± 0.002
Ecoli	0.248 ± 0.042	0.299 ± 0.003	0.202 ± 0.017	0.197 ± 0.020	0.303 ± 0.003	0.303 ± 0.004	0.218 ± 0.004	0.219 ± 0.010
Glass	0.218 ± 0.058	0.268 ± 0.007	0.216 ± 0.015	0.216 ± 0.021	0.274 ± 0.009	0.271 ± 0.008	0.239 ± 0.018	0.226 ± 0.004
Haberman	0.293 ± 0.083	0.365 ± 0.079	0.153 ± 0.011	0.122 ± 0.012	0.260 ± 0.057	0.296 ± 0.043	0.402 ± 0.023	0.165 ± 0.044
Hayes Roth	0.268 ± 0.045	0.323 ± 0.013	0.250 ± 0.014	0.236 ± 0.010	0.302 ± 0.021	0.312 ± 0.018	0.314 ± 0.022	0.281 ± 0.025
Heart	0.193 ± 0.116	0.400 ± 0.009	0.115 ± 0.011	0.097 ± 0.015	0.382 ± 0.016	0.393 ± 0.012	0.233 ± 0.040	0.163 ± 0.040
Iris	0.246 ± 0.059	0.288 ± 0.037	0.119 ± 0.032	0.107 ± 0.051	0.218 ± 0.054	0.234 ± 0.049	0.130 ± 0.011	0.100 ± 0.010
Led7Digit	0.141 ± 0.007	0.155 ± 0.001	0.149 ± 0.003	0.149 ± 0.005	0.159 ± 0.000	0.160 ± 0.001	0.096 ± 0.002	0.098 ± 0.001
Monk2	0.223 ± 0.088	0.446 ± 0.015	0.170 ± 0.026	0.163 ± 0.022	0.396 ± 0.010	0.380 ± 0.023	0.354 ± 0.106	0.196 ± 0.024
Newthyroid	0.258 ± 0.113	0.396 ± 0.022	0.139 ± 0.004	0.115 ± 0.004	0.318 ± 0.059	0.340 ± 0.050	0.138 ± 0.006	0.136 ± 0.004
Pima	0.399 ± 0.047	0.464 ± 0.002	0.239 ± 0.005	0.239 ± 0.007	0.451 ± 0.010	0.457 ± 0.014	0.391 ± 0.018	0.270 ± 0.003
Saheart	0.346 ± 0.097	0.444 ± 0.008	0.189 ± 0.028	0.185 ± 0.025	0.404 ± 0.014	0.392 ± 0.050	0.388 ± 0.024	0.199 ± 0.021
Soybean	0.000 ± 0.000	0.009 ± 0.011	0.018 ± 0.009	0.004 ± 0.009	0.058 ± 0.011	0.067 ± 0.014	0.000 ± 0.000	0.000 ± 0.000
Tae	0.206 ± 0.024	0.325 ± 0.016	0.205 ± 0.021	0.200 ± 0.007	0.302 ± 0.029	0.312 ± 0.034	0.293 ± 0.031	0.229 ± 0.009
Titanic (muestra)	0.263 ± 0.063	0.468 ± 0.007	0.219 ± 0.004	0.215 ± 0.006	0.364 ± 0.054	0.365 ± 0.047	0.352 ± 0.016	0.262 ± 0.028
Wine	0.144 ± 0.078	0.312 ± 0.027	0.100 ± 0.013	0.096 ± 0.028	0.342 ± 0.006	0.346 ± 0.003	0.057 ± 0.006	0.069 ± 0.006
Zoo	0.060 ± 0.030	0.107 ± 0.009	0.094 ± 0.037	0.111 ± 0.051	0.182 ± 0.010	0.186 ± 0.005	0.049 ± 0.011	0.060 ± 0.011
Promedio	0.237 ± 0.111	0.320 ± 0.127	0.173 ± 0.078	0.166 ± 0.079	0.294 ± 0.102	0.299 ± 0.101	0.242 ± 0.134	0.180 ± 0.093

Tabla 8.7: Hipervolumenes - 10 % restricciones

Dataset	MOEAD-ADA	MOEAD-ADA <sub>Norm</sub>	MOEAD-ADA-PD	MOEAD-ADA-PD-SSPS	MOEAD-ADA-PD <sub>NormA</sub>	MOEAD-ADA-PD <sub>NormB</sub>	MOEAD-ADA-KM	MOEAD-ADA-PD-KM
Appendicitis	3853,827 ± 4089,804	26605,769 ± 2121,138	62562,791 ± 16082,039	60446,777 ± 15516,287	50752,500 ± 5833,849	48925,257 ± 4132,670	55849,779 ± 8357,798	76490,315 ± 1957,848
Balance	4278717,324 ± 1335213,534	923946,241 ± 64540,603	6950488,352 ± 2269891,261	6730268,715 ± 1709680,588	2302395,104 ± 1544993,231	1921000,945 ± 1272150,153	16845229,385 ± 509655,531	16486197,586 ± 491180,724
Banana (muestra)	39090237,405 ± 8530426,836	49479602,093 ± 53138958,313	45231130,987 ± 17526259,809	42715865,161 ± 2118539,352	10089849,182 ± 65493120,351	145796547,855 ± 78311594,730	358109489,111 ± 31421527,736	370919140,095 ± 39735494,272
Bupa	74879,456 ± 53112,542	23058,739 ± 1699,539	128102,331 ± 6264,614	126407,383 ± 18827,653	25017,465 ± 1166,344	26397,419 ± 4172,768	295262,063 ± 119446,456	374048,219 ± 6627,369
Ecoli	297827,791 ± 175862,393	51104,302 ± 3991,352	342466,622 ± 51348,920	412713,407 ± 68072,857	65905,147 ± 14704,250	61036,762 ± 19490,741	1048288,467 ± 28414,805	1040816,128 ± 36151,861
Class	128543,346 ± 46272,019	42552,726 ± 7308,786	175122,696 ± 24518,868	173137,861 ± 15517,296	51608,196 ± 10968,814	49318,516 ± 6890,575	290687,125 ± 12071,932	273044,012 ± 9761,595
Haberman	691617,272 ± 157395,077	1044979,970 ± 1012688,154	2021281,485 ± 651605,961	1150084,665 ± 180395,643	1962847,254 ± 879568,928	2380846,435 ± 683379,072	1990372,594 ± 159732,446	2151691,252 ± 95295,269
Hayes Roth	88814,761 ± 23922,797	67305,602 ± 11614,603	185584,103 ± 13567,700	175522,903 ± 23191,289	122231,610 ± 13292,650	126549,943 ± 29609,508	248348,556 ± 12589,478	262820,374 ± 17158,483
Heart	763892,119 ± 287546,428	167498,095 ± 30133,374	673016,614 ± 86545,316	620970,339 ± 136301,302	162438,564 ± 18183,274	155641,164 ± 17687,534	977093,873 ± 153125,213	1062182,573 ± 28671,832
Iris	29472,137 ± 20675,769	23381,344 ± 23608,670	62662,691 ± 125765,306	589551,605 ± 109701,756	630486,987 ± 199235,636	542108,941 ± 142752,200	1242968,834 ± 17654,854	1266141,620 ± 21855,139
LeafDigit	298167,795 ± 243310,464	107377,722 ± 31164,657	705053,865 ± 70706,911	735428,460 ± 50880,280	109695,485 ± 18595,297	111259,579 ± 24322,838	3883177,448 ± 74495,544	3764835,103 ± 79628,658
Monks	2999551,136 ± 1005651,158	450425,513 ± 12453,477	2728877,925 ± 312970,414	284851,319 ± 439706,469	703275,735 ± 331273,124	599212,857 ± 43700,234	310721,297 ± 344175,949	2989456,581 ± 32244,166
Newthyroid	856643,585 ± 463753,331	209586,436 ± 48615,544	999580,901 ± 210572,690	778302,601 ± 145095,818	539413,910 ± 267289,423	453158,798 ± 129217,233	1984519,823 ± 97548,398	2088761,751 ± 30226,703
Pima	6738136,522 ± 5240483,883	614340,752 ± 97067,927	7943116,892 ± 1119977,086	700797,505 ± 1543900,266	690266,160 ± 119908,447	734463,332 ± 188991,978	13691517,459 ± 1436616,787	15550463,078 ± 706266,640
Saheart	3919167,713 ± 860784,569	523693,492 ± 44919,884	4356923,745 ± 225396,049	3920389,641 ± 661182,013	823828,798 ± 173493,431	852912,363 ± 149350,062	5846318,308 ± 659084,763	6054219,145 ± 318970,222
Soybean	725577 ± 31,094	408445 ± 79,517	647190 ± 23,059	652453 ± 44,662	336740 ± 18,853	292,338 ± 36,322	754,952 ± 12,785	758,262 ± 2,443
Tae	105111,797 ± 35300,754	62564,071 ± 5344,807	215704,092 ± 29603,166	2082177,78 ± 24163,409	95172,740 ± 3498,310	88929,010 ± 9086,372	242825,445 ± 21763,732	242178,143 ± 8300,007
Titanic (muestra)	21218011,201 ± 972775,555	7633109,450 ± 1440231,952	49207771,177 ± 12982679,474	56492773,691 ± 4499511,535	29294375,399 ± 7578031,828	41367920,588 ± 8174051,656	103466232,442 ± 5811359,375	103591610,472 ± 3376627,156
Wine	267893,673 ± 24968,806	56178,441 ± 7362,281	233567,904 ± 44276,057	236768,326 ± 44441,921	61779,742 ± 7056,269	62535,146 ± 4870,066	367657,401 ± 5936,019	364582,736 ± 6682,050
Zoo	15603,273 ± 2819,450	8207,487 ± 842,804	19925,890 ± 3605,519	19827,761 ± 1390,594	6230,072 ± 560,331	5768,507 ± 791,398	24553,778 ± 1163,725	24545,778 ± 900,022
Promedio	3695868,335 ± 781475,683	30862996,335 ± 10767883,726	6139951,913 ± 13888842,079	6249850,019 ± 14758584,623	6929245,340 ± 22458815,306	9769241,298 ± 32458230,304	25686921,328 ± 79465212,226	26418219,625 ± 82138340,316

Tabla 8.8: Hipervolumenes - 15 % restricciones

Dataset	MOEAD-ADA	MOEAD-ADA <sub>Norm</sub>	MOEAD-ADA-PD	MOEAD-ADA-PD-SSPS	MOEAD-ADA-PD <sub>NormA</sub>	MOEAD-ADA-PD <sub>NormB</sub>	MOEAD-ADA-KM	MOEAD-ADA-PD-KM
Appendicitis	43054,441 ± 6369,162	27871,795 ± 2252,715	84735,762 ± 28398,815	96869,665 ± 32657,111	79361,514 ± 14085,932	70378,210 ± 7387,335	82104,132 ± 13062,380	120383,640 ± 6642,148
Balance	29157322,252 ± 4065556,818	1754670,151 ± 85459,699	26379391,526 ± 594896,860	25734283,094 ± 396984,211	6567769,815 ± 5370900,125	5734571,446 ± 4200194,080	37900589,614 ± 2068506,111	39455489,686 ± 3021914,955
Banana (muestra)	710043521,792 ± 328576899,079	171254763,143 ± 23120627,806	3018640571,530 ± 662290180,227	271046047,113 ± 804824070,796	1678065223,303 ± 316098653,826	1892722399,562 ± 83464674,405	3009986636,756 ± 222878676,155	4088796523,141 ± 67939067,960
Bupa	94187,366 ± 86139,809	47271,546 ± 3427,994	276863,426 ± 17426,114	286171,523 ± 28491,792	52388,368 ± 3496,285	53291,594 ± 2834,679	44259,765 ± 235853,333	783742,400 ± 6468,107
Ecoli	567701,914 ± 308151,819	69554,053 ± 11533,396	61830,979 ± 161544,000	57628,940 ± 16430,578	72803,136 ± 14083,428	196620,405 ± 92948,202	1954395,443 ± 43566,420	1954395,443 ± 43566,420
Class	280859,026 ± 125014,180	88418,104 ± 15985,833	404583,830 ± 19098,162	373350,099 ± 33697,425	127852,345 ± 42520,552	121280,038 ± 54792,384	659979,156 ± 14938,171	635127,461 ± 20777,903
Haberman	1974771,361 ± 65317,222	2438447,189 ± 2434314,636	4755601,995 ± 2635073,231	6794040,379 ± 2799090,181	4994042,785 ± 1573978,309	5489657,244 ± 871469,294	317675,507 ± 510417,407	7485792,499 ± 282327,895
Hayes Roth	212581,980 ± 70319,872	147945,591 ± 18170,978	453645,792 ± 76337,117	467967,354 ± 109765,822	268555,250 ± 52589,283	267914,057 ± 61072,827	479318,932 ± 17136,753	522962,255 ± 12138,927
Heart	138363,271 ± 349195,109	152409,819 ± 22980,256	145005,041 ± 25206,290	1443551,369 ± 168616,931	156945,215 ± 23315,570	151690,220 ± 32450,980	1589242,054 ± 57914,054	1821884,562 ± 57914,054
Iris	115986,712 ± 12941,797	118712,510 ± 19158,833	115753,024 ± 70180,427	460797,016 ± 169896,950	390429,362 ± 88654,781	366964,496 ± 172887,231	874168,976 ± 13041,658	886059,660 ± 14347,094
LeafDigit	40863,346 ± 333590,299	159863,054 ± 44054,861	1158160,449 ± 124138,270	1149974,050 ± 86681,715	186790,811 ± 36767,383	198614,899 ± 56678,751	6832860,678 ± 397852,081	6946037,680 ± 103601,458
Monks	8095964,587 ± 3387538,273	700466,137 ± 30791,298	8679097,923 ± 741940,698	9018466,012 ± 1163900,560	1463661,059 ± 585999,941	1338118,867 ± 325278,218	5015484,413 ± 848551,024	10464549,732 ± 829872,192
Newthyroid	829596,191 ± 244462,252	142958,283 ± 3196,634	1137124,170 ± 72484,290	994750,250 ± 165495,328	249625,314 ± 72500,664	319867,950 ± 116546,162	1617900,059 ± 7991,628	1635013,115 ± 13407,876
Pima	28204407,176 ± 10195922,713	2113475,429 ± 128443,816	31496165,598 ± 89261,972	31192261,848 ± 2955697,988	2295536,010 ± 47462,353	230205,309 ± 422900,631	4895357,203 ± 501754,860	54299187,182 ± 3841279,187
Saheart	6799533,673 ± 705613,347	628549,445 ± 69351,352	7311046,848 ± 115753,423	714752,262 ± 351825,958	1158292,256 ± 537764,912	1146747,360 ± 385085,439	10398606,439 ± 888489,304	10398606,439 ± 888489,304
Soybean	2146,794 ± 1079,731	1917,694 ± 176,080	2669,372 ± 222,563	2455,528 ± 195,881	1280,057 ± 51,017	1253,510 ± 117,836	2857,458 ± 48,381	2927,017 ± 31,327
Tae	146804,710 ± 69244,971	61970,596 ± 3712,325	22658,843 ± 11724,149	242970,329 ± 25768,003	99641,574 ± 10814,510	98530,707 ± 19921,977	273728,494 ± 23893,717	266470,857 ± 19910,712
Titanic (muestra)	9063334,917 ± 5403439,304	1306706,437 ± 216745,008	17817083,249 ± 1166815,506	1826056,579 ± 1479651,280	817353,312 ± 518836,347	10953667,267 ± 4399271,717	32141757,550 ± 1229597,315	37018505,853 ± 1596570,160
Wine	36392,974 ± 71741,377	47951,463 ± 6198,222	33388,023 ± 42736,886	318234,378 ± 26710,767	48334,334 ± 4008,850	47397,127 ± 5824,400	529284,666 ± 11969,423	525407,547 ± 5136,275
Zoo	25863,505 ± 1289,063	15017,824 ± 2169,173	29083,203 ± 617,183	26807,997 ± 5331,546	7133,583 ± 1266,391	6528,277 ± 813,709	38900,236 ± 2019,072	37561,387 ± 1586,682
Promedio	38890652,899 ± 153977818,678	9063925,463 ± 37216468,797	155940550,069 ± 656794930,183	140759901,389 ± 389583897,524	85227447,863 ± 365429400,229	96074198,564 ± 417188408,134	1580337,57 ± 654424889,510	213201411,328 ± 889251635,251

Tabla 8.9: Hipervolumenes - 20 % restricciones

Dataset	MOEAD-ADA	MOEAD-ADA <sub>Norm</sub>	MOEAD-ADA-PD	MOEAD-ADA-PD-SSPS	MOEAD-ADA-PD <sub>NormB</sub>	MOEAD-ADA-PD <sub>NormA</sub>	MOEAD-ADA-KM	MOEAD-ADA-PD-KM
Appendicitis	142233.641 ± 16413.063	94263.805 ± 19723.913	317407.718 ± 26903.160	305986.241 ± 75995.432	176465.660 ± 2327.1188	176465.660 ± 32278.690	238184.760 ± 17911.623	300593.843 ± 25112.978
Balance	1564806.890 ± 905558.017	1241498.935 ± 60386.228	25373.167 ± 2125294.325	26800412.279 ± 2203420.122	4813269.067 ± 35623.68.294	6682072.536 ± 4534323.863	3637560.463 ± 1906873.399	4047278.144 ± 3734427.304
Banana (muestra)	1127616251.737 ± 16372215.408	2068472307.208 ± 2741794465.132	1138654631.184 ± 939622755.437	12347113425.466 ± 634296430.037	5447570356.472 ± 1027020417.214	5561450900.220 ± 1752337760.214	8927509397.497 ± 135126584.664	13657445947.957 ± 778480739.214
Bupa	157461.056 ± 144331.332	85030.951 ± 7317.969	417640.308 ± 22684.979	44007.174 ± 43677.141	100867.809 ± 11619.546	88587.310 ± 12601.503	1050383.748 ± 152151.232	1234992.221 ± 31743.169
Ecoli	814178.806 ± 460387.677	70458.093 ± 11392.746	1214851.721 ± 114870.624	120207.537 ± 146780.235	65819.230 ± 4355.503	68480.008 ± 4537.975	2812820.710 ± 194276.078	2697110.996 ± 172894.053
Glass	342924.413 ± 94686.692	76303.924 ± 19436.924	407933.387 ± 3487.498	492026.638 ± 63759.899	101133.176 ± 29393.945	97611.381 ± 37643.707	554305.944 ± 35607.068	550622.139 ± 54584.340
Haberman	32712926.172 ± 2172456.637	13588389.292 ± 14010827.866	44434536.664 ± 3025389.360	47925871.307 ± 4325676.258	31969407.576 ± 13907202.330	25800006.548 ± 10778797.400	19826767.629 ± 3310503.589	42727182.610 ± 5791810.204
Hayes Roth	550789.813 ± 196258.456	262048.731 ± 55009.411	891009.871 ± 119075.645	999893.943 ± 49683.411	487216.133 ± 14777.4.403	419619.839 ± 91800.456	78847.157 ± 3242.497	921892.091 ± 82216.244
Heart	2621472.564 ± 626232.866	256197.620 ± 19052.440	2981754.135 ± 80028.698	3180560.130 ± 126758.135	340210.149 ± 77500.767	305239.595 ± 110292.564	333921.1530 ± 295104.103	3669693.703 ± 18549.155
Iris	426356.957 ± 383991.499	293415.644 ± 244442.897	1502897.030 ± 209115.941	1538586.353 ± 384781.895	768875.906 ± 325334.834	692128.639 ± 277067.280	1770525.631 ± 69321.780	193227.639 ± 42725.113
LeafDigit	437034.303 ± 386263.251	190874.432 ± 105622.200	140773.817 ± 141556.996	1381048.720 ± 175711.543	201564.380 ± 40724.568	174827.737 ± 355037.277	11266895.579 ± 353037.188	10731981.346 ± 194485.493
Monks	12736970.580 ± 3122463.025	882467.276 ± 158899.008	13956227.244 ± 1512626.707	13927105.284 ± 1614109.026	3065853.173 ± 1238471.715	844452.170 ± 3938021.188	12814412.591 ± 1247616.787	12814412.591 ± 1247616.787
Newthyroid	1215971.615 ± 482202.272	146209.724 ± 46223.336	1707225.749 ± 51065.916	1872346.040 ± 254937.798	524519.830 ± 347933.270	413618.191 ± 205806.866	2526616.114 ± 430709.694	2482031.143 ± 33508.407
Pima	24109364.161 ± 8249843.272	1660834.766 ± 98855.047	33341050.529 ± 309103.080	34756042.007 ± 2333.693.406	2362053.482 ± 86093.641	2021942.105 ± 331699.303	45907033.723 ± 3946637.043	54374588.891 ± 1564834.718
Saheart	7979304.410 ± 3277663.007	595867.311 ± 37713.745	9393004.990 ± 465941.478	8305238.439 ± 657429.054	1403328.219 ± 4537.66.949	2554331.726 ± 2086322.827	9339393.604 ± 885373.186	13153525.956 ± 1064119.110
Soybean	8217.690 ± 151.042	6081.242 ± 657.070	7695.613 ± 193.404	7972.144 ± 428.290	4227.379 ± 294.883	4232.159 ± 166.526	8483.169 ± 90.868	8658.790 ± 51.086
Tae	385720.150 ± 102719.181	385720.150 ± 102719.181	470059.382 ± 36288.381	471752.744 ± 26136.086	172914.550 ± 71598.047	161105.747 ± 78889.259	408100.165 ± 33819.222	466183.220 ± 31556.498
Titanic (muestra)	115266187.717 ± 23433854.524	6369399.235 ± 803276.264	117911033.987 ± 2321103.896	11887890.636 ± 5836443.352	54356322.368 ± 3365967.932	57705888.757 ± 32810923.345	168926616.236 ± 3603754.739	202031392.175 ± 7804405.928
Wine	1530472.248 ± 374033.778	275230.361 ± 113923.549	1691982.073 ± 101935.751	1728400.191 ± 224406.419	200236.704 ± 19797.772	193422.639 ± 14829.307	228278.217 ± 63264.506	1707824.193 ± 854870.205
Zoo	41972.819 ± 5130.697	19978.539 ± 1686.531	40407.311 ± 8763.019	38062.666 ± 10910.916	8167.891 ± 1988.531	7418.820 ± 3215.714	57267.607 ± 2546.734	57227.607 ± 2546.734
Promedio	672239704.287 ± 244595304.819	104722112.652 ± 450523715.919	382170688.605 ± 2478839542.004	630573600.597 ± 2688097718.727	277386211.158 ± 1211013128.961	2831079654.159 ± 711013128.961	462194781.712 ± 1942429510.822	702488445.863 ± 2972404374.593

Tabla 8.10: Tamaños frentes de Pareto - 10 % restricciones

Dataset	MOEAD-ADA	MOEAD-ADA <sub>Norm</sub>	MOEAD-ADA-PD	MOEAD-ADA-PD-SSPS	MOEAD-ADA-PD <sub>NormA</sub>	MOEAD-ADA-PD <sub>NormB</sub>	MOEAD-ADA-KM	MOEAD-ADA-PD-KM
Appendicitis	34,200 ± 11,374	49,800 ± 3,655	32,800 ± 6,997	39,800 ± 17,058	24,600 ± 6,280	18,000 ± 6,325	17,800 ± 2,315	17,000 ± 4,147
Balance	46,600 ± 26,867	53,400 ± 15,226	39,000 ± 19,799	58,000 ± 24,503	44,400 ± 18,095	39,800 ± 6,462	28,000 ± 6,293	20,600 ± 4,630
Banana (muestra)	60,200 ± 11,479	51,400 ± 15,028	33,600 ± 8,890	36,200 ± 18,819	31,800 ± 18,324	40,600 ± 7,864	38,000 ± 13,653	71,600 ± 11,842
Bupa	70,800 ± 21,729	74,600 ± 17,258	88,800 ± 19,385	82,800 ± 17,429	53,800 ± 8,908	62,200 ± 10,515	43,800 ± 12,560	41,600 ± 6,184
Ecoli	57,800 ± 23,438	60,200 ± 5,980	49,600 ± 13,720	56,600 ± 8,523	45,600 ± 8,523	56,600 ± 14,051	20,400 ± 6,020	23,600 ± 5,783
Glass	54,000 ± 27,698	56,200 ± 9,064	54,200 ± 13,242	53,800 ± 7,521	55,800 ± 10,943	43,800 ± 10,553	54,600 ± 8,777	36,600 ± 4,630
Haberman	57,800 ± 16,666	57,200 ± 8,134	54,000 ± 9,317	51,200 ± 22,498	42,800 ± 17,452	33,200 ± 3,655	41,600 ± 8,015	45,400 ± 5,314
Hayes Roth	54,200 ± 13,732	42,200 ± 17,543	54,600 ± 13,017	69,600 ± 19,866	36,000 ± 7,239	25,200 ± 4,069	21,200 ± 12,750	25,200 ± 5,528
Heart	15,600 ± 7,200	47,600 ± 13,002	52,200 ± 16,179	58,800 ± 24,252	37,000 ± 10,237	41,800 ± 20,302	26,200 ± 10,496	29,600 ± 4,030
Iris	46,000 ± 17,053	52,600 ± 18,073	52,800 ± 11,600	51,600 ± 8,333	34,600 ± 6,974	40,600 ± 9,769	9,400 ± 3,441	13,000 ± 5,254
LeafDigit	62,200 ± 24,087	62,600 ± 9,200	49,000 ± 11,082	52,600 ± 5,276	51,600 ± 7,632	48,200 ± 17,104	19,000 ± 2,966	23,000 ± 9,737
Monks	35,000 ± 23,426	55,000 ± 11,331	21,600 ± 7,060	29,400 ± 10,052	30,400 ± 15,907	23,600 ± 8,523	25,000 ± 7,155	27,400 ± 1,960
Newthyroid	33,000 ± 7,510	61,600 ± 11,412	37,800 ± 12,937	47,200 ± 9,453	29,600 ± 9,891	31,800 ± 14,063	8,600 ± 2,939	9,800 ± 5,036
Pima	25,600 ± 13,705	59,800 ± 15,224	24,000 ± 7,563	38,200 ± 11,391	39,400 ± 17,142	43,400 ± 11,926	30,400 ± 8,476	45,000 ± 12,506
Saheart	36,800 ± 8,841	61,000 ± 17,493	31,400 ± 7,658	45,400 ± 13,822	43,800 ± 9,621	40,200 ± 7,521	26,200 ± 8,818	32,600 ± 6,468
Soybean	16,200 ± 5,418	18,200 ± 1,600	32,200 ± 11,788	27,600 ± 16,169	17,200 ± 5,564	24,600 ± 6,406	5,200 ± 2,482	11,400 ± 2,728
Tae	50,200 ± 16,810	56,000 ± 2,280	57,600 ± 18,618	85,200 ± 34,073	44,400 ± 4,317	37,800 ± 5,418	44,600 ± 6,741	41,200 ± 4,490
Titanic (muestra)	78,600 ± 12,893	38,600 ± 13,230	34,800 ± 8,035	26,600 ± 8,823	19,400 ± 4,079	36,000 ± 10,334	29,600 ± 6,119	28,600 ± 3,611
Wine	37,600 ± 22,357	38,400 ± 3,611	43,400 ± 26,242	57,800 ± 18,148	43,400 ± 10,365	45,800 ± 21,507	9,400 ± 2,926	9,400 ± 3,262
Zoo	20,600 ± 11,377	31,800 ± 6,046	28,000 ± 8,786	39,000 ± 15,100	36,000 ± 6,693	35,200 ± 10,496	14,000 ± 5,215	10,800 ± 3,600
Promedio	44,650 ± 17,308	51,410 ± 12,298	43,570 ± 15,072	50,370 ± 15,850	38,080 ± 10,306	38,420 ± 10,526	25,670 ± 13,212	28,170 ± 15,250

Tabla 8.1.1: Tamaños frentes de Pareto - 15 % restricciones

Dataset	MOEAD-ADA	MOEAD-ADA <sub>Norm</sub>	MOEAD-ADA-PD	MOEAD-ADA-PD-SSFS	MOEAD-ADA-PD <sub>NormA</sub>	MOEAD-ADA-PD <sub>NormB</sub>	MOEAD-ADA-KM	MOEAD-ADA-PD-KM
Appendicitis	34,400 ± 10,111	37,200 ± 9,152	18,000 ± 11,950	18,000 ± 10,954	6,800 ± 3,311	13,400 ± 4,079	12,600 ± 3,007	9,600 ± 2,154
Balance	25,200 ± 7,859	27,000 ± 9,899	20,600 ± 7,736	23,400 ± 10,072	35,000 ± 21,241	24,200 ± 12,828	25,800 ± 7,935	25,400 ± 3,826
Banana (muestra)	55,400 ± 29,097	64,600 ± 20,896	38,200 ± 8,976	36,800 ± 5,269	35,400 ± 4,883	33,400 ± 6,280	43,000 ± 4,147	58,200 ± 8,841
Bupa	93,600 ± 20,915	74,800 ± 4,956	103,400 ± 33,536	90,600 ± 22,446	65,400 ± 17,602	60,800 ± 15,381	49,600 ± 19,085	46,200 ± 11,338
Ecoli	56,600 ± 12,043	62,600 ± 12,955	46,600 ± 12,093	58,400 ± 18,597	49,800 ± 7,111	47,600 ± 15,932	15,800 ± 3,970	18,000 ± 4,817
Glass	50,400 ± 10,557	55,000 ± 11,367	50,200 ± 14,048	51,200 ± 15,315	41,400 ± 12,435	38,200 ± 5,528	53,000 ± 12,280	43,600 ± 6,800
Haberman	46,200 ± 11,496	49,800 ± 12,844	33,800 ± 13,600	63,200 ± 15,917	35,000 ± 14,128	43,800 ± 20,153	50,200 ± 15,065	85,400 ± 24,229
Hayes Roth	66,000 ± 26,480	49,200 ± 3,763	61,000 ± 14,381	81,800 ± 26,476	35,000 ± 11,679	29,600 ± 14,305	36,600 ± 10,288	40,200 ± 12,480
Heart	21,000 ± 8,270	38,800 ± 15,302	16,800 ± 2,638	23,400 ± 10,707	15,400 ± 6,859	15,600 ± 3,441	19,000 ± 9,940	23,000 ± 5,762
Iris	58,000 ± 10,564	41,000 ± 17,675	24,800 ± 4,214	39,600 ± 13,321	21,000 ± 6,132	18,200 ± 6,615	8,600 ± 4,409	11,400 ± 3,007
Led7Digit	55,400 ± 15,781	60,000 ± 6,723	49,600 ± 8,523	58,400 ± 21,444	51,800 ± 8,931	59,800 ± 11,889	19,200 ± 7,626	17,400 ± 3,382
Monk2	20,600 ± 7,915	52,400 ± 11,465	16,000 ± 7,720	21,000 ± 6,229	18,200 ± 3,187	12,400 ± 4,317	29,200 ± 12,007	18,800 ± 7,139
Newthyroid	43,600 ± 24,212	40,200 ± 5,564	35,000 ± 8,626	37,000 ± 19,142	20,600 ± 5,748	19,200 ± 10,438	15,200 ± 6,431	12,000 ± 2,449
Pima	29,400 ± 8,499	59,200 ± 14,428	27,400 ± 8,188	36,200 ± 14,148	36,600 ± 13,215	37,200 ± 10,206	34,000 ± 9,529	40,400 ± 5,238
Saheart	31,200 ± 13,106	45,600 ± 8,452	30,000 ± 10,714	39,400 ± 6,946	24,000 ± 13,342	20,000 ± 3,847	32,200 ± 11,565	51,000 ± 9,757
Soybean	10,000 ± 6,957	23,200 ± 5,492	27,000 ± 11,349	32,600 ± 15,907	21,200 ± 4,833	20,200 ± 5,913	4,800 ± 1,470	13,000 ± 3,286
Tae	41,200 ± 13,121	54,000 ± 7,925	52,800 ± 8,328	48,800 ± 12,576	33,000 ± 7,746	34,600 ± 4,409	36,000 ± 7,155	32,000 ± 6,782
Titanic (muestra)	38,600 ± 29,131	40,000 ± 18,439	34,200 ± 5,946	25,000 ± 5,762	21,000 ± 12,296	20,000 ± 6,603	17,400 ± 4,841	48,800 ± 9,516
Wine	24,800 ± 14,689	44,600 ± 15,200	15,200 ± 7,167	35,200 ± 12,921	41,000 ± 6,723	33,400 ± 13,705	6,200 ± 2,713	6,600 ± 4,716
Zoo	29,600 ± 14,773	23,000 ± 5,550	32,400 ± 13,764	45,800 ± 19,156	31,800 ± 7,985	34,800 ± 7,859	15,800 ± 3,970	17,600 ± 5,886
Promedio	41,560 ± 18,913	47,110 ± 13,496	36,650 ± 20,058	43,290 ± 19,074	31,970 ± 13,617	30,820 ± 13,943	26,210 ± 14,667	30,930 ± 19,849

Tabla 8.1.2: Tamaños frentes de Pareto - 20 % restricciones

Dataset	MOEAD-ADA	MOEAD-ADA <sub>Norm</sub>	MOEAD-ADA-PD	MOEAD-ADA-PD-SSFS	MOEAD-ADA-PD <sub>NormA</sub>	MOEAD-ADA-PD <sub>NormB</sub>	MOEAD-ADA-KM	MOEAD-ADA-PD-KM
Appendicitis	24,200 ± 10,458	27,600 ± 16,057	15,600 ± 7,310	18,400 ± 9,646	8,000 ± 4,243	10,000 ± 4,899	15,800 ± 3,868	22,200 ± 8,376
Balance	45,200 ± 25,278	47,600 ± 10,327	26,400 ± 7,392	28,000 ± 11,541	20,600 ± 8,188	17,600 ± 8,114	22,000 ± 10,954	40,800 ± 9,558
Banana (muestra)	69,400 ± 3,382	53,200 ± 28,138	36,400 ± 9,992	40,600 ± 8,663	33,400 ± 6,184	30,200 ± 8,060	37,200 ± 10,225	62,000 ± 6,197
Bupa	78,800 ± 14,246	80,600 ± 16,107	107,000 ± 48,814	113,800 ± 18,269	68,800 ± 19,894	62,800 ± 9,495	99,800 ± 20,498	45,800 ± 7,705
Ecoli	43,000 ± 20,189	59,200 ± 10,852	35,600 ± 6,771	38,000 ± 5,020	44,200 ± 17,600	48,400 ± 15,756	15,400 ± 7,499	13,200 ± 5,418
Glass	50,000 ± 18,221	51,400 ± 14,009	37,800 ± 8,750	40,600 ± 8,868	42,200 ± 17,116	38,000 ± 10,826	68,400 ± 16,620	51,400 ± 13,094
Haberman	63,000 ± 17,297	50,400 ± 12,043	51,400 ± 9,372	56,200 ± 16,241	38,000 ± 16,162	29,400 ± 10,837	51,000 ± 10,020	62,200 ± 18,335
Hayes Roth	48,000 ± 18,601	49,800 ± 2,786	58,600 ± 20,906	56,000 ± 11,419	32,600 ± 3,262	33,200 ± 2,993	56,200 ± 13,761	52,800 ± 11,634
Heart	19,800 ± 8,256	39,200 ± 11,125	16,000 ± 3,578	13,600 ± 3,774	9,200 ± 4,996	12,000 ± 9,859	17,800 ± 5,036	26,200 ± 4,833
Iris	47,000 ± 23,841	26,600 ± 4,499	15,200 ± 16,558	16,800 ± 14,020	18,000 ± 6,812	15,800 ± 6,969	8,800 ± 3,763	13,400 ± 5,161
Led7Digit	62,200 ± 16,339	55,600 ± 13,425	48,600 ± 13,139	71,600 ± 30,539	47,600 ± 10,538	52,400 ± 17,693	13,600 ± 4,271	13,000 ± 4,000
Monk2	18,600 ± 2,417	34,800 ± 15,740	9,200 ± 1,166	21,200 ± 8,704	12,200 ± 5,418	8,600 ± 4,800	38,200 ± 13,600	13,200 ± 6,369
Newthyroid	30,200 ± 15,778	36,000 ± 9,633	21,800 ± 4,956	20,600 ± 3,441	21,200 ± 6,177	20,800 ± 4,534	14,000 ± 4,147	15,600 ± 4,363
Pima	36,800 ± 14,552	50,600 ± 15,525	25,200 ± 4,490	22,800 ± 6,735	21,600 ± 9,625	25,000 ± 7,563	32,600 ± 12,579	40,400 ± 7,446
Saheart	28,200 ± 16,314	51,200 ± 16,606	26,800 ± 10,980	35,200 ± 10,147	24,200 ± 9,662	18,800 ± 10,685	28,600 ± 6,312	47,000 ± 6,261
Soybean	5,600 ± 2,653	20,800 ± 7,494	35,000 ± 16,840	29,800 ± 23,146	28,200 ± 6,554	23,200 ± 6,853	4,800 ± 1,720	13,600 ± 4,317
Tae	52,800 ± 26,156	49,600 ± 8,015	39,600 ± 14,291	51,400 ± 18,095	30,600 ± 8,381	33,000 ± 11,419	49,600 ± 10,442	50,000 ± 7,720
Titanic (muestra)	31,800 ± 6,705	31,600 ± 7,228	40,000 ± 2,530	44,800 ± 8,612	17,800 ± 7,026	15,200 ± 5,776	18,800 ± 11,771	62,400 ± 17,973
Wine	15,800 ± 12,544	38,200 ± 19,093	8,600 ± 3,611	12,600 ± 10,052	27,600 ± 7,710	31,800 ± 13,585	6,200 ± 1,327	4,600 ± 2,332
Zoo	26,600 ± 3,666	28,000 ± 6,419	37,000 ± 13,387	38,200 ± 20,390	32,000 ± 7,294	26,200 ± 6,882	10,600 ± 3,262	11,200 ± 5,564
Promedio	39,850 ± 18,949	44,100 ± 13,754	34,590 ± 21,395	38,510 ± 23,269	28,900 ± 14,201	27,620 ± 14,039	30,470 ± 23,756	33,050 ± 19,679

## 8.2 ANÁLISIS ESTADÍSTICO DE LOS RESULTADOS

Referencia principal: [52]

## 8.3 CONCLUSIONES

PD-KM es el mejor.

## 8.4 POSIBLES TRABAJOS FUTUROS

Uso de búsqueda local

Probar nuevos esquemas de representación

Probar una nueva métrica de distancia para criterio de vecindario  
en espacio de soluciones



## BIBLIOGRAFÍA

---

- [1] David Reinsel, John Gantz y John Rydning. «Data age 2025: The evolution of data to life-critical». En: *Don't Focus on Big Data 2* (2017).
- [2] Arthur L Samuel. «Some studies in machine learning using the game of checkers». En: *IBM Journal of research and development* 3.3 (1959), págs. 210-229.
- [3] Tom M Mitchell y col. *Machine learning*. McGraw-hill New York, 1997.
- [4] Yaser S Abu-Mostafa, Malik Magdon-Ismael y Hsuan-Tien Lin. *Learning from data*. Vol. 4. AMLBook New York, NY, USA: 2012.
- [5] Brian S Everitt, Sabine Landau, Morven Leese y Daniel Stahl. *Cluster analysis 5th ed*. John Wiley, 2011.
- [6] Paul E Green, Ronald E Frank y Patrick J Robinson. «Cluster analysis in test market selection». En: *Management science* 13.8 (1967), B-387.
- [7] Chuck Chakrapani. *Statistics in market research*. Oxford University Press, 2004.
- [8] M Faúndez-Abans, MI Ormeno y M de Oliveira-Abans. «Classification of planetary nebulae by cluster analysis and artificial neural networks». En: *Astronomy and Astrophysics Supplement Series* 116.2 (1996), págs. 395-402.
- [9] Gilles Celeux y Gérard Govaert. «A classification EM algorithm for clustering and two stochastic versions». En: *Computational statistics & Data analysis* 14.3 (1992), págs. 315-332.
- [10] Frank R Hodson. «Numerical typology and prehistoric archaeology». En: *Mathematics in the archaeological and historical Sciences* (1971), págs. 30-45.
- [11] Mark Q Sutton y Karl J Reinhard. «Cluster analysis of the coprolites from Antelope House: implications for Anasazi diet and cuisine». En: *Journal of Archaeological Science* 22.6 (1995), págs. 741-750.
- [12] Michael B Eisen, Paul T Spellman, Patrick O Brown y David Botstein. «Cluster analysis and display of genome-wide expression patterns». En: *Proceedings of the National Academy of Sciences* 95.25 (1998), págs. 14863-14868.
- [13] Adán José-García y Wilfrido Gómez-Flores. «Automatic clustering using nature-inspired metaheuristics: A survey». En: *Applied Soft Computing* 41 (2016), págs. 192-213.

- [14] James MacQueen y col. «Some methods for classification and analysis of multivariate observations». En: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, págs. 281-297.
- [15] Sergios Theodoridis y Konstantinos Koutroumbas. «Pattern recognition and neural networks». En: *Advanced Course on Artificial Intelligence*. Springer. 1999, págs. 169-195.
- [16] Julia Handl y Joshua Knowles. «An evolutionary approach to multiobjective clustering». En: *IEEE transactions on Evolutionary Computation* 11.1 (2007), págs. 56-76.
- [17] Tadeusz Caliński y Jerzy Harabasz. «A dendrite method for cluster analysis». En: *Communications in Statistics-theory and Methods* 3.1 (1974), págs. 1-27.
- [18] David L Davies y Donald W Bouldin. «A cluster separation measure». En: *IEEE transactions on pattern analysis and machine intelligence* 2 (1979), págs. 224-227.
- [19] Peter J Rousseeuw. «Silhouettes: a graphical aid to the interpretation and validation of cluster analysis». En: *Journal of computational and applied mathematics* 20 (1987), págs. 53-65.
- [20] Olivier Chapelle, Bernhard Scholkopf y Alexander Zien. *Semi-supervised learning*. Cambridge, Massachusettes: The MIT Press View Article, 2006.
- [21] Ian Davidson y Sugato Basu. «A survey of clustering with instance level constraints». En: *ACM Transactions on Knowledge Discovery from data* 1.1-41 (2007), págs. 2-42.
- [22] Kiri Wagstaff y Claire Cardie. «Clustering with instance-level constraints». En: *AAAI/IAAI* 1097 (2000), págs. 577-584.
- [23] Sugato Basu, Ian Davidson y Kiri Wagstaff. *Constrained clustering: Advances in algorithms, theory, and applications*. CRC Press, 2008.
- [24] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl y col. «Constrained k-means clustering with background knowledge». En: *Icml*. Vol. 1. 2001, págs. 577-584.
- [25] Ian Davidson y SS Ravi. «Hierarchical clustering with constraints: Theory and practice». En: *Proceedings of the ninth European principles and practice of KDD (PKDD)* (2005), págs. 59-70.
- [26] Sugato Basu, Arindam Banerjee y Raymond J Mooney. «Active semi-supervision for pairwise constrained clustering». En: *Proceedings of the 2004 SIAM international conference on data mining*. SIAM. 2004, págs. 333-344.
- [27] Eran Segal, Haidong Wang y Daphne Koller. «Discovering molecular pathways from protein interaction and gene expression data». En: *Bioinformatics* 19.suppl\_1 (2003), págs. i264-i272.

- [28] Ian Davidson y SS Ravi. «Clustering with constraints: Feasibility issues and the k-means algorithm». En: *Proceedings of the 2005 SIAM international conference on data mining*. SIAM. 2005, págs. 138-149.
- [29] Martin HC Law, Alexander Topchy y Anil K Jain. «Model-based clustering with probabilistic constraints». En: *Proceedings of the 2005 SIAM international conference on data mining*. SIAM. 2005, págs. 641-645.
- [30] Ian Davidson, Kiri L Wagstaff y Sugato Basu. «Measuring constraint-set utility for partitional clustering algorithms». En: *European conference on principles of data mining and knowledge discovery*. Springer. 2006, págs. 115-126.
- [31] Carlos A Coello Coello, Gary B Lamont, David A Van Veldhuizen y col. *Evolutionary algorithms for solving multi-objective problems*. Vol. 5. Springer, 2007.
- [32] Anirban Mukhopadhyay, Ujjwal Maulik, Sanghamitra Bandyopadhyay y Carlos Artemio Coello Coello. «A survey of multi-objective evolutionary algorithms for data mining: Part I». En: *IEEE Transactions on Evolutionary Computation* 18.1 (2013), págs. 4-19.
- [33] Kaisa Miettinen. *Nonlinear multiobjective optimization*. Vol. 12. Springer Science & Business Media, 2012.
- [34] El-Ghazali Talbi. *Metaheuristics: from design to implementation*. Vol. 74. John Wiley & Sons, 2009.
- [35] Scott Kirkpatrick, C Daniel Gelatt y Mario P Vecchi. «Optimization by simulated annealing». En: *science* 220.4598 (1983), págs. 671-680.
- [36] Fred Glover. «Tabu search: A tutorial». En: *Interfaces* 20.4 (1990), págs. 74-94.
- [37] John Henry Holland y col. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.
- [38] Rainer Storn y Kenneth Price. «Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces». En: *Journal of global optimization* 11.4 (1997), págs. 341-359.
- [39] James Kennedy y Russell Eberhart. «Particle swarm optimization». En: *Proceedings of ICNN'95-international conference on neural networks*. Vol. 4. IEEE. 1995, págs. 1942-1948.
- [40] Marco Dorigo, Vittorio Maniezzo y Alberto Colorni. «Ant system: optimization by a colony of cooperating agents». En: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 26.1 (1996), págs. 29-41.

- [41] Rahila H Sheikh, Mukesh M Raghuwanshi y Anil N Jaiswal. «Genetic algorithm based clustering: a survey». En: *2008 First International Conference on Emerging Trends in Engineering and Technology*. IEEE. 2008, págs. 314-319.
- [42] Germán González-Almagro, Julián Luengo, José-Ramón Cano y Salvador García. «DILS: constrained clustering through dual iterative local search». En: *Computers & Operations Research* 121 (2020), pág. 104979.
- [43] Qingfu Zhang y Hui Li. «MOEA/D: A multiobjective evolutionary algorithm based on decomposition». En: *IEEE Transactions on evolutionary computation* 11.6 (2007), págs. 712-731.
- [44] Ryoji Tanabe e Hisao Ishibuchi. «A framework to handle multi-modal multiobjective optimization in decomposition-based evolutionary algorithms». En: *IEEE Transactions on Evolutionary Computation* 24.4 (2019), págs. 720-734.
- [45] Ryoji Tanabe e Hisao Ishibuchi. «A decomposition-based evolutionary algorithm for multi-modal multi-objective optimization». En: *International conference on parallel problem solving from nature*. Springer. 2018, págs. 249-261.
- [46] Larry J Eshelman y J David Schaffer. «Real-coded genetic algorithms and interval-schemata». En: *Foundations of genetic algorithms*. Vol. 2. Elsevier, 1993, págs. 187-202.
- [47] Kalyanmoy Deb, Ram Bhushan Agrawal y col. «Simulated binary crossover for continuous search space». En: *Complex systems* 9.2 (1995), págs. 115-148.
- [48] Zhenkun Wang, Qingfu Zhang, Aimin Zhou, Maoguo Gong y Licheng Jiao. «Adaptive replacement strategies for MOEA/D». En: *IEEE transactions on cybernetics* 46.2 (2015), págs. 474-486.
- [49] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M Fonseca y Viviane Grunert Da Fonseca. «Performance assessment of multiobjective optimizers: An analysis and review». En: *IEEE Transactions on evolutionary computation* 7.2 (2003), págs. 117-132.
- [50] Eckart Zitzler y Lothar Thiele. «Multiobjective optimization using evolutionary algorithms—a comparative case study». En: *International conference on parallel problem solving from nature*. Springer. 1998, págs. 292-301.
- [51] Nicola Beume, Carlos M Fonseca, Manuel Lopez-Ibanez, Luis Paquete y Jan Vahrenhold. «On the complexity of computing the hypervolume indicator». En: *IEEE Transactions on Evolutionary Computation* 13.5 (2009), págs. 1075-1082.
- [52] Alessio Benavoli, Giorgio Corani, Janez Demšar y Marco Zafalon. «Time for a change: a tutorial for comparing multiple classifiers through Bayesian analysis». En: *The Journal of Machine Learning Research* 18.1 (2017), págs. 2653-2688.

## COLOFÓN

Este documento ha sido escrito usando el paquete de estilo `classicthesis`, creado por André Miede e Ivo Pletikosić.

`classicthesis` está disponible para  $\text{\LaTeX}$  en el siguiente enlace:

<https://bitbucket.org/amiede/classicthesis/>