



Universidad Nacional Autónoma de México

Posgrado en Ciencia e Ingeniería de la
Computación

El efecto de la heterogeneidad temporal en la presión de
selección de los algoritmos evolutivos

T E S I S

Que Para Optar Por El Grado De
Maestro en Ciencia e Ingeniería de la Computación

PRESENTA:
Víctor Manuel Sánchez Sánchez

Tutor Principal:
Dr. Carlos Gershenson García, Instituto de Investigaciones en Matemáticas
Aplicadas y en Sistemas - UNAM
Co-Tutor:
Dr. Carlos Ignacio Hernández Castellanos, Instituto de Investigaciones en
Matemáticas Aplicadas y en Sistemas - UNAM

México, CDMX. (Febrero) 2024



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas

Tesis Digitales

Restricciones de uso

DERECHOS RESERVADOS ©

PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Resumen

Los algoritmos evolutivos de última generación han demostrado ser altamente elitistas, privilegiando una selección determinista ($\mu + \lambda$) que favorece a los mejores individuos para la siguiente generación. Esta aproximación, aunque eficiente en ciertos contextos, puede limitar la diversidad genética necesaria para una exploración efectiva del espacio de soluciones, especialmente en problemas complejos. La investigación actual se enfoca en evaluar cómo la introducción de la heterogeneidad temporal en los algoritmos evolutivos puede equilibrar la exploración y explotación, a través de la modificación de la presión de selección entre estrategias ($\mu + \lambda$) y (μ, λ). Esta adaptación se ha examinado tanto en contextos de optimización mono-objetivo como multi-objetivo, aplicando los algoritmos a conjuntos de datos referenciales en el ámbito académico.

Los resultados preliminares muestran diferencias estadísticamente significativas en la optimización mono-objetivo y multi-objetivo, lo que sugiere que la adaptación de la presión de selección podría ser crucial para mejorar la eficacia de los algoritmos en diferentes problemas y etapas de la búsqueda. Este hallazgo subraya la importancia de la flexibilidad en los algoritmos evolutivos y plantea la necesidad de una mayor adaptabilidad para enfrentar la variedad y complejidad de los problemas de optimización. Este estudio aporta valiosas perspectivas sobre la importancia de la adaptabilidad de los algoritmos evolutivos en función de la naturaleza y complejidad de los problemas abordados, promoviendo una mejor comprensión de las estrategias de selección en la resolución de desafíos de optimización complejos y variados.

Reconocimientos

En primer lugar, deseo extender mi profundo agradecimiento al CONACyT (Consejo Nacional de Ciencia y Tecnología) por la generosa beca otorgada, que ha hecho posible la continuación de mis estudios de maestría en el campo de las ciencias de la computación.

A la Universidad Nacional Autónoma de México (UNAM), agradezco por abrirme las puertas de sus aulas y brindarme la oportunidad de formar parte de esta prestigiosa institución. Mi gratitud se dirige a mis distinguidos profesores, cuyo compromiso y generosidad al compartir su vasto conocimiento y experiencia han sido fundamentales en mi formación académica. Asimismo, agradezco a mis compañeros de estudios, con quienes he compartido este enriquecedor camino académico y de quienes he aprendido mucho.

Quiero expresar mi reconocimiento y agradecimiento a mis tutores, el Dr. Carlos Gershenson y el Dr. Carlos Hernández, cuya guía, orientación y apoyo me han permitido alcanzar este punto en mi investigación. Su compromiso con la excelencia académica ha sido una fuente constante de inspiración y motivación.

Por último, no puedo dejar de agradecer a mi familia y amigos, cuyo apoyo incondicional y ánimo constante han sido fundamentales en mi travesía académica. Sus palabras de aliento y su presencia en cada paso de este camino han sido un verdadero motor para superar los desafíos y lograr mis metas.

Este logro no hubiera sido posible sin la colaboración y el apoyo de todos ustedes. Sus contribuciones han dejado una huella imborrable en mi experiencia académica y personal, por lo que les estoy profundamente agradecido.

Contenido

1. Introducción	6
1.1. Motivación	6
1.2. El problema	7
1.3. Hipótesis	7
1.4. Objetivos	7
1.4.1. Generales	8
1.4.2. Particulares	8
1.5. Contribuciones y productos de investigación	8
1.6. Organización de la tesis	9
2. Marco Teórico	10
2.1. Optimización	10
2.1.1. Fundamentos de la optimización multi-Objetivo	11
2.1.2. Dominancia de Pareto	11
2.1.3. Diversidad de soluciones	14
2.2. El balance entre exploración y explotación	14
2.2.1. El impacto en MOEAs	14
2.2.2. Descripción de selección de siguiente población	15
2.3. Descripción de los algoritmos	16
2.3.1. Mono-objetivo	17
2.3.2. Multi-objetivo	22
2.4. Software	23
2.4.1. Pymoo	24
2.4.2. COCO-Framework	25
2.5. Problemas de prueba	25
2.5.1. BBOB	26
2.5.2. BiBBOB	29
2.5.3. WFG	30
2.6. Indicadores de desempeño	31
2.6.1. Hipervolumen	31
2.6.2. IGD/+	33
2.7. Ciencias de la complejidad	34
2.7.1. Diversidad y adaptación en sistemas complejos	34
2.7.2. Técnicas de adaptación y auto-organización	35
2.7.3. Heterogeneidad temporal	35

2.8.	Modelos de estudio	35
2.8.1.	Kruskal-Wallis	36
2.8.2.	Wilcoxon ranksum	37
2.8.3.	Conteo de Borda	38
2.9.	Explicabilidad de modelos de aprendizaje	38
2.9.1.	SHAP	39
2.9.2.	P-Flacco	40
2.10.	Ajuste de parámetros	41
2.10.1.	Random Forest Regression	41
2.11.	Estado del arte	42
3.	Metodología	44
3.1.	Diseño global con parámetro de heterogeneidad temporal	44
3.2.	Marco experimental	45
3.2.1.	Experimentos mono-objetivo	45
3.2.2.	Experimentos multi-objetivo	48
3.3.	Análisis	50
3.3.1.	Esquema general	51
3.3.2.	Base de datos	52
3.3.3.	Gráficas de convergencia	53
3.3.4.	Modelos	54
3.3.5.	Análisis de los espacios de búsqueda	54
4.	Resultados	56
4.1.	Mono-objetivo	56
4.1.1.	Análisis empírico	56
4.1.2.	Gráficos de convergencia	57
4.1.3.	Pruebas estadísticas	59
4.1.4.	Explicabilidad	66
4.2.	Multi-objetivo	69
4.2.1.	Análisis empírico	69
4.2.2.	Gráficos de convergencia	72
4.2.3.	Pruebas estadísticas	73
4.2.4.	Explicabilidad	84
4.3.	Discusión	86
5.	Conclusiones y trabajo futuro	87
5.1.	Resultados en problemas mono-objetivo	87
5.2.	Resultados en problemas multi-objetivo	87
5.3.	Explicabilidad de los resultados	88
5.4.	Trabajo futuro	89

Índice de figuras

2.1. Ilustración de la representación de la dominancia de Pareto del punto x8.	12
2.2. Ilustración de un problema bi-objetivo con el frente de Pareto destacado mediante una línea en negrita.	13
2.3. Ilustración de la definición del campo de normalización usando el punto Ideal y de Nadir	32
4.1. Gráfico de convergencia para el algoritmo genético en problemas de baja dimensionalidad.	58
4.2. Gráfico de convergencia para el algoritmo genético en problemas de alta dimensionalidad.	58
4.3. Evolución Diferencial - Dos dimensiones	58
4.4. Evolución Diferencial - Cuarenta dimensiones	59
4.5. Estrategias Evolutivas - Dos dimensiones	59
4.6. Estrategias Evolutivas - Cuarenta dimensiones	59
4.7. Gráfica de diferencias críticas mono-objetivo	65
4.8. Mapa de correlación de características de los espacios mono-objetivo . .	67
4.9. Análisis Mono-Objetivo de SHAP destacando las características más influyentes en la elección de heterogeneidad.	68
4.10. NSGA-III - Hipervolumen - Dos dimensiones	72
4.11. NSGA-III - Hipervolumen - Cuarenta dimensiones	72
4.12. NSGA-III - Boxplots - Dos dimensiones	73
4.13. NSGA-III - Boxplots - Cuarenta dimensiones	73
4.14. Gráfica de diferencias críticas bi-objetivo	81
4.15. Gráfica de diferencias críticas bi-objetivo con archivo externo	82
4.16. Gráfica de diferencias críticas multi-objetivo	82
4.17. Mapa de correlación de características de los espacios Bi-Objetivo . . .	84
4.18. Análisis Bi-Objetivo de SHAP destacando las características más influyentes en la elección de heterogeneidad.	85

Índice de tablas

4.1. Resumen del análisis empírico mono-objetivo para algunos problemas en particular	57
4.2. Resumen de la prueba de Kruskal Wallis para algunos problemas	60
4.3. Resumen de la prueba de Willcoxon ranksum en evolución diferencial para algunos problemas	62
4.4. Resumen de la prueba de Wilcoxon ranksum en estrategias evolutivas para algunos problemas	63
4.5. Resumen de la prueba de Willcoxon ranksum en el algoritmo genético para algunos problemas	64
4.6. Resumen de conteos de Borda	66
4.7. Resumen de resultados BiBBOB con hipervolumen	69
4.8. Resumen de resultados WFG con hipervolumen	70
4.9. Resumen de resultados WFG con NSGA-II en IGD+	71
4.10. Resumen de resultados WFG con NSGA-III en IGD+	71
4.11. Resumen de la prueba BiBBOB de Kruskal Wallis con NSGA-II en hipervolumen	74
4.12. Resumen de la prueba BiBBOB de Kruskal Wallis con NSGA-III en hipervolumen	75
4.13. Resumen de la prueba WFG de Kruskal Wallis con NSGAII	76
4.14. Resumen de la prueba WFG de Kruskal Wallis NSGAIII	77
4.15. Resumen de la prueba BiBBOB de Wilcoxon NSGAII en 2 dimensiones	78
4.16. Resumen de la prueba BiBBOB de Wilcoxon NSGAII en 40 dimensiones	79
4.17. Resumen de la prueba BiBBOB de Wilcoxon NSGAIII en 2 dimensiones	80
4.18. Resumen de la prueba BiBBOB de Wilcoxon NSGAIII en 40 dimensiones	81
4.19. Tabla de resultados del conteo de Borda en BiBBOB	83
4.20. Tabla de resultados del conteo de Borda en WFG con hipervolumen . .	83
4.21. Tabla de resultados del conteo de Borda en WFG con IGD+	83

Capítulo 1

Introducción

Los algoritmos evolutivos, tanto mono-objetivo como multi-objetivo (MOEAs - por sus siglas en inglés), de última generación destacan por su enfoque elitista. Estos algoritmos suelen implementar una selección determinista ($\mu + \lambda$), en la cual se eligen exclusivamente los mejores individuos para avanzar a la siguiente generación. En el contexto mono-objetivo, esta selección se centra en optimizar un único criterio, mientras que en el multi-objetivo se busca un equilibrio entre múltiples criterios. No obstante, en ambos enfoques se ha identificado la importancia de fomentar la diversidad dentro de la población. Esto permite una exploración más amplia del espacio de búsqueda, vital para identificar soluciones óptimas en problemas complejos, ya sea enfocándose en un único objetivo o equilibrando entre varios.

1.1. Motivación

En la era actual, dominada por avances significativos en informática y tecnología, la optimización efectiva de procesos y la toma de decisiones eficaz son cruciales para superar desafíos en una amplia gama de disciplinas, desde la ingeniería, la investigación operativa y la ciencia de datos. Dentro de este contexto, tanto los algoritmos evolutivos mono-objetivo como los MOEAs desempeñan un papel indispensable en problemas no lineales. Estos algoritmos se han convertido en herramientas clave para abordar problemas de optimización complejos, donde la tarea puede involucrar la mejora de un único criterio o el equilibrio entre múltiples criterios simultáneamente, lo cual es esencial en situaciones donde se requiere considerar diversas metas concurrentemente.

La motivación detrás de esta tesis radica en explorar el efecto de la heterogeneidad temporal en los algoritmos evolutivos, tanto en su aplicación mono como multi-objetivo, en cómo puede enfatizar la exploración para descubrir soluciones aceptables en la práctica. En particular, se pone énfasis en la capacidad de los MOEAs para identificar conjuntos de soluciones cercanas a óptimas, conocidas como soluciones de Pareto, en escenarios donde se busca un compromiso entre múltiples objetivos. Simultáneamente, se investigará cómo los algoritmos mono-objetivo contribuyen a la optimización enfocada, abordando problemas donde un único objetivo predomina.

Esta investigación aspira a aportar conocimiento nuevo mediante la exploración de esta nueva perspectiva en aplicación de algoritmos evolutivos, tanto mono como multi-objetivo. Se busca avanzar en la comprensión de cómo estas herramientas pueden ser empleada para enfrentar los desafíos de optimización que presentan las disciplinas mencionadas, ofreciendo soluciones que puedan ser aplicadas en la resolución de problemas complejos y multifacéticos.

1.2. El problema

La mayoría de los MOEAs de última generación han adoptado estrategias altamente elitistas que priorizan la selección de los mejores individuos en cada generación, lo que comúnmente se denomina selección $(\mu + \lambda)$. Si bien esta estrategia ha demostrado ser efectiva en la explotación de soluciones óptimas, puede resultar insuficiente en situaciones donde la exploración del espacio de búsqueda es esencial [8, 36, 35, 53, 30]. En tales contextos, es necesario fomentar una mayor diversidad en la población para descubrir soluciones no dominadas adicionales, evitar la convergencia prematura hacia un óptimo local y además de añadir adaptabilidad a diferentes tipos de problemas.

1.3. Hipótesis

La heterogeneidad temporal influye significativamente en el equilibrio entre la exploración y la explotación en algoritmos evolutivos multi-objetivo, mediante la incorporación de un parámetro que permita transitar entre estrategias $(\mu + \lambda)$, orientadas hacia la explotación, y estrategias (μ, λ) , más inclinadas hacia la exploración, optimizando así el desempeño del algoritmo en entornos dinámicos [40].

Este estudio se propone investigar el impacto de la heterogeneidad temporal mediante la incorporación de un parámetro que ajusta la presión de selección, variando desde estrategias $(\mu + \lambda)$, que inclinan la balanza hacia la explotación, hasta estrategias (μ, λ) , que promueven la exploración.

1.4. Objetivos

En el presente trabajo, se busca abordar el impacto de la heterogeneidad temporal en los algoritmos evolutivos, explorando cómo esta característica puede influir en el equilibrio entre la exploración y la explotación durante la búsqueda de soluciones óptimas. La investigación se enfoca tanto en contextos mono-objetivo como multi-objetivo, proponiendo la introducción de un parámetro adicional que modula la presión de selección como una vía para mejorar el desempeño de estos algoritmos. A continuación, se detallan los objetivos generales y específicos que guiarán este estudio.

1.4.1. Generales

Estudiar el impacto de la heterogeneidad temporal en el equilibrio entre exploración y explotación en algoritmos evolutivos, tanto en contextos mono-objetivo como multi-objetivo, mediante la introducción de un parámetro adicional que modula la presión de selección.

1.4.2. Particulares

- Analizar la influencia de la heterogeneidad temporal en la efectividad de los algoritmos evolutivos para encontrar soluciones óptimas en problemas tanto de optimización mono-objetivo como multi-objetivo.
- Diseñar y validar un mecanismo adaptativo que permita ajustar la presión de selección en algoritmos evolutivos, evaluando su efecto en estrategias de selección que van desde $(\mu + \lambda)$, enfocadas en la explotación, hasta (μ, λ) , orientadas hacia la exploración.
- Realizar experimentos comparativos para evaluar el rendimiento de los algoritmos evolutivos con este parámetro de presión de selección ajustable, en una variedad de problemas de optimización, tanto mono como multi-objetivo.
- Investigar cómo la adaptación de la presión de selección basada en la heterogeneidad temporal mejora la convergencia hacia soluciones óptimas y la diversidad de soluciones en el espacio de búsqueda.
- Desarrollar guías para la aplicación y ajuste del parámetro de presión de selección en algoritmos evolutivos, considerando las peculiaridades de los problemas de optimización a los que se aplican, incluyendo el análisis de la complejidad del espacio de búsqueda y la interacción entre objetivos.

1.5. Contribuciones y productos de investigación

Esta investigación busca contribuir a la comprensión y mejora de las estrategias de selección utilizadas en MOEAs, promoviendo la adaptabilidad de estos algoritmos en función de la naturaleza y complejidad de los problemas abordados. A continuación, se presentan las principales contribuciones y productos derivados de este trabajo:

- La adaptación de algoritmos para considerar el nuevo parámetro.
- Generar una amplia base de datos de resultados estadísticos de la introducción del nuevo parámetro.
- Un paquete de software para pruebas y ajuste fuera de línea en GitHub.
- Una publicación realizada en el evento de GECCO 2023 [47].

1.6. Organización de la tesis

Esta tesis se compone de cinco capítulos, cada uno de los cuales se enfoca en aspectos específicos de la investigación en el campo de algoritmos evolutivos y la adaptación de algoritmos evolutivos. A continuación, se presenta una breve descripción de cada capítulo:

- Capítulo 1: Introducción
 - Este capítulo introductorio establece el contexto y los objetivos de la investigación.
 - Se plantea la importancia de los algoritmos evolutivos en la optimización multi-objetivo y se delinea la estructura de la tesis.
- Capítulo 2: Marco teórico
 - En este capítulo, se presentan los conceptos básicos de la optimización multi-objetivo y los sistemas complejos fundamentales para comprender la investigación.
 - Se incluye un análisis de trabajos relevantes en el estado del arte relacionados con la investigación y sus propuestas.
- Capítulo 3: Metodología
 - Este capítulo detalla la metodología de la investigación, describiendo los entornos de desarrollo utilizados.
 - Se proporciona información sobre los parámetros aplicados en el desarrollo y se describen los mecanismos de selección modificados de los algoritmos utilizados.
- Capítulo 4: Análisis de resultados
 - En este capítulo se realiza un análisis detallado de los resultados obtenidos en la investigación.
 - Se presentan tablas de resultados y comparaciones importantes de los objetivos estudiados.
- Capítulo 5: Conclusiones y trabajo futuro
 - El capítulo de conclusiones resume las principales conclusiones extraídas de la investigación.
 - Se presentan propuestas de trabajo futuro que podrían desarrollarse a partir de los hallazgos de esta tesis.

Capítulo 2

Marco Teórico

En este capítulo, se introducen los principios fundamentales y conceptos clave para comprender la investigación en el campo de la optimización a través de algoritmos evolutivos, abarcando tanto la optimización mono-objetivo como la multi-objetivo. Los algoritmos evolutivos son herramientas poderosas que se aplican para optimizar una amplia gama de problemas, utilizando estrategias de exploración para descubrir nuevas áreas del espacio de búsqueda y estrategias de explotación para afinar y mejorar las soluciones existentes hacia óptimos locales o globales. Estas estrategias son vitales en ambos contextos de optimización; en el mono-objetivo, se enfocan en encontrar la mejor solución posible para un único criterio, mientras que en el multi-objetivo, buscan un conjunto de soluciones que representen un equilibrio óptimo entre varios criterios contradictorios.

2.1. Optimización

La optimización es el proceso de encontrar la mejor solución o resultado posible para un problema dado. Enfocándonos en la optimización multi-objetivo, esta aborda el desafío de tomar decisiones en contextos donde múltiples criterios conflictivos deben ser considerados simultáneamente. Los problemas multi-objetivos requieren un enfoque más complejo debido a la naturaleza intrínsecamente conflictiva de los objetivos involucrados. Esta sección revisa los fundamentos de la optimización multi-objetivo, destacando los conceptos de dominancia de Pareto y el Frente de Pareto, así como la importancia de la diversidad de soluciones en la toma de decisiones. Se discuten también métodos de solución y las implicaciones de la diversidad de soluciones en la práctica.

2.1.1. Fundamentos de la optimización multi-Objetivo

Un problema de optimización multi-objetivo, también denominado problema de optimización multi-criterio o de múltiples objetivos, se refiere a una categoría de problemas de optimización que requieren la optimización simultánea de más de un objetivo [6].

Estas se definen como:

encontrar el vector $\vec{x}^* = [x_1^*, x_2^*, \dots, x_n^*]^T \in F$ que satisfaga las m restricciones de desigualdad:

$$g_i(\vec{x}) \preceq 0 \quad i = 1, 2, \dots, m, \quad (2.1)$$

las p restricciones de igualdad:

$$h_j(\vec{x}) = 0 \quad j = 1, 2, \dots, p, \quad (2.2)$$

y optimiza la función vectorial:

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

donde $\vec{x} = [x_1, x_2, \dots, x_n]^T$ es el vector de variables de decisión y F es la región factible definida por las restricciones p y m .

Estos problemas son prevalentes en una amplia gama de campos, incluidos la ingeniería, la economía y la logística, en los cuales es necesario tomar decisiones que incorporen varios criterios, los cuales frecuentemente presentan conflictos entre sí. Cabe destacar que, en el caso de que el problema involucre un único objetivo, este se clasifica como un problema de optimización mono-objetivo.

2.1.2. Dominancia de Pareto

La dominancia de Pareto es un concepto central en el análisis de decisiones multi-objetivo y en la teoría de juegos, nombrado así por el economista italiano Vilfredo Pareto. Se utiliza para comparar diferentes soluciones en problemas donde hay múltiples objetivos a considerar, y estos objetivos a menudo entran en conflicto entre sí [10].

A lo cual decimos que un vector $\vec{x} = [x_1, \dots, x_k]^T$ se dice que domina a otro vector $\vec{y} = [y_1; \dots; y_k]^T$ (denotado por $\vec{x} \preceq \vec{y}$) si y sólo si x es parcialmente menor que y :

$$\forall i \in \{1, \dots, k\}, x_i \preceq y_i \quad \text{y} \quad \exists i \in \{1, \dots, k\} : x_i \prec y_i. \quad (2.4)$$

De tal manera, para dos vectores de decisión en un problema multi-objetivo, se dice que:

- \vec{x} domina fuertemente a \vec{y} (denotado por $\vec{x} \prec \vec{y}$) si $\forall i \in \{1, \dots, k\} : f_i(\vec{x}) \prec f_i(\vec{y})$.
- \vec{x} domina débilmente a \vec{y} (denotado por $\vec{x} \preceq \vec{y}$) si $\forall i \in \{1, \dots, k\} : f_i(\vec{x}) \preceq f_i(\vec{y})$.

La dominancia de Pareto ayuda a identificar un conjunto de soluciones óptimas en el sentido de que ninguna otra solución en consideración es mejor en todos los objetivos simultáneamente, como se puede apreciar en la Figura 2.1.

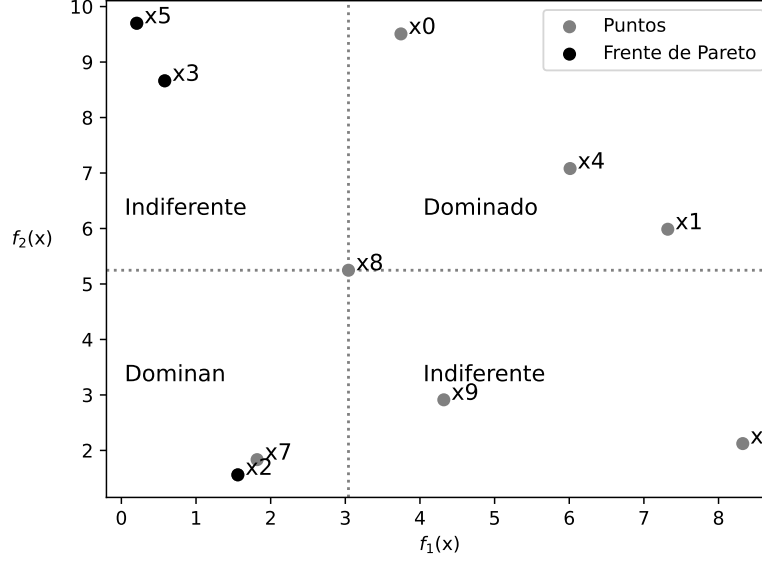


Figura 2.1: Ilustración de la representación de la dominancia de Pareto del punto x_8 .

No dominado

Un punto (o solución) se considera no dominado si no existe otro punto que mejore en todos los criterios o dimensiones evaluados vistos en la Ecuación 2.4. Esto significa que, para ser "no dominado", un punto debe ser al menos tan bueno como cualquier otro en todos los criterios y mejor en al menos uno. En el contexto de la optimización multi-objetivo, identificar soluciones no dominadas es crucial para entender el trade-off entre diferentes objetivos, ya que mejorar en un objetivo puede significar empeorar en otro.

$$\forall i \in \{i, \dots, k\} : (f_i(\vec{x}) \not\leq f_i(\vec{y})) \wedge \forall i \in \{i, \dots, k\} : (f_i(\vec{y}) \not\leq f_i(\vec{x})). \quad (2.5)$$

Conjunto de óptimos de Pareto

El conjunto de óptimos de Pareto, es un concepto que extiende la idea de soluciones no dominadas al definir un estado de asignaciones de recursos en el cual no es posible mejorar la situación de un individuo sin empeorar la situación de otro. En términos de soluciones a problemas, un punto es un Óptimo de Pareto si no es posible moverse a otro punto que mejore al menos un objetivo sin empeorar al menos otro objetivo. Por lo mismo el conjunto de óptimos de Pareto consiste en el conjunto de todas las

soluciones óptimas de Pareto, es decir, aquellas soluciones que no son dominadas por ninguna otra solución en el espacio de las variables de decisión [43].

$$SP^* = \{\vec{x} \in F \mid \nexists y \in F, \vec{f}(\vec{y}) \not\leq \vec{f}(\vec{x})\}. \quad (2.6)$$

donde SP^* es el conjunto de Óptimos de Pareto y \vec{x} y \vec{y} son vectores de decisión que pertenecen al conjunto de soluciones factibles F .

Frente de Pareto

El conjunto de óptimos de Pareto está definido en el espacio de las variables de decisión. La evaluación de dicho conjunto en el espacio de las funciones objetivo es conocido como frente de Pareto (FP) [17]. Estas soluciones se visualizan como en la Figura 2.2.

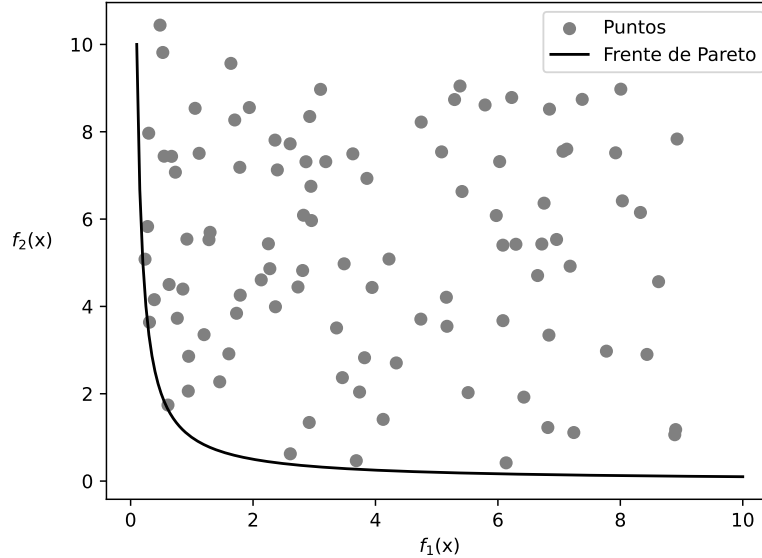


Figura 2.2: Ilustración de un problema bi-objetivo con el frente de Pareto destacado mediante una línea en negrita.

El frente de Pareto es una herramienta crucial para los tomadores de decisiones en campos como la economía, la ingeniería, la gestión de proyectos y la planificación ambiental. Al proporcionar un conjunto de soluciones óptimas desde el punto de vista de la eficiencia de Pareto, facilita la identificación de las mejores opciones posibles bajo criterios múltiples y conflictivos. La selección final de una solución dentro del Frente de Pareto suele requerir consideraciones adicionales, como las preferencias personales o institucionales, valores éticos o criterios de mantenibilidad.

2.1.3. Diversidad de soluciones

La diversidad de soluciones en el contexto de problemas de optimización multi-objetivo y, específicamente, al referirse al frente de Pareto, destaca la existencia de múltiples soluciones óptimas que ofrecen distintas compensaciones entre los objetivos en conflicto [6]. En problemas donde se deben considerar varios criterios simultáneamente, es raro encontrar una solución única que sea la mejor en todos los aspectos. En cambio, lo que se encuentra es un conjunto de soluciones óptimas en el sentido de Pareto, cada una de las cuales es inmejorable sin empeorar en algún otro objetivo.

2.2. El balance entre exploración y explotación

En diversas situaciones, nos encontramos ante la necesidad de tomar decisiones que implican encontrar un equilibrio delicado entre dos estrategias fundamentales. Por un lado, está la exploración de opciones desconocidas, que busca adquirir información adicional y descubrir nuevas soluciones. Como señala Holland [24], esta exploración es fundamental en la adaptación y evolución, permitiendo la identificación de posibles soluciones innovadoras y eficaces. Por otro lado, se encuentra la explotación de opciones conocidas, que busca obtener recompensas inmediatas mediante la mejora de soluciones ya existentes. Goldberg [19] enfatiza la importancia de un equilibrio adecuado entre exploración y explotación en algoritmos evolutivos, argumentando que una inclinación excesiva hacia la explotación puede llevar a la convergencia prematura y limitar el alcance de la búsqueda.

2.2.1. El impacto en MOEAs

La dualidad entre la exploración y la explotación adquiere una relevancia particular en el contexto de los problemas multi-objetivo, que son inherentemente de naturaleza matemática. Estos problemas se caracterizan por la necesidad de optimizar simultáneamente múltiples funciones objetivo (véase en la Ecuación 2.4). En este marco, las técnicas del cómputo evolutivo emergen como herramientas esenciales para abordar y resolver dichos problemas, que plantean un desafío significativo debido a la complejidad que surge de sus múltiples criterios de optimización y las restricciones asociadas.

Kalyanmoy Deb [10] explora este balance en el contexto de MOEAs, demostrando cómo diferentes estrategias de selección pueden influir significativamente en el rendimiento del algoritmo. La elección entre métodos de selección más elitistas o aquellos que promueven la diversidad puede tener un impacto directo en la capacidad del algoritmo para explorar eficientemente el espacio de búsqueda.

En los Algoritmos Evolutivos Multiobjetivo (MOEAs) existentes, se ha asumido tradicionalmente que la selección basada en el rendimiento en el espacio objetivo es suficiente para mantener la diversidad en el espacio de búsqueda. Sin embargo, esta perspectiva ha evolucionado con el reconocimiento de que un equilibrio entre exploración y explotación es esencial para el éxito de estos algoritmos. A la fecha, se han incorporado

múltiples mecanismos explícitos en los MOEAs para controlar este equilibrio de manera efectiva. Estos mecanismos incluyen, entre otros, técnicas de nicho, métodos de selección basados en diversidad, operadores de mutación y cruce diversificados, así como estrategias adaptativas y dinámicas. Estas innovaciones están diseñadas para asegurar que, además de optimizar en el espacio objetivo, los algoritmos mantengan una diversidad adecuada en el espacio de búsqueda. Esto es crucial para prevenir la convergencia prematura hacia soluciones subóptimas y para facilitar la exploración exhaustiva del espacio de soluciones disponibles. Por lo tanto, es más preciso afirmar que los MOEAs modernos están equipados con una variedad de herramientas que les permiten gestionar activamente la diversidad de la población, asegurando un equilibrio óptimo entre la exploración de nuevas áreas del espacio de búsqueda y la explotación de soluciones prometedoras ya descubiertas. En la práctica, Jianyong y Hu Zhang [57] han encontrado que la colaboración de múltiples operadores de recombinación puede adaptarse a la forma y las propiedades locales del paisaje de aptitud, lo que contribuye a equilibrar la exploración y la explotación de manera efectiva.

Dentro del campo del cómputo evolutivo, el tratamiento del dilema entre exploración y explotación se realiza mediante la implementación de dos estrategias evolutivas fundamentales, que son $(\mu + \lambda)$ y (μ, λ) . La estrategia $(\mu + \lambda)$, donde μ representa el número de padres en la población actual y λ denota el número de descendientes generados, se orienta hacia una exploración intensiva del espacio de búsqueda. En esta estrategia, se seleccionan los mejores individuos de la unión de padres y descendientes para la próxima generación, lo cual promueve una búsqueda informada y una convergencia gradual hacia soluciones óptimas al mantener una mezcla de generaciones.

Por el contrario, la estrategia (μ, λ) introduce un componente de aleatorización más significativo, ya que únicamente los λ descendientes compiten para convertirse en los próximos μ padres, sin que los padres actuales tengan la oportunidad de pasar directamente a la siguiente generación. Esto favorece la diversidad genética y evita la convergencia prematura, permitiendo una mayor adaptabilidad y flexibilidad en la búsqueda de soluciones. La ausencia de los padres actuales en la selección para la próxima generación subraya la importancia de la variabilidad y la capacidad de la población para adaptarse a cambios o descubrir nuevas regiones del espacio de búsqueda.

El desafío de encontrar el equilibrio óptimo entre exploración y explotación representa un aspecto crucial en la optimización de problemas multi-objetivo. Aunque se ha realizado una investigación exhaustiva sobre las estrategias (μ, λ) y $(\mu + \lambda)$, persisten avances significativos en este campo, con el objetivo de explorar nuevas ideas y capitalizar el conocimiento ya existente.

2.2.2. Descripción de selección de siguiente población

La selección en los algoritmos evolutivos [2] presenta dos variantes fundamentales, cada una influyendo en el proceso evolutivo de manera distinta. Dependiendo de si la población de padres en la generación actual se incluye en el proceso de selección, se

distinguen dos estrategias: la selección de más, denotada por $(\mu + \lambda)$, y la selección de coma, representada como (μ, λ) .

Estrategia (μ, λ)

En la selección (μ, λ) , solo los λ individuos recién generados, es decir, la población actual, contribuyen al grupo de selección. En este enfoque, los padres de la generación actual son olvidados, incluso si superan a toda la descendencia. Es evidente que una condición necesaria para el algoritmo es que $\mu < \lambda$. La igualdad $\mu = \lambda$ resulta en que toda la descendencia se selecciona como padres, lo que conduce a una pérdida de información relevante para la búsqueda y, como resultado, la población realiza un paseo aleatorio en el espacio de búsqueda.

Estrategia $(\mu + \lambda)$

Por otro lado, la selección de mas, representada por $(\mu + \lambda)$, incorpora a los padres en el proceso de selección. La notación indica que tanto los padres como la descendencia se copian en el grupo de selección, que tiene un tamaño $\gamma = \mu + \lambda$. A diferencia de la selección de coma, no hay restricción teórica sobre el número de descendientes λ . Casos con $\mu = \lambda$ o $\mu > \lambda$ son posibles. El caso especial $(\mu + 1)$, conocido como algoritmos evolutivos de estado uniforme, se emplea comúnmente en implementaciones asíncronas en sistemas multiprocesador[37]. La selección de más asegura la supervivencia del mejor individuo encontrado hasta el momento, y debido a esta preservación, estas técnicas de selección son denominadas elitistas. El elitismo se erige como una condición necesaria que un operador de selección debe cumplir para demostrar la propiedad de convergencia global en los algoritmos evolutivos. Gracias al elitismo en las estrategias de más, los padres pueden perdurar indefinidamente y mantener a la solución óptima en caso de ser encontrada.

Ambas variantes de selección encuentran aplicaciones específicas. La selección (μ, λ) se recomienda para espacios de búsqueda no acotados [7], especialmente cuando $Y = R^N$, mientras que la selección $(\mu + \lambda)$ se utiliza en espacios de búsqueda discretos de tamaño finito [39], como en problemas de optimización combinatoria [18].

2.3. Descripción de los algoritmos

En los algoritmos evolutivos, la diversidad de la población en el espacio de búsqueda se utiliza comúnmente para medir y controlar el equilibrio entre exploración y explotación. La exploración es posible si la población es diversa, y se espera que esta diversidad disminuya a medida que se realiza la explotación.

En esta sección, se proporciona una visión detallada de los algoritmos evolutivos y su aplicación en la resolución de problemas de optimización. Se abordará la mecánica fundamental de estos algoritmos, destacando las diferencias en su enfoque dependiendo de si el problema es de naturaleza mono-objetivo o multi-objetivo. Se explorará la

importancia de las operaciones clave como la recombinación y la selección ambiental, y cómo éstas contribuyen al equilibrio entre la exploración del espacio de búsqueda y la explotación de las soluciones encontradas.

2.3.1. Mono-objetivo

Los algoritmos evolutivos (EAs) son una familia de métodos de optimización inspirados en el proceso natural de la evolución, y utilizan una serie de operaciones bioinspiradas para explorar el espacio de soluciones de un problema. La recombinación, o cruce, es una de estas operaciones esenciales y consiste en combinar partes de dos o más soluciones parentales para producir una nueva solución, con la esperanza de heredar las características deseables de cada uno de los padres.

Además de la recombinación, otros operadores importantes en los EAs incluyen la mutación y la selección. La mutación introduce variabilidad al azar en las soluciones, permitiendo que el algoritmo explore nuevas regiones del espacio de búsqueda que no se generarían a través de la recombinación sola. Esta operación es crucial para mantener la diversidad genética de la población y evitar que el algoritmo se estanque en óptimos locales.

La selección, por su parte, es el proceso por el cual se decide qué soluciones se mantendrán para la siguiente generación. La selección ambiental es una forma de selección que se realiza después de que la recombinación y la mutación hayan tenido lugar, evaluando las soluciones recién creadas y determinando su idoneidad para sobrevivir en base a su aptitud o *fitness*. Esta aptitud suele medirse en función de cuán bien la solución satisface los objetivos del problema de optimización.

Otro operador involucrado en los EAs es la selección de padres, que determina qué individuos de la población actual se utilizarán para la creación de descendientes a través de recombinación y mutación. La selección de padres a menudo se realiza de tal manera que las soluciones con mejor aptitud tengan una mayor probabilidad de ser elegidas, aunque también se pueden emplear estrategias que promuevan la diversidad.

Finalmente, los EAs pueden incluir mecanismos de reemplazo para decidir cómo las soluciones descendientes reemplazarán a las soluciones parentales en la población. El objetivo de estos mecanismos es encontrar un equilibrio adecuado entre preservar las soluciones de alta calidad y permitir la introducción de nuevas soluciones potencialmente prometedoras.

La combinación de estos operadores —recombinación, mutación, selección de padres, selección ambiental y mecanismos de reemplazo— permite que los algoritmos evolutivos simulen el proceso de evolución natural y se adapten continuamente a medida que buscan soluciones óptimas o satisfactorias para problemas complejos de optimización.

En esta sección veremos un panorama conciso de los algoritmos implementados en

este estudio, enfocándose en sus atributos distintivos, metodologías y sus campos de aplicación.

Algoritmo genético

El algoritmo genético (GA) [25] es un algoritmo de optimización inspirado en la teoría de la evolución de Darwin. Es un algoritmo de búsqueda basado en población que utiliza el concepto de supervivencia de los más aptos. Las nuevas poblaciones se producen mediante el uso iterativo de operadores genéticos en los individuos presentes en la población. La representación del cromosoma, la selección, el cruce, la mutación y el cálculo de la función de aptitud son elementos clave del GA.

El procedimiento del GA es el siguiente:

1. **Inicialización:** Se genera una población inicial de N individuos de manera aleatoria, representados por:

$$C_i = \{g_1, g_2, \dots, g_m\}. \quad (2.7)$$

Donde m es el número de genes en el cromosoma y C_i es el i -ésimo cromosoma.

2. **Evaluación:** Cada individuo C_i es evaluado mediante una función de aptitud $f(C_i)$, que mide la calidad de la solución representada por el cromosoma.
3. **Selección:** Los individuos son seleccionados para la reproducción con una probabilidad proporcional a su aptitud, dada por:

$$P(C_i) = \frac{f(C_i)}{\sum_{j=1}^N f(C_j)}. \quad (2.8)$$

4. **Cruzamiento (Crossover):** Pares de individuos son seleccionados para producir descendencia con una probabilidad de cruzamiento p_c . Los genes son intercambiados entre los padres para formar descendientes.
5. **Mutación:** Con una probabilidad de mutación p_m , se realizan cambios aleatorios en los genes de los individuos descendientes:

$$g'_i = g_i + \delta. \quad (2.9)$$

Donde g'_i es el gen mutado, g_i es el gen original, y δ es una alteración aleatoria.

6. **Reemplazo:** La nueva generación de individuos reemplaza a la generación anterior, según la estrategia de reemplazo elegida.
7. **Terminación:** El algoritmo repite los pasos 2 a 6 hasta que se cumple uno o más criterios de terminación, como alcanzar un número máximo de generaciones G_{\max} , lograr una aptitud por encima de un umbral $f_{\text{threshold}}$, o una mejora mínima entre generaciones.

Los algoritmos genéticos cambian dinámicamente el proceso de búsqueda a través de las probabilidades de cruce y mutación y alcanzan la solución óptima. Los GA pueden modificar los genes codificados. Los GA pueden evaluar múltiples individuos y producir múltiples soluciones óptimas. Por lo tanto, los GA tienen una mejor capacidad de búsqueda global. La descendencia producida a partir del cruce de cromosomas parentales tiene la probabilidad de eliminar los esquemas genéticos presentes en los cromosomas de los padres. Un esquema genético, en el contexto de los algoritmos evolutivos, se refiere a un patrón o conjunto de genes que se mantiene en varias generaciones y que contribuye de manera significativa al rendimiento del individuo. La fórmula utilizada para el cruce se define como:

$$\beta = \begin{cases} (2u)^{\frac{1}{\eta_c+1}}, & \text{si } u \leq 0,5, \\ \left(\frac{1}{2(1-u)}\right)^{\frac{1}{\eta_c+1}}, & \text{si } u > 0,5. \end{cases} \quad (2.10)$$

Tal que:

$$y_1 = 0,5 \left((1 + \beta)x_1 + (1 - \beta)x_2 \right), \quad (2.11)$$

$$y_2 = 0,5 \left((1 - \beta)x_1 + (1 + \beta)x_2 \right). \quad (2.12)$$

Donde:

- u es un número aleatorio entre 0 y 1.
- η_c es el parámetro de distribución, que controla la forma de la distribución SBX.
- β es el factor de distribución, calculado a partir de u y η_c .
- x_1 y x_2 son los valores de los padres.
- y_1 y y_2 son los valores calculados para los descendientes.

Según el teorema del esquema, el esquema original debe ser reemplazado por un esquema modificado. Para mantener la diversidad en la población, el nuevo esquema conserva la población inicial durante las primeras etapas de la evolución. Al final de la evolución, se producirá el esquema apropiado para evitar cualquier distorsión de los excelentes esquemas genéticos.

Evolución diferencial

El algoritmo de evolución diferencial (DE) [13] se considera como uno de los optimizadores más destacados y populares para abordar problemas de optimización continua. Este algoritmo pertenece a la familia de los Algoritmos Evolutivos y se destaca por su capacidad para resolver una amplia gama de problemas de optimización. La DE trabaja con poblaciones de soluciones y utiliza un enfoque de recombinación para generar nuevas descendencias bajo ciertas condiciones.

A diferencia de algunos otros EAs que generan descendencia al perturbar las soluciones con vectores de diferencia escalados, la DE se enfoca en la recombinación de soluciones existentes. En este proceso, la solución individual actual puede ser reemplazada si es superada por la nueva solución descendiente. Esto le otorga a la DE su característica de robustez y simplicidad, ya que su proceso de búsqueda se rige por un número reducido de parámetros específicos del algoritmo, como el factor de escala y la tasa de cruce.

Al igual que otros EA, DE opera a través de tres mecanismos clave: mutación, cruce y selección. Entre estos mecanismos, la mutación y el cruce desempeñan un papel fundamental en el rendimiento de búsqueda del algoritmo.

■ **DE/rand/1:**

$$v_{i,G} = x_{r1,G} + F \cdot (x_{r2,G} - x_{r3,G}). \quad (2.13)$$

■ **DE/best/1:**

$$v_{i,G} = x_{best,G} + F \cdot (x_{r1,G} - x_{r2,G}). \quad (2.14)$$

■ **DE/rand-to-best/1:**

$$v_{i,G} = x_{r1,G} + F \cdot (x_{best,G} - x_{r1,G}) + F \cdot (x_{r2,G} - x_{r3,G}). \quad (2.15)$$

■ **DE/current/1:**

$$v_{i,G} = x_{i,G} + F \cdot (x_{r1,G} - x_{r2,G}). \quad (2.16)$$

■ **DE/current-to-best/1:**

$$v_{i,G} = x_{i,G} + F \cdot (x_{best,G} - x_{i,G}) + F \cdot (x_{r1,G} - x_{r2,G}). \quad (2.17)$$

En estas fórmulas, $x_{best,G}$ representa el mejor individuo de la población en la generación G , y $x_{i,G}$ es el individuo actual siendo mutado. Los índices $r1$, $r2$, $r3$, e i son seleccionados del conjunto $\{1, 2, 3, \dots, NP\}$, donde NP es el tamaño total de la población, asegurando que $r1 \neq r2 \neq r3 \neq i$. Estos índices representan diferentes individuos seleccionados aleatoriamente de la población para participar en la mutación, promoviendo la diversidad genética.

El factor de escala F es un parámetro crucial que influye en la magnitud de la mutación aplicada a las soluciones. F es un número real positivo, generalmente entre 0 y 2, que ajusta el efecto de perturbación de la operación de mutación. Un F bajo favorece una explotación más fina de las soluciones existentes, mientras que un F alto promueve la exploración de nuevas áreas del espacio de búsqueda.

El cruce, por su parte, fusiona las soluciones de manera que se puedan combinar características deseables de múltiples individuos para formar nuevas soluciones. El vector objetivo $u_{i,G} = u_{i,1,G}, u_{i,2,G}, \dots, u_{i,D,G}$ se genera mediante la operación de cruce, esto dado por la operación:

$$u_{i,j,G} = \begin{cases} v_{i,j,G}, & \text{si } \text{rand}(0,1) \preceq CR \text{ o } j = jrand, \\ x_{i,j,G}, & \text{en caso contrario.} \end{cases} \quad (2.18)$$

En esta operación, i representa el índice de un individuo dentro de la población, con $i \in \{1, 2, \dots, NP\}$. El índice j representa la dimensión de un individuo, con $j \in \{1, 2, \dots, D\}$, donde D es la dimensión del problema, es decir, el número de variables de decisión que define el espacio de soluciones del problema.

El término $jrand$ es un entero aleatorio seleccionado en el rango de 1 a D , asegurando que al menos una dimensión del vector mutado $v_{i,j,G}$ sea transferida al vector objetivo $u_{i,j,G}$.

La tasa de cruce, CR , es un parámetro que determina la probabilidad con la que se escogen elementos del vector mutado $v_{i,j,G}$ en lugar de elementos del vector original $x_{i,j,G}$ durante la generación del vector objetivo. CR es un número real en el rango de 0 a 1, donde un valor cercano a 0 significa que la mayoría de los componentes del vector objetivo provendrán del vector original, mientras que un valor cercano a 1 aumenta la probabilidad de que los componentes provengan del vector mutado, fomentando una mayor diversidad en las soluciones generadas.

Estrategias evolutivas

Las Estrategias Evolutivas (ES) [2] se presentan como un enfoque innovador en el ámbito de la optimización y búsqueda heurística, tomando inspiración de los principios evolutivos que observamos en la naturaleza. Este algoritmo ha adquirido una relevancia significativa al abordar problemas complejos y de alta dimensionalidad en diversas disciplinas, como la ingeniería, la inteligencia artificial y la investigación operativa. Su estructura modular, compuesta por elementos clave, define su funcionamiento eficiente y versátil.

Entre estos elementos, destacan la representación de las soluciones, los operadores genéticos de mutación y recombinación, la función de evaluación objetiva y la estrategia de selección, que desempeña un papel crucial en la diversidad de estrategias $(\mu/\rho+, \lambda)$ incorporadas en su diseño.

La notación $(\mu/\rho+, \lambda)$ -ES encapsula la esencia de la estrategia evolutiva, donde:

- μ : Tamaño de la población actual.
- ρ : Número de padres seleccionados para la recombinación.
- λ : Tamaño de la población de descendencia generada por recombinación.
- $+$: El operador más indica que la población de descendencia se combina con la población actual.

- ,: El operador coma indica que la población de descendencia reemplaza la población actual.

El objetivo habitual de una estrategia evolutiva es optimizar alguna función objetivo(s) o de calidad (F) dada con respecto a un conjunto de variables de decisión o parámetros de control $y := (y_1, y_2, \dots)$ – en el contexto de ES – a menudo denominados parámetros de objeto.

2.3.2. Multi-objetivo

En el caso de los MOEAs, es esencial mantener la diversidad de las soluciones tanto en el espacio objetivo como en el espacio de búsqueda. Además, a diferencia de los algoritmos de objetivo único, la diversidad de la población no debe disminuir durante la búsqueda, ya que se requiere para asegurar la diversidad de las soluciones finales en el espacio objetivo.

NSGA-II

Propuesto por Kalyanmoy Deb y sus colaboradores en 2002 [12], (Non-dominated Sorting Genetic Algorithm II), es un algoritmo evolutivo desarrollado para la optimización multi-objetivo. Su diseño se basa en la idea de clasificar las soluciones en frentes no dominados descritos en la Ecuación 2.5, lo que permite identificar las soluciones que son mejores en todos los objetivos en comparación con otras soluciones.

Este criterio de dominancia constituye el cimiento del potencial del algoritmo, destacándose en su capacidad para clasificar la población a través de la evaluación. El proceso de clasificación comprende los siguientes pasos:

- **Asignación de rangos:** Inicialmente, se identifica el conjunto de soluciones que no son dominadas por ninguna otra en la población. Estas forman el primer frente. Este proceso se realiza mediante la comparación de cada solución con las demás, evaluando si alguna domina a otra en función de todos los objetivos. Luego, estas soluciones se eliminan temporalmente de la consideración y se identifica un nuevo conjunto de soluciones no dominadas, formando el segundo frente. Este proceso se repite, creando sucesivos frentes de soluciones, hasta que todas han sido clasificadas.
- **Crowding Distance:** Además del rango, se calcula la *crowding distance* para cada solución. Para cada solución, se calcula su *crowding distance* como la suma de las diferencias normalizadas en los valores de la función objetivo entre sus dos vecinos más cercanos en cada objetivo. Las soluciones en los extremos del frente tienen un *crowding distance* infinito, lo que garantiza su selección. El cálculo del *crowding distance* desempeña un papel crucial al mantener la diversidad en la población y al favorecer soluciones bien distribuidas en el frente de Pareto.

Con estas fases claves, NSGA-II se destaca como una herramienta versátil y eficaz para abordar problemas complejos de optimización multi-objetivo, brindando una perspectiva innovadora para la toma de decisiones en situaciones donde varios objetivos compiten por ser optimizados.

NSGA-III

Desarrollado por Kalyanmoy Deb y sus colaboradores [11] como una respuesta a la necesidad de un algoritmo más eficiente en el manejo de la diversidad y la distribución de las soluciones en el espacio objetivo. NSGA-III (Non-dominated Sorting Genetic Algorithm III) es una extensión significativa del algoritmo NSGA-II, diseñado para manejar problemas de optimización con múltiples objetivos, especialmente en escenarios con más de tres objetivos.

El NSGA-III introduce una serie de mejoras sobre su predecesor, como el uso de puntos de referencia para guiar el proceso de selección en lugar del cálculo de crowding distance y mantener la diversidad en el espacio de objetivos. Esto lo hace particularmente efectivo para problemas con cuatro o más objetivos.

Con esto el desarrollo del algoritmo en su proceso de selección estaría dado de la siguiente manera:

- **Selección Basada en Referencia:** Cada solución se asocia con el punto de referencia más cercano. La generación de estos puntos de referencia se basa en una partición del espacio de objetivos utilizando el hipervolumen, con el fin de cubrir uniformemente todas las direcciones posibles en el frente de Pareto.
- **Asignación de Nichos:** Se asignan nichos a las soluciones en función de su proximidad a los puntos de referencia. Un nicho se refiere a una región del espacios de los objetivos dado por los puntos de referencia.
- **Llenado de Frentes:** Los frentes no dominados se llenan en orden hasta que la capacidad de la población se agota, teniendo en cuenta la asignación de nichos.
- **Selección de Últimas Soluciones:** Si es necesario, las últimas soluciones se eligen basándose en la distancia mínima perpendicular a cada punto de referencia.

Estos procesos aseguran que la población mantenga una diversidad razonable y que las soluciones bien distribuidas en el frente de Pareto sean favorecidas, consolidando así el impacto y la efectividad del NSGA-III en la resolución de problemas de optimización multi-objetivo avanzados.

2.4. Software

La optimización se ha convertido en una herramienta esencial para resolver una amplia gama de problemas complejos en diversos campos, como la ingeniería, la economía, la

ciencia de datos y la investigación operativa. Para abordar estos desafíos, se han desarrollado y perfeccionado una serie de técnicas y algoritmos de optimización. En esta sección, nos enfocamos en dos herramientas de software que han sido fundamentales en el avance de la investigación y la aplicación práctica en el ámbito de la optimización: Pymoo y COCO-Framework. Ambas plataformas ofrecen características únicas y complementarias que facilitan la experimentación, el benchmarking y la implementación de algoritmos de optimización de manera eficiente y efectiva. Al ofrecer una amplia gama de algoritmos y herramientas para el análisis de rendimiento, estas bibliotecas de software permiten a investigadores y practicantes explorar soluciones óptimas para problemas de complejidad creciente, promoviendo así la innovación y el progreso en sus respectivos campos.

2.4.1. Pymoo

Pymoo [3] es una biblioteca de Python de código abierto especializada en la optimización multiobjetivo. Ofrece una amplia gama de algoritmos de optimización, adecuados tanto para problemas de objetivo único como múltiple. Las características clave de Pymoo incluyen:

- **Diversidad de algoritmos:** Pymoo proporciona algoritmos conocidos y ampliamente utilizados como NSGA-II [12], NSGA-III [11], MOEAD [58], así como algoritmos genéticos (GA) [25], evolución diferencial (DE) [13], estrategias evolutivas (ES) [2] y optimización por enjambre de partículas (PSO) [32]. También integra variantes de estos algoritmos y otros métodos más especializados.
- **Problemas de optimización soportados:** La biblioteca maneja una variedad de problemas de optimización, incluyendo aquellos con objetivo único, multiobjetivo, muchos objetivos y dinámicos como son las funciones ZDT [59], DTLZ [1] y WFG [27]. Ofrece soporte para problemas con restricciones y para diferentes tipos de datos, como binarios, discretos y permutaciones.
- **Personalización y flexibilidad:** Pymoo permite a los usuarios personalizar y ampliar su funcionalidad. Esto incluye la definición de problemas personalizados, la implementación de operadores de evolución propios, y la adaptación de algoritmos existentes para satisfacer necesidades específicas.
- **Indicadores de rendimiento y análisis:** La biblioteca incluye indicadores de rendimiento comunes, como el hipervolumen y la distancia generacional, facilitando la evaluación y comparación de las soluciones generadas por los algoritmos de optimización.

Estas características hacen de Pymoo una herramienta valiosa en el campo de la optimización multi-objetivo, proporcionando a los investigadores y practicantes una plataforma versátil para experimentación y desarrollo.

2.4.2. COCO-Framework

COCO (COMparing Continuous Optimisers) [22] es un *framework* de código abierto, diseñado para la evaluación exhaustiva y el *benchmarking* de algoritmos de optimización, facilitando la comparación efectiva mediante un conjunto estandarizado de herramientas y procedimientos, siendo sus características clave:

- **Benchmarking riguroso:** COCO proporciona un conjunto de funciones de prueba y escenarios de *benchmarking* bien definidos. Esto permite evaluar la eficacia y eficiencia de los algoritmos de optimización en una amplia gama de problemas.
- **Plataforma agnóstica:** Es compatible con múltiples lenguajes de programación, incluidos Python, Java, MATLAB, y C, lo que facilita su uso en diversos entornos de investigación y desarrollo.
- **Análisis de datos:** Ofrece herramientas para el análisis detallado de los resultados del *benchmarking*, incluyendo la generación de gráficos y tablas que resumen el rendimiento de los algoritmos evaluados.
- **Reproducibilidad:** Al proporcionar un marco estandarizado para el *benchmarking*, COCO ayuda a garantizar que los resultados sean reproducibles, lo que es esencial para la validación científica.
- **Flexibilidad:** Aunque COCO viene con un conjunto predeterminado de funciones de prueba, también ofrece la flexibilidad para agregar nuevas funciones de prueba, lo que permite a los usuarios adaptar el *framework* a sus necesidades específicas.
- **Comunidad y documentación:** Cuenta con una comunidad activa de usuarios y desarrolladores, así como documentación detallada que facilita su adopción y uso efectivo.

El uso de COCO facilita la comparación directa de nuestros resultados con trabajos previos, contribuyendo así a un cuerpo coherente y comparativo de conocimiento sobre algoritmos de optimización continua. Además, la adopción de este framework apoya la transparencia y la reproducibilidad en la investigación de optimización, principios fundamentales para el avance científico en este campo.

2.5. Problemas de prueba

En el campo de la optimización numérica, el uso de funciones de prueba desempeña un papel crucial al proporcionar un medio estandarizado para evaluar y comparar el rendimiento de diversos algoritmos. Estas funciones, también conocidas como funciones de referencia o funciones objetivo sintéticas, son esenciales en la investigación operativa y la ciencia computacional para simular una amplia gama de problemas de optimización en un entorno controlado. Su principal objetivo es ofrecer escenarios que imiten

las características y desafíos de problemas reales, permitiendo así evaluar la eficiencia, precisión, robustez y otras propiedades críticas de los algoritmos de optimización.

Las funciones de prueba se categorizan típicamente según la naturaleza de los problemas que modelan y pueden ser clasificadas en varias dimensiones: unimodales o multimodales, con ruido o sin ruido, y con restricciones o sin restricciones. Las funciones unimodales, que poseen un único óptimo global, son útiles para evaluar la rapidez de convergencia de un algoritmo. Por otro lado, las funciones multimodales, con múltiples óptimos locales, son esenciales para probar la capacidad de los algoritmos de escapar de óptimos locales subóptimos y encontrar el óptimo global.

Además, las funciones con ruido incorporado simulan la incertidumbre o variabilidad que a menudo se encuentra en las mediciones de datos reales, desafiando la capacidad del algoritmo para manejar perturbaciones y ofrecer soluciones estables. Las funciones con restricciones, por otro lado, reflejan las condiciones limitantes frecuentemente presentes en problemas prácticos, donde las soluciones deben cumplir con ciertos límites o requerimientos.

En este contexto, los problemas de Black-Box Optimization Benchmarking (BBOB) [21] y las funciones Walker-Fournier-Guerin (WFG) [28] se destacan como herramientas útiles en la evaluación de la heurística de optimización en los algoritmos con el nuevo parámetro.

2.5.1. BBOB

Los problemas de Black-Box Optimization Benchmarking (BBOB) [23] son parte de una serie de talleres y una herramienta de evaluación para algoritmos de optimización en dominios continuos y mixtos enteros.

Los problemas BBOB se presentan en varias suites de prueba, cada una enfocada en diferentes aspectos de la optimización:

- **Suite bbob:** Contiene 24 funciones sin ruido en un dominio continuo.
- **Suite bbob-noisy:** Compuesta por 30 funciones con ruido.
- **Suite bbob-biobj:** Incluye 55 funciones bi-objetivo sin ruido, generadas a partir de la suite bbob.
- **Suite bbob-largescale:** Abarca 24 funciones sin ruido en dimensiones de 20 a 640.
- **Suite bbob-mixint:** Contiene 24 funciones sin ruido de tipo mixto entero.
- **Suite bbob-biobj-mixint:** Consta de 92 funciones bi-objetivo sin ruido y de tipo mixto entero.

- **Suite bbob-constrained:** Incluye 10 funciones sin ruido con un número variable de restricciones.

Siendo de nuestro interés en esta sección la suite bbob, las cuales están categorizadas en cinco grupos distintos. Cada grupo está diseñado para evaluar diferentes capacidades de los algoritmos de optimización:

- **Funciones Separables:** Este grupo incluye funciones como la esférica, la elipsoidal separable y Rastrigin, poniendo a prueba la habilidad de los algoritmos para lidiar con la separabilidad de variables.
- **Funciones con Bajo o Moderado Acondicionamiento:** Se centran en evaluar el desempeño de los algoritmos con funciones que presentan condiciones menos exigentes.
- **Funciones con Alto Acondicionamiento y Unimodales:** Dirigidas a evaluar la eficacia de los algoritmos en el manejo de funciones unimodales bajo condiciones de alta dificultad.
- **Funciones Multimodales con Estructura Global Adecuada:** Contemplan funciones con múltiples óptimos locales y una estructura global bien definida.
- **Funciones Multimodales con Estructura Global Débil:** Representan un desafío mayor debido a su estructura global difusa y la presencia de numerosos óptimos locales.

Características comunes:

- **Límites de la Función:** La mayoría de las funciones tienen límites definidos en el intervalo de $[-5, 5]$.
- **Dimensiones:** Todas las funciones son capaces de operar en un rango de dimensiones que va desde 2 hasta 40.
- **Instancia:** Para la consistencia en los experimentos, todas las funciones se evaluaron en su primera instancia.

Excepciones en límites y características únicas: Algunas funciones presentan límites distintos, tales como:

- Rosenbrock (original y rotada) con límites de $[-5, 10]$.
- Schaffer's F7 y su variante con ruido en $[-100, 100]$.
- Schwefel con un rango de $[-500, 500]$.
- Weierstrass en el intervalo de $[-0,5, 0,5]$.

Además, estas funciones se distinguen por propiedades únicas, incluyendo:

- Buche-Rastrigin como asimétrica.
- Attractive Sector con atractores.
- Paso de Esfera caracterizada por su discontinuidad.
- Rosenbrock (original) por su naturaleza no convexa.
- Sharp Ridge con una cresta afilada.
- Weierstrass con una estructura fractal.
- Schaffer's F7 por su ruido inherente.
- Función Compuesta con múltiples componentes.
- Gallagher's Gaussian 21-hi Peaks y 101-hi Peaks con numerosas cimas.
- Katsuura como no separable.
- Lunacek bi-Rastrigin destacando por su bi-modalidad.

Estas *suites* proporcionan una gama diversa de desafíos de optimización y son fundamentales para comparar diferentes algoritmos de optimización. Además, el entorno BBOB se centra en la medición del rendimiento de los algoritmos basándose en el tiempo de ejecución, definido como el número de evaluaciones realizadas en un problema dado, y en la capacidad del algoritmo para alcanzar o superar un valor objetivo específico.

El enfoque de BBOB en la optimización de caja negra es especialmente relevante para situaciones en las que la forma analítica de las funciones objetivo no está disponible o es demasiado compleja, haciendo que la información sobre derivadas sea inaccesible o costosa de obtener. Estos problemas son representativos de muchos desafíos de optimización en el mundo real y proporcionan un marco de referencia valioso para evaluar la eficacia de los algoritmos de optimización.

Esta *suite* está provista en el paquete IOHexperimenter [14], que es una herramienta de *benchmarking* diseñada para la experimentación con Heurísticas de Optimización Iterativas (IOHs).

Construido en C++ e implementado en Python, ofrece una interfaz eficiente entre los problemas de optimización y sus solucionadores. Su enfoque permite un registro granular del proceso de optimización, facilitando el análisis interactivo de datos y acelerando la implementación de un *pipeline* de *benchmarking*. IOHexperimenter es parte del proyecto IOHprofiler y es adecuado para comparar y evaluar heurísticas como la búsqueda local, algoritmos evolutivos y genéticos, y técnicas de optimización bayesiana.

2.5.2. BiBBOB

La *suite* bbob-biobj, conocida también como BiBBOB [5], se especializa en la generación de problemas bi-objetivos. Esta *suite* combina funciones de la *suite* BBOB, seleccionando dos funciones representativas de cada grupo para formar pares de objetivos en una tarea de minimización sin restricciones. La formulación matemática de estos problemas bi-objetivos es la siguiente:

$$\min_{x \in \mathbb{R}^n} F(x) = (f_a(x), f_b(x)). \quad (2.19)$$

Esta selección evita sesgos hacia cualquier grupo específico y garantiza una representación equilibrada. Algunos ejemplos de pares de funciones seleccionadas son:

- En funciones separables: la función esfera f_1 y elipsoide separable f_2 .
- Con acondicionamiento bajo o moderado: sector atractivo f_6 y Rosenbrock original f_8 .
- Unimodales con alto acondicionamiento: cresta aguda f_{13} y sumas de diferentes potencias f_{14} .
- Multimodales con estructura global adecuada: Rastrigin f_{15} y la función F7 de Schaffer con número de condición 10 f_{17} .
- Multimodales con estructura global débil: Schwefel f_{20} y 10^1 picos de Gallagher f_{21} .

Estas funciones no están normalizadas, lo que desafía a los algoritmos de optimización a trabajar eficazmente con objetivos de magnitudes variables. Para la evaluación del rendimiento, se normalizan las funciones usando los puntos ideal y nadir antes de calcular el indicador de hipervolumen.

- Las funciones son no acotadas, pero las soluciones extremas del conjunto de Pareto se encuentran generalmente dentro de un hipercubo definido en el rango de $[-5, 5]$ en cada dimensión del espacio de búsqueda. Aunque el conjunto de Pareto puede extenderse parcialmente fuera de este hipercubo, se espera que se encuentre mayoritariamente dentro de él.
- Cada instancia de función bi-objetivo tiene un ID entero asociado a cada instancia de la función. La relación entre el ID de la instancia K_F^{ID} de una función bi-objetivo $F = (f_a, f_b)$ y los IDs de instancia $K_{f_a}^{ID}$ y $K_{f_b}^{ID}$ de sus respectivas funciones de objetivo único subyacentes, sigue dos condiciones:
 1. Para la primera función objetivo f_a , su ID de instancia es $K_{f_a}^{ID} = 2 \times K_F^{ID} + 1$.
 2. Para la segunda función objetivo f_b , su ID de instancia es $K_{f_b}^{ID} = K_F^{ID} + 1$.

Estas dos condiciones deben cumplirse para todas las dimensiones y funciones en el conjunto $b\text{-}bob \rightarrow biobj$. Si estas condiciones no se cumplen, se incrementa el ID de la instancia de la segunda función objetivo sucesivamente hasta que ambas propiedades se satisfagan.

Además, las instancias de las funciones de prueba se parametrizan, presentando condiciones específicas como la distancia euclidiana mínima entre las soluciones óptimas y entre los puntos ideal y nadir en el espacio objetivo normalizado. Esto asegura una consistencia en la dificultad a través de diferentes instancias y contribuye a la evaluación comparativa robusta de los algoritmos.

2.5.3. WFG

Las funciones WFG son un conjunto de nueve funciones de referencia diseñadas para evaluar algoritmos de optimización multi-objetivo. Desarrolladas por Simon Huband, Phil Hingston, Luigi Barone, y Lyndon While [27], estas funciones son una mejora sobre las funciones de prueba anteriores y se utilizan ampliamente en la investigación de optimización evolutiva y multi-objetivo.

Estas se dividen en varios tipos, cada uno con características únicas. Algunos ejemplos incluyen:

- **WFG1:** Convexa/mixta, multimodal, con transiciones no lineales.
- **WFG2:** No convexa en algunas regiones, no separable, con regiones insensibles.
- **WFG3:** Similar a WFG2, frente de Pareto degenerado.
- **WFG4:** Completamente multimodal, muchos óptimos locales.
- **WFG5:** Multimodal, patrón de búsqueda engañoso por discontinuidad.
- **WFG6:** No separable, con dependencia entre variables.
- **WFG7:** Separable, con fuerte sesgo.
- **WFG8:** No separable, fuerte sesgo y dependencia entre variables.
- **WFG9:** No separable, óptimos locales, frente de Pareto discontinuo.

Las funciones WFG están disponibles en varias bibliotecas y paquetes de software de optimización como lo es Pymoo [3], y su implementación estándar se puede encontrar en diversos lenguajes de programación utilizados en la investigación de optimización.

2.6. Indicadores de desempeño

La evaluación del desempeño de algoritmos de optimización multi-objetivo juega un papel crítico en el desarrollo y la mejora de métodos capaces de encontrar soluciones eficientes y diversificadas para problemas complejos. Una parte integral de este proceso de evaluación involucra el uso de indicadores de desempeño que proporcionan medidas cuantitativas de la calidad de las soluciones generadas. Estos indicadores no solo ayudan a entender el comportamiento de los algoritmos bajo diferentes escenarios, sino que también facilitan la comparación objetiva entre distintas técnicas de optimización. En esta sección, profundizaremos en dos indicadores de desempeño fundamentales en el campo de la optimización multi-objetivo: el hipervolumen y el Indicador de Generación de Diversidad Invertida (IGD), junto con su variante mejorada, el IGD+. Cada uno de estos indicadores ofrece perspectivas únicas sobre la efectividad de los algoritmos de optimización, considerando aspectos cruciales como la convergencia hacia el frente de Pareto y la distribución de las soluciones en el espacio de objetivos. A través de una exploración detallada de estos indicadores, se revelará cómo pueden ser aplicados para evaluar y guiar el desarrollo de estrategias de optimización más efectivas y eficientes.

2.6.1. Hipervolumen

En la investigación y aplicación de la optimización multi-objetivo, la capacidad para cuantificar y comparar la efectividad de distintas soluciones es fundamental. Una herramienta destacada en este ámbito es el indicador de hipervolumen (HV) [20], que mide el volumen en el espacio de objetivos dominado por un conjunto de soluciones respecto a un punto de referencia específico. Este indicador no solo ofrece una métrica para la calidad y diversidad de las soluciones encontradas sino también facilita la comparación objetiva entre diferentes conjuntos de soluciones, proporcionando una base sólida para la evaluación de algoritmos de optimización multi-objetivo.

El proceso para calcular el hipervolumen implica varios pasos metodológicos. A continuación, se detallan estos pasos:

1. Definición del Conjunto de Soluciones y el Punto de Referencia:

- Identifique el conjunto de soluciones S cuyo hipervolumen desea calcular. Este conjunto suele ser el frente de Pareto obtenido de un proceso de optimización multi-objetivo.
- Elija un punto de referencia y_{ref} , que debe ser dominado por todas las soluciones. Este punto suele establecerse basándose en el punto de nadir ajustado o un punto ligeramente más allá de las peores soluciones conocidas.

$$HV(S, y_{\text{ref}}) := \mathbf{V} \left(\bigcup_{y \in S} \{y' \mid y \leq y_{\text{ref}}\} \right), \quad S \subseteq FP. \quad (2.20)$$

Donde FP representa el conjunto de soluciones en el Frente de Pareto.

2. **Normalización de los Datos (Opcional):** Para facilitar el cálculo y permitir la comparación entre diferentes conjuntos de soluciones, se normalizan los valores de los objetivos utilizando los puntos de nadir e ideal como referencia. La normalización se realiza de la siguiente manera:

$$S' = \frac{S - \text{Nadir}}{\text{Ideal} - \text{Nadir}}. \quad (2.21)$$

Donde:

- **Punto Ideal:** Vector que contiene los mejores valores alcanzables de forma independiente para cada función objetivo:

$$z^{\text{ideal}} = (z_1^{\text{ideal}}, z_2^{\text{ideal}}, \dots, z_k^{\text{ideal}}).$$

Donde cada $z_i^{\text{ideal}} = \min_{x \in F} f_i(x)$, y F denota el conjunto de soluciones no dominadas.

- **Punto Nadir:** Vector que agrupa los peores valores de cada objetivo obtenidos por una solución dentro del conjunto Pareto-óptimo:

$$z^{\text{nadir}} = (z_1^{\text{nadir}}, z_2^{\text{nadir}}, \dots, z_k^{\text{nadir}}).$$

Donde cada $z_i^{\text{nadir}} = \max_{x \in F} f_i(x)$, y F denota el conjunto de soluciones no dominadas.

Este proceso asegura que el espacio de objetivos se normalice con respecto a los rangos definidos por estos puntos críticos como se puede observar en la Figura 2.3.

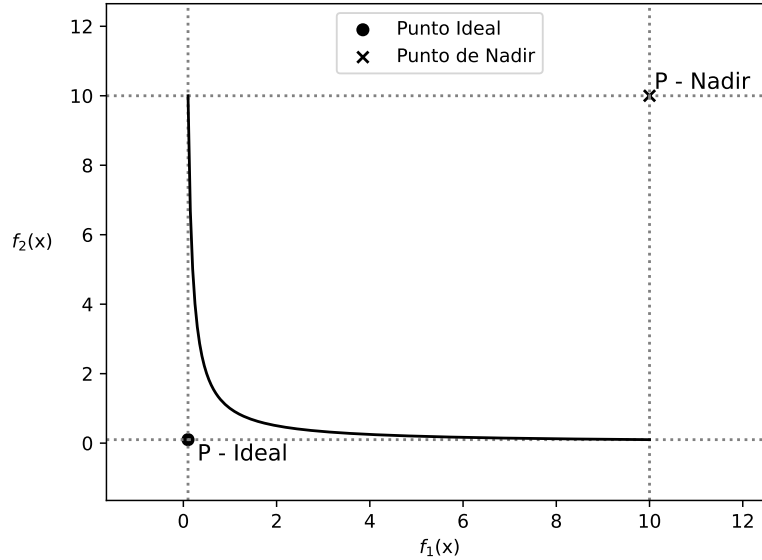


Figura 2.3: Ilustración de la definición del campo de normalización usando el punto Ideal y de Nadir

3. **Cálculo del Hipervolumen:** El hipervolumen se calcula como el volumen del espacio cubierto por el conjunto de soluciones y limitado por el punto de referencia [50]. Para un conjunto de soluciones en un espacio de 2 o 3 objetivos, este cálculo puede ser directo. Para más de 3 objetivos, se utilizan algoritmos especializados:

- Para 2 objetivos, el hipervolumen HV puede calcularse sumando el área de los rectángulos formados por cada solución $s_i \in S'$ y el punto de referencia R' :

$$HV(S') = \sum_{i=1}^{n-1} (x_{i+1} - x_i) \cdot (y_{\text{ref}} - y_i). \quad (2.22)$$

Donde n es el número de soluciones, ordenadas por su valor en el primer objetivo x , y y_{ref} es la coordenada y del punto de referencia R' .

- Para más de 3 objetivos, el cálculo se vuelve computacionalmente más complejo y se recomienda utilizar algoritmos para descomponer el problema y calcular el hipervolumen de manera eficiente como el propuesto en [16].
4. **Interpretación:** El valor del hipervolumen indica el tamaño del espacio dominado por el conjunto de soluciones evaluado. Un mayor hipervolumen indica un mejor conjunto de soluciones en términos de acercamiento al frente de Pareto ideal y diversidad entre las soluciones.

2.6.2. IGD/+

El indicador de Generación de Diversidad Invertida (IGD) [51] se basa en la distancia euclidiana para medir cuán cerca y uniformemente distribuidas están las soluciones generadas por un algoritmo respecto al frente de Pareto óptimo. Por otro lado, el IGD+ mejora este enfoque al considerar distancias en el espacio objetivo que reflejan mejor la dirección de mejora hacia el frente de Pareto, proporcionando así una medida más precisa de la calidad de las soluciones en términos de convergencia y diversidad.

Para calcular el indicador IGD, se sigue el procedimiento detallado a continuación:

1. **Conjunto de Referencia:** Seleccionar o generar un conjunto de referencia P que represente adecuadamente el frente de Pareto óptimo. Este conjunto debe estar compuesto por puntos distribuidos de manera uniforme a lo largo del frente de Pareto.
2. **Evaluación del Conjunto de Soluciones:** Tomar el conjunto de soluciones S generadas por el algoritmo de optimización bajo evaluación.
3. **Cálculo de Distancias:** Para cada punto $p \in P$, encontrar el punto más cercano $s \in S$ y calcular la distancia euclidiana entre ellos. Esto se puede expresar como:

$$d(p, s) = \sqrt{\sum_{i=1}^n (p_i - s_i)^2}. \quad (2.23)$$

Donde n es la dimensión del espacio objetivo.

4. **Promedio de Distancias:** Calcular el promedio de todas las distancias mínimas encontradas para obtener el IGD. La fórmula para el IGD es:

$$\text{IGD}(P, S) = \frac{1}{|P|} \sum_{p \in P} \min_{s \in S} d(p, s). \quad (2.24)$$

Donde $|P|$ indica el número total de puntos en el conjunto de referencia.

Mientras que el procedimiento para calcular el IGD+ [29] es similar al del IGD, con algunas modificaciones en el cálculo de las distancias:

1. **Conjunto de Referencia:** Igual que para el IGD, seleccionar o generar un conjunto de referencia P que represente el frente de Pareto óptimo.
2. **Cálculo de Distancias Modificadas:** Para cada punto $p \in P$, identificar el punto en el conjunto de soluciones S que minimiza una distancia modificada, la cual considera la dirección de mejora hacia el frente de Pareto. La distancia modificada se calcula como:

$$d^+(p, s) = \sqrt{\sum_{i=1}^n (\max\{p_i - s_i, 0\})^2}. \quad (2.25)$$

3. **Promedio de Distancias Modificadas:** El IGD+ se obtiene como el promedio de todas las distancias modificadas, utilizando la fórmula:

$$\text{IGD}^+(P, S) = \frac{1}{|P|} \sum_{p \in P} \min_{s \in S} d^+(p, s). \quad (2.26)$$

2.7. Ciencias de la complejidad

Las ciencias de la complejidad ofrecen un marco conceptual para entender cómo sistemas compuestos por numerosos componentes interconectados pueden exhibir propiedades emergentes y comportamientos no triviales. En la optimización, propiedades emergentes como la diversidad y la adaptación son fundamentales y se alinean estrechamente con los principios de los sistemas complejos.

2.7.1. Diversidad y adaptación en sistemas complejos

El elitismo en los algoritmos evolutivos, que se enfoca en mantener una variedad de soluciones no dominadas en el frente de Pareto, refleja la diversidad emergente observada en sistemas complejos. Esta diversidad es clave para una exploración efectiva del espacio de búsqueda, mientras que la convergencia hacia soluciones de alta calidad ilustra la formación de patrones y estructuras en sistemas complejos adaptativos.

2.7.2. Técnicas de adaptación y auto-organización

Las técnicas de adaptación y auto-organización, fundamentales en las ciencias de la complejidad, son igualmente pertinentes en la optimización multi-objetivo. Los algoritmos que se adaptan a cambios en el paisaje de objetivos y restricciones abordan de manera efectiva el desafío de equilibrar exploración y explotación. Estas técnicas imitan la capacidad de los sistemas complejos para adaptarse y reorganizarse en respuesta a cambios en su entorno [26].

2.7.3. Heterogeneidad temporal

La heterogeneidad en sistemas complejos puede extender la criticidad, un estado que equilibra el orden y el caos, y es crucial para la complejidad, la vida y la computación. Al analizar modelos clásicos homogéneos y sus versiones heterogéneas, se demuestra que la heterogeneidad puede ampliar la criticidad sin necesidad de un ajuste fino de parámetros. Esto sugiere que la selección natural podría explotar la heterogeneidad para evolucionar complejidad de manera más económica y que este principio se podría aplicar en sistemas artificiales y diseño biológico.

La heterogeneidad temporal añade una dimensión adicional a la optimización multi-objetivo, proporcionando robustez y adaptabilidad. En este contexto, la adaptación se convierte en un componente esencial, permitiendo que los individuos más importantes de la población cambien a un ritmo más lento que los menos importantes, generando así diversidad. Esta diversidad fomenta una exploración efectiva del espacio de búsqueda y facilita la identificación rápida de soluciones de alta calidad con una eficiencia computacional optimizada [48].

Además, investigaciones como la de Martínez Arévalo Yoshio [41] abordan la aplicación de la heterogeneidad temporal en algoritmos genéticos para problemas de optimización. Dicho trabajo propone variar las probabilidades de cruce en función de la aptitud de los individuos y aplica este enfoque a problemas como el de las N-reinas y el del vendedor viajero. La investigación revela que la heterogeneidad temporal mejora la eficiencia y precisión de estos algoritmos, ofreciendo una estrategia efectiva para optimizar su rendimiento en problemas complejos. Este enfoque representa un avance significativo en la personalización y adaptabilidad de los algoritmos genéticos para resolver desafíos complejos en optimización.

2.8. Modelos de estudio

Dentro del ámbito de la investigación científica, la estadística juega un papel crucial al permitir el análisis y la interpretación de datos empíricos. Entre los modelos estadísticos frecuentemente empleados en estudios científicos, destacan el Test de Wilcoxon Rank Sum, el Test de Kruskal-Wallis y el Test de los Conteos de Borda. Estos métodos estadísticos son fundamentales para evaluar diferencias significativas entre grupos o

poblaciones de datos. Dichas diferencias se investigan a través de dos hipótesis fundamentales:

- **Hipótesis Nula (H0):** Postula que las medianas de todos los grupos son iguales. Esta hipótesis asume que no hay diferencias significativas entre los grupos.
- **Hipótesis Alternativa (H1):** Sostiene que al menos una de las medianas difiere de las demás. Esta hipótesis se adopta cuando los datos sugieren variaciones estadísticamente significativas entre los grupos.

La elección de estos modelos estadísticos depende de la naturaleza de los datos y del objetivo del análisis. Cada uno de estos *tests* tiene particularidades que los hacen más adecuados para ciertos tipos de datos y situaciones de estudio. Por ejemplo, el Test de Wilcoxon es útil para comparar dos grupos independientes, mientras que el Test de Kruskal-Wallis se aplica para comparar más de dos grupos (estas son pruebas no paramétricas dado que no se garantiza que los evolutivos cumplan las condiciones de normalidad). El Test de los Conteos de Borda se utiliza para rankear múltiples alternativas con base a las preferencias. La correcta aplicación de estos *tests* garantiza una interpretación rigurosa y fiable de los resultados empíricos obtenidos en la investigación.

2.8.1. Kruskal-Wallis

El *test* de Kruskal-Wallis [42] es un método no paramétrico, se utiliza para determinar diferencias estadísticas significativas entre dos o más grupos de una variable independiente, ya sea en una escala ordinal o continua. Este *test* es una generalización del *test* de Mann-Whitney para múltiples grupos y resulta particularmente útil cuando los datos no cumplen con la suposición de normalidad, lo que haría inapropiado el uso del ANOVA unidireccional.

El procedimiento del test de Kruskal-Wallis se desarrolla de la siguiente manera:

1. **Rangos:** Se asignan rangos a todas las observaciones, independientemente del grupo al que pertenecen, desde el valor más pequeño al más grande. Los rangos comienzan en 1 para el valor más bajo. En caso de empates, se asigna a cada observación el promedio de los rangos que corresponderían en ausencia de empate.
2. **Suma de Rangos:** Se calcula la suma total de rangos para cada grupo.
3. **Estadístico de Prueba:** El estadístico de prueba H se determina utilizando las sumas de rangos de cada grupo, el tamaño de cada grupo, y el total de observaciones. La fórmula es:

$$H = \frac{12}{N(N+1)} \sum_{i=1}^k \frac{R_i^2}{n_i} - 3(N+1). \quad (2.27)$$

Donde N es el total de observaciones, k es el número de grupos, R_i es la suma de rangos del i -ésimo grupo, y n_i es el número de observaciones en el i -ésimo grupo.

4. **Distribución de Chi-cuadrado:** Bajo la hipótesis nula, y asumiendo un tamaño de muestra adecuado, el estadístico H sigue una distribución chi-cuadrado aproximada con $k - 1$ grados de libertad.
5. **Decisión:** Se compara el valor calculado de H con el valor crítico de la distribución chi-cuadrado para $k - 1$ grados de libertad y un nivel de significancia α (usualmente 0.05). Si H es mayor que el valor crítico, se rechaza la hipótesis nula.

2.8.2. Wilcoxon ranksum

El test de suma de rangos de Wilcoxon, también conocido como *test* de Wilcoxon-Mann-Whitney [31], es un método no paramétrico ampliamente utilizado para comparar dos grupos independientes de muestras. Este test es una alternativa al test t de Student para muestras independientes, aplicable cuando los supuestos de normalidad no se cumplen. Es adecuado tanto para datos ordinales como para datos de intervalo/*ratio* no distribuidos normalmente.

El procedimiento del test de Wilcoxon ranksum se desarrolla de la siguiente manera:

1. **Combinar y Ordenar los Datos:** Se unen las muestras de ambos grupos y se ordenan todas las observaciones del menor al mayor valor.
2. **Asignar Rangos:** Se otorgan rangos a todas las observaciones ordenadas. En caso de empates, se asigna el promedio de los rangos correspondientes a cada observación empatada.
3. **Calcular Sumas de Rangos por Grupo:** Se determinan las sumas de rangos para cada grupo, denotadas como R_1 y R_2 .
4. **Estadístico de Prueba U :** Se calculan los estadísticos de prueba U_1 y U_2 para cada grupo mediante las fórmulas:

$$\begin{aligned} U_1 &= R_1 - \frac{n_1(n_1 + 1)}{2}, \\ U_2 &= R_2 - \frac{n_2(n_2 + 1)}{2}. \end{aligned} \tag{2.28}$$

Donde R_1 y R_2 son las sumas de rangos, y n_1 y n_2 son los tamaños de muestra de los grupos 1 y 2 respectivamente. El estadístico U corresponde al menor entre U_1 y U_2 .

5. **Normalización del Estadístico de Prueba:** Para muestras grandes, U puede normalizarse a una distribución aproximadamente normal con la fórmula:

$$Z = \frac{U - \mu_U}{\sigma_U}. \tag{2.29}$$

Donde $\mu_U = \frac{n_1 n_2}{2}$ es la media y $\sigma_U = \sqrt{\frac{n_1 n_2 (n_1 + n_2 + 1)}{12}}$ es la desviación estándar de U bajo la hipótesis nula.

6. **Decisión:** Se compara el valor de Z con los valores críticos de la distribución normal estándar para un nivel de significancia elegido (comúnmente 0.05). Si el valor absoluto de Z supera el valor crítico, se rechaza la hipótesis nula.

2.8.3. Conteo de Borda

El conteo de Borda [45], un método de votación por orden de preferencia, fue desarrollado por el matemático francés Jean-Charles de Borda en 1770. Este sistema de votación preferencial se utiliza en decisiones de elección única o múltiple, donde los votantes clasifican las opciones según su preferencia personal.

El procedimiento del recuento de Borda se desarrolla de la siguiente manera:

1. **Clasificación de Opciones:** Cada votante ordena todas las opciones de acuerdo con sus preferencias, desde la más preferida hasta la menos preferida.
2. **Asignación de Puntos:** Se asignan puntos a cada opción basados en su posición en cada votación. Si hay n opciones, la opción más preferida recibe n puntos, la segunda $n - 1$ puntos, y así sucesivamente, hasta que la opción menos preferida recibe 1 punto.
3. **Suma de Puntos:** Los puntos asignados a cada opción se suman a través de todas las votaciones.
4. **Determinación de Ganadores:** La opción con la mayor cantidad total de puntos acumulados se considera la ganadora.

El método de Borda es valorado por su simplicidad y por su capacidad de reflejar de manera más integral el espectro completo de preferencias de los votantes. A pesar de ciertas limitaciones, se destaca por proporcionar una visión más holística de las preferencias en comparación con otros métodos que solo consideran la opción más preferida de cada votante.

2.9. Explicabilidad de modelos de aprendizaje

La explicabilidad en los modelos de aprendizaje representa un aspecto crucial que ha ido ganando relevancia en el campo de la inteligencia artificial y la optimización. La creciente complejidad de los problemas abordados y la proliferación de algoritmos cada vez más sofisticados han impulsado la necesidad de desarrollar y emplear herramientas que permitan una mejor comprensión y selección de algoritmos adecuados. Entre estas herramientas, dos de ellas destacan por su relevancia y aplicabilidad: Shapley Additive Explanations (SHAP) [56, 9] y Python Feature-Based Landscape Analysis of Continuous and Constrained Optimization Problems (P-Flacco) [34, 55].

Shapley Additive Explanations (SHAP) es una técnica que se basa en la teoría de juegos para explicar la salida de cualquier modelo de *machine learning*. SHAP asigna a cada

característica un valor que indica su importancia en la decisión tomada por el modelo, permitiendo así una interpretación más detallada y justificada de sus resultados.

Por otro lado, Python Feature-Based Landscape Analysis of Continuous and Constrained Optimization Problems (P-Flacco) es una herramienta que proporciona un análisis detallado del espacio de búsqueda de problemas de optimización. P-Flacco permite identificar características clave del paisaje de optimización, lo que ayuda en la selección y adaptación de estrategias de búsqueda más eficientes.

La integración de estas herramientas en la evaluación y selección de algoritmos evolutivos multi-objetivo no solo mejora la comprensión de los espacios de búsqueda complejos, sino que también contribuye a la eficacia y eficiencia de los procesos de optimización.

2.9.1. SHAP

SHAP (SHapley Additive exPlanations) [38] es una biblioteca de Python diseñada para interpretar las predicciones de modelos de *machine learning*. Basada en la teoría de juegos, SHAP utiliza los valores de Shapley para asignar a cada característica su importancia en la predicción de un modelo de manera justa y coherente.

Dentro del marco de SHAP, cada característica en un modelo de *machine learning* es considerada como un jugador en el juego de hacer una predicción. Esto se debe a la naturaleza aditiva de las explicaciones que SHAP proporciona, descomponiendo la predicción en una suma de efectos atribuibles a cada característica individual. Esto permite una comprensión detallada de cómo cada característica influye en el resultado final de una predicción.

$$g(z') = \phi_0 + \sum_{i=1}^M \phi_i z'_i. \quad (2.30)$$

Donde $z' \in \{0, 1\}^M$ representa una instancia simplificada de entrada, M es el número de características de entrada, y ϕ_i son los valores SHAP para cada característica.

Las principales funciones de SHAP incluyen:

- **Explainers:** Modelos en SHAP que pueden explicar las predicciones de diversos algoritmos de *machine learning*. Existen diferentes explicadores adecuados para varios tipos de modelos, como árboles de decisión, modelos lineales y redes neuronales.
- **SHAP Values:** Para una predicción específica, SHAP calcula valores que determinan la importancia de cada característica. Un valor SHAP alto, ya sea positivo o negativo, indica una influencia significativa en la predicción.
- **Visualizaciones:** SHAP ofrece varias visualizaciones para interpretar los valores SHAP, incluyendo:

- Force Plots: Muestran la contribución de cada característica a una predicción específica.
- Summary Plots: Ofrecen una visión general de la importancia de las características y su impacto en el modelo.
- Dependence Plots: Ilustran cómo el efecto de una característica varía con respecto a otra.

Los modelos basados en árboles de SHAP son particularmente apreciados en el *machine learning* debido a su capacidad para aprender relaciones complejas en los datos. Son especialmente efectivos en el análisis de conjuntos de datos tabulares y se prefieren en muchas situaciones por su rendimiento, eficiencia computacional y facilidad de uso.

SHAP se ha establecido como una herramienta indispensable en el ámbito de la interpretación de modelos de *machine learning*, proporcionando explicaciones matemáticamente justificadas y accesibles para las predicciones de una amplia variedad de modelos.

2.9.2. P-Flacco

P-Flacco es un paquete de Python desarrollado para el análisis de paisajes de características en problemas de optimización continua y restringida. Esta herramienta representa una implementación en Python del paquete R Flacco, originalmente desarrollado por Pascal Kerschke [33]. Ofreciendo una interfaz de Python para Flacco, P-Flacco integra características adicionales y mejoras. Entre sus características más destacadas se encuentran:

- **Análisis de paisajes de características:** Utilizado para caracterizar numéricamente problemas de optimización de objetivo único, tanto en contextos continuos como restringidos, mediante el análisis detallado de paisajes de características.
- **Soporte para diversos tipos de problemas de optimización:** P-Flacco es capaz de analizar tanto problemas de optimización sin restricciones como restringidos, lo que lo hace versátil para una variedad de aplicaciones prácticas.
- **Eficiencia en la extracción de características:** Ofrece un método eficiente para extraer características relevantes del paisaje de optimización, facilitando la comprensión y el análisis de problemas complejos de optimización.
- **Selección de optimizadores:** Ayuda en la elección del optimizador más adecuado de un conjunto de algoritmos, siendo particularmente útil cuando las funciones objetivo son desconocidas o de naturaleza caja negra.
- **Análisis basado en robustez, embudos y gradientes:** Realiza un análisis sobre la robustez que busca medir la estabilidad de las soluciones frente a pequeñas perturbaciones en el espacio de búsqueda, los embudos que es la identificación de regiones en el espacio de búsqueda donde las soluciones tienden a converger hacia

un único óptimo global o local y los gradientes que examinan las pendientes o inclinaciones del espacio de búsqueda, evaluando su impacto en el rendimiento de algoritmos como la Optimización de Enjambres de Partículas (PSO).

P-Flacco se ha consolidado como una herramienta esencial para profesionales y académicos en el ámbito de la optimización, ofreciendo un enfoque basado en Python para el análisis exhaustivo y detallado del paisaje de optimización.

2.10. Ajuste de parámetros

La optimización de parámetros, también conocida como ajuste de hiperparámetros, es un componente esencial en el aprendizaje automático para maximizar el rendimiento de los modelos. Este proceso es especialmente relevante en la investigación y desarrollo de algoritmos evolutivos multi-objetivo.

En el contexto del aprendizaje automático aplicado a algoritmos evolutivos, es fundamental comprender cómo diversas técnicas pueden influir en la adaptabilidad y eficacia de estos algoritmos a lo largo del tiempo. Una técnica que ha cobrado notable popularidad es Random Forest Regressor, que se basa en una extensión del método de árboles de decisión.

2.10.1. Random Forest Regression

El Random Forest es un método de ensamble que utiliza la combinación de múltiples árboles de decisión para lograr predicciones más precisas y estables que las que se obtendrían de un único árbol de decisión. Este enfoque reduce notablemente el riesgo de sobreajuste, un problema común en los árboles de decisión individuales. Específicamente, el Random Forest Regressor se aplica a problemas de regresión, donde el objetivo es predecir un valor continuo, a diferencia de los problemas de clasificación, que buscan predecir etiquetas de clases discretas.

Las principales ventajas del Random Forest Regressor incluyen:

- **Robustez:** Su uso de múltiples árboles lo hace menos susceptible al sobreajuste en comparación con un único árbol de decisión.
- **Manejo de Datos No Lineales:** Capacidad para capturar relaciones no lineales entre las características y la variable objetivo.
- **Flexibilidad:** Eficacia en una amplia variedad de tipos de datos, funcionando bien en problemas con múltiples características y estructuras de datos complejas.
- **Importancia de Características:** Ofrece una evaluación útil sobre la importancia de las características, facilitando la interpretación del modelo.

En la biblioteca Scikit-learn [46], el Random Forest Regressor se implementa mediante la clase `RandomForestRegressor`. Esta implementación permite una fácil configuración y ajuste del modelo, ofreciendo parámetros como el número de árboles (`n_estimators`), la profundidad máxima de los árboles (`max_depth`), entre otros. Estos parámetros pueden ser óptimamente ajustados utilizando técnicas como GridSearchCV, que realiza una búsqueda exhaustiva sobre un *grid* de valores de hiperparámetros especificados. Este proceso evalúa y compara el rendimiento del modelo para cada combinación de hiperparámetros, facilitando la identificación de la configuración más efectiva.

El uso de técnicas avanzadas de aprendizaje automático, como el Random Forest, en el ajuste de hiperparámetros es un área de prometedora que ofrece oportunidades significativas para la mejora continua de los algoritmos, especialmente en contextos dinámicos y cambiantes.

2.11. Estado del arte

Dentro del estado del arte, encontramos múltiples trabajos relevantes para la optimización multi-objetivo evolutiva. Uno de los estudios destacados es el de Guiseppe [8]. Esta investigación introduce una técnica innovadora para superar la convergencia en óptimos locales mediante reinicios. El enfoque propuesto se centra en reiniciar la búsqueda en áreas menos exploradas del espacio de búsqueda, seleccionando individuos según su novedad. Este método ha demostrado ser eficaz en problemas de prueba multi-modales, resaltando su utilidad en la búsqueda de soluciones óptimas y la exploración efectiva en contextos complejos. Este estudio aporta una perspectiva valiosa sobre el equilibrio entre exploración y explotación, y cómo la incorporación de la novedad puede enriquecer los métodos de optimización multi-objetivo.

Otro desarrollo significativo es el de Tanabe [52], que se enfoca en el uso de un archivo externo para almacenar soluciones no dominadas. Este estudio compara algoritmos evolutivos multi-objetivo elitistas y no elitistas en términos de selecciones ambientales, evaluando su rendimiento en el conjunto de problemas BiBOB. Los resultados indican que los algoritmos no elitistas, combinados con ciertos métodos de cruce, son efectivos en problemas con muchas variables de decisión, especialmente cuando se utiliza un archivo externo.

Recientemente, un enfoque intrigante en la investigación de algoritmos genéticos ha surgido: la exploración de la heterogeneidad temporal en la asignación de probabilidades de cruce. A diferencia de los valores uniformes tradicionales, se propone asignar probabilidades de cruce heterogéneas, variando según la aptitud de los individuos. El trabajo reciente de Arévalo [40] aplica estos conceptos al problema de las N reinas y al Problema del Viajante. Los resultados preliminares sugieren que la heterogeneidad temporal puede mejorar significativamente la eficiencia y la calidad de las soluciones en algoritmos genéticos, abriendo una nueva dirección de investigación prometedora para la resolución de problemas complejos en ingeniería y disciplinas relacionadas.

Por último, en el campo de la selección y configuración automatizada de algoritmos, la predicción del rendimiento es un aspecto crucial. Los modelos de aprendizaje automático supervisado, a menudo considerados como cajas negras, plantean desafíos en términos de explicabilidad. El trabajo de Trajanov [54], titulado 'Explainable Landscape Analysis in Automated Algorithm Performance Prediction', aborda este desafío. Utiliza el Análisis Exploratorio del Paisaje (ELA) para calcular características del espacio de búsqueda, agrupándolas en diferentes categorías y dividiéndolas en grupos de características baratas y costosas. Su investigación emplea un *pipeline* de *machine learning* para explorar la explicabilidad de las características ELA en la predicción del rendimiento, resaltando la importancia de la explicabilidad en esta área.

Capítulo 3

Metodología

En esta investigación, nos centramos en la heterogeneidad temporal, un fenómeno que se manifiesta cuando diferentes elementos de un sistema experimentan cambios a distintas velocidades. Identificamos que, en nuestro estudio, los componentes críticos varían a una velocidad menor. Este aspecto subraya que la tasa de cambio en el sistema es un parámetro crucial que necesita ser ajustado. En el ámbito de los algoritmos evolutivos, esto implica evaluar en qué medida se conservan las mejores soluciones.

En el presente capítulo, abordaremos en profundidad la metodología adoptada para nuestra investigación.

3.1. Diseño global con parámetro de heterogeneidad temporal

Para abordar la heterogeneidad temporal en algoritmos evolutivos, hemos introducido un parámetro adicional, denotado como H . Este parámetro representa el grado de heterogeneidad en la composición de la población. Específicamente, H define el porcentaje de la población que está compuesto por soluciones elitistas, mientras que el restante $1 - H$ se compone de soluciones aleatorias. Este enfoque presenta similitudes con los algoritmos evolutivos de estado estable, como se describe en [44]. Ambos comparten características fundamentales, como la actualización gradual de la población, el mantenimiento de la diversidad y una adaptabilidad dinámica ante las variaciones del entorno. Sin embargo, una distinción importante radica en que, mientras la heterogeneidad temporal se enfoca en la variabilidad de las condiciones ambientales, los algoritmos de estado estable centran su mecanismo en las estrategias de reemplazo de individuos dentro de la población, lo que subraya diferencias en la forma en que se gestionan las adaptaciones y la evolución.

El algoritmo propuesto, se ilustra en el Algoritmo 1, sigue la estructura de un algoritmo evolutivo tradicional, pero introduce modificaciones significativas en las etapas de selección de individuos para incorporar el parámetro H . En las líneas 6 y 7 del algoritmo, donde se lleva a cabo la selección de individuos óptimos y aleatorios, respectivamente,

de acuerdo con el valor de H .

Utilizando $H = 1$, el algoritmo se comporta de manera idéntica a la implementación clásica de un algoritmo evolutivo, donde solo se seleccionan individuos elitistas. Por otro lado, un valor de $H = 0$ resulta en una selección completamente aleatoria, promoviendo una exploración exhaustiva del espacio de búsqueda sin priorizar a los individuos elitistas. Este enfoque permite ajustar dinámicamente el balance entre explotación y exploración.

La integración del nuevo parámetro H indica el porcentaje de la población compuesto por soluciones elitistas. El restante $1 - H$ se compone de soluciones aleatorias restantes. Al permitir seleccionar la proporción de mejores genes que influyen en la población en cada iteración, buscamos mantener la exploración efectiva sin sacrificar la calidad de la población elitista.

Algorithm 1 Algoritmo evolutivo con heterogeneidad temporal

Require: $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$: función objetivo, $H \in [0, 1]$: heterogeneidad, $npop$: tamaño de la población

Ensure: *population*: población final

```

1: population  $\leftarrow$  initPopulation(npop)
2: fpopulation  $\leftarrow$  evaluatePopulation(f, population)
3: for  $i = 1 \dots \text{maxIter}$  do
4:   parents  $\leftarrow$  selectParents(population, fpopulation)
5:   children  $\leftarrow$  geneticOperators(parents)
6:   best  $\leftarrow$  selectBest(population  $\cup$  children,  $\text{ceil}(H * npop)$ )
7:   random  $\leftarrow$  selectRandom(population  $\cup$  children,  $\text{floor}((1 - H) * npop)$ )
8:   population  $\leftarrow$  best  $\cup$  random
9: end for
```

3.2. Marco experimental

Los experimentos diseñados para esta investigación se llevaron a cabo en contextos tanto de problemas de objetivo único como de problemas de objetivo múltiple, con el fin de evaluar la eficacia de los algoritmos evolutivos bajo diferentes escenarios. A continuación se explicará el marco experimental para cada caso.

3.2.1. Experimentos mono-objetivo

Los experimentos en el contexto de problemas mono-objetivo involucraron el uso de Algoritmos Genéticos (GA), Evolución Diferencial (DE) y Estrategias Evolutivas (ES), aplicados al conjunto de pruebas BBOB, que incluye veinticuatro funciones distintas. Se realizaron pruebas en varias dimensiones: 2, 3, 5, 10, 20 y 40, siendo cada combinación de dichos parámetros una instancia de pruebas. Para cada instancia, se llevó a cabo un

total de veinte experimentos, lo que resulta en cincuenta mil cuatrocientas ejecuciones. Cada instancia se ejecutó por mil generaciones, y se realizaron muestreos en cada generación. Esto nos llevó a acumular un total de cincuenta millones cuatrocientos mil resultados, que fueron registrados para su posterior análisis.

Implementación del conjunto de pruebas

Los experimentos se llevaron a cabo utilizando las 24 funciones sin ruido del conjunto BBOB, cada una con características únicas que las hacen adecuadas para probar distintos aspectos de los algoritmos de optimización. A pesar de compartir ciertas similitudes, estas funciones presentan una variedad de características y limitaciones que influyen en su comportamiento en el contexto de optimización. Estas características y limitaciones específicas de las funciones del conjunto BBOB fueron consideradas en la implementación de los experimentos para evaluar el efecto de la heterogeneidad en los algoritmos de optimización mono-objetivo bajo diferentes condiciones y desafíos.

Implementación de algoritmos

Implementamos los algoritmos evolutivos con un enfoque en la heterogeneidad temporal, adaptando los parámetros de cada algoritmo para integrar este concepto. A continuación, se describen los ajustes específicos realizados en cada algoritmo:

Algoritmo genético (GA):

- **Codificación Real:** Representación de individuos mediante vectores de números reales, adecuados para problemas de optimización continua [19].
- **Cruza BLX- α :** Método de cruce que genera hijos con genes extendidos más allá de los valores parentales [15].
- **Mutación Uniforme:** Mutación aleatoria de genes individuales con probabilidad r_mut [19].
- **Selección Greedy:** Elección de los mejores individuos basada en rendimiento, integrando un componente aleatorio según el parámetro de heterogeneidad como se propone en el Algoritmo 1.

Evolución diferencial (DE):

- **Codificación Real:** Uso de vectores de números reales para representar las soluciones [13].
- **Mutación Diferencial:** Generación de vectores mutados combinando tres individuos distintos [13].
- **Cruce Uniforme:** Mezcla de vectores mutados y objetivos para formar nuevos individuos [13].

- **Selección Greedy:** Elección de los mejores individuos basada en rendimiento, integrando un componente aleatorio según el parámetro de heterogeneidad como se propone en el Algoritmo 1.

Estrategia evolutiva (ES):

- **Codificación Real:** Uso de vectores de números reales para representar las soluciones [49].
- **Mutación Gaussiana:** Aplicación de ruido gaussiano a los genes para generar hijos [49].
- **Selección Greedy:** Elección de los mejores individuos basada en rendimiento, integrando un componente aleatorio según el parámetro de heterogeneidad como se propone en el Algoritmo 1.

Configuración de parámetros

Cada algoritmo se configuró con parámetros específicos para adecuarse al contexto de optimización mono-objetivo y para incorporar la heterogeneidad temporal y se configuraron de la siguiente forma:

Parámetros comunes en cada algoritmo

- **n_iter: 1,000** Número de iteraciones del algoritmo. Es el número de veces que el algoritmo se ejecutará en su ciclo principal, intentando mejorar la solución actual.
- **n_pop: 200** Tamaño de la población. Es el número de soluciones individuales que se mantendrán en la población en cada iteración del algoritmo.
- **r_mut: 0.1** Tasa de mutación. Es un parámetro que influye en la operación de mutación del algoritmo, controlando cuánto cambian las soluciones durante la mutación.
- **r_cross: 0.8** Tasa de cruce (*crossover*). Determina la probabilidad de que cada elemento de una solución se cambie por el correspondiente en la solución mutada durante la operación de cruce.
- **heterogeneity** Especifica el porcentaje de la población que se seleccionará de las mejores soluciones actuales frente a una selección aleatoria.
- **alpha: 0.25** Factor utilizado en la operación de cruce.

Estrategia evolutiva:

- **step_size: 0.15** Tamaño del paso utilizado en la generación de hijos.

Indicadores de desempeño y ejecuciones

Los resultados se registran como el valor óptimo encontrado por la población hasta la generación actual, actualizándose solo cuando un individuo encuentra una mejor solución. Configuramos los algoritmos, problemas y dimensiones para evaluar el efecto de la heterogeneidad. A partir de veinte semillas por instancia, se realizaron múltiples ejecuciones, almacenando los resultados en un *DataFrame* de Pandas. Cada algoritmo se inicializó con un valor específico de heterogeneidad y se ejecutó desde la generación cero, pasando por procesos de evaluación, selección, cruce y mutación.

3.2.2. Experimentos multi-objetivo

En el ámbito de los experimentos multi-objetivo, optamos por emplear los algoritmos NSGA-II y NSGA-III para evaluar su rendimiento en problemas de optimización con múltiples objetivos. Estos experimentos involucraron el uso de un archivo externo, el cual contiene la información de todos los individuos de la población por cada generación que se probaron en los dos conjuntos de pruebas bien establecidos: Bi-BBOB y WFG.

Con estos experimentos, se buscó obtener una comprensión integral del impacto de la heterogeneidad temporal en distintos escenarios de optimización, evaluando su eficacia en los algoritmos NSGA-II y NSGA-III en una amplia gama de problemas y configuraciones.

Implementación del conjunto de pruebas

En nuestro enfoque experimental multi-objetivo, abordamos una diversidad de funciones y configuraciones, con un enfoque particular en los efectos en múltiples objetivos, variando desde dos hasta diez objetivos. Profundicemos en las características de los conjuntos de pruebas BiBBOB y WFG.

BiBBOB Para el contexto bi-objetivo, se utilizaron las cincuenta y cinco funciones del conjunto Bi-BBOB. Mediante el paquete IOHexperimenter [14], accedimos a las funciones BBOB y las adaptamos para ser compatibles con Pymoo, siguiendo la estructura e instancia mencionada en [5]. Las pruebas se realizaron manteniendo las mismas dimensiones que en los experimentos mono-objetivo. En total, se efectuaron veinte experimentos por instancia, similar a los experimentos mono-objetivo, lo que llevó a un total de sesenta y seis mil instancias. En cada generación de estas instancias, se realizó un muestreo, acumulando sesenta y seis millones de resultados que se registraron en una tabla final.

WFG Para más de dos objetivos, nos centramos en las nueve funciones de WFG, las cuales fueron provistas por el paquete Pymoo, modificando el parámetro de dimensiones por el total de objetivos, con valores de $k = 3, 5, 7$ y 10 . Al igual que con BiBBOB, se evaluó cada valor de heterogeneidad realizando veinte experimentos por instancia.

Esto resultó en un total de siete mil doscientas instancias, y al igual que en los otros experimentos, se tomaron muestras en cada generación, sumando un total de siete millones doscientos mil resultados, que también se registraron para su análisis posterior.

Los parámetros utilizados para estas funciones son:

- **Dimensiones:** Se trabajó con 24 variables de decisión.
- **Límites de la Función:** Cada dimensión i tiene límites definidos en el intervalo $[0, 2i]$.

Estas configuraciones de los conjuntos de pruebas BiBBOB y WFG permitieron evaluar de manera exhaustiva la efectividad de los algoritmos NSGA-II y NSGA-III en un amplio espectro de escenarios de optimización multi-objetivo.

Implementación de algoritmos

Siguiendo un enfoque similar al utilizado en los experimentos mono-objetivo, modificamos los algoritmos NSGA-II y NSGA-III del paquete Pymoo para incorporar la heterogeneidad temporal. Este ajuste influye significativamente en el mecanismo de selección de ambos algoritmos.

Modificaciones en NSGA-II y NSGA-III: En NSGA-II, el nuevo parámetro de heterogeneidad interviene después de calcular los frentes de Pareto y la *crowding distance*. La población se divide en dos grupos: los elitistas, seleccionados en su totalidad como supervivientes por el algoritmo, y los pluralistas, generalmente descartados por este. La heterogeneidad determina la proporción de individuos seleccionados de ambos grupos, equilibrando entre los mejores de los elitistas y una selección aleatoria de la población restante.

En NSGA-III, esta lógica se adapta al reemplazar la distancia de hacinamiento con el cálculo relativo a los puntos de referencia. La heterogeneidad modula la selección de individuos en relación con estos puntos de referencia, manteniendo la diversidad en el frente de Pareto.

Parámetros comunes en cada algoritmo:

- **n_iter: 1,000** - Define el número de iteraciones del algoritmo, determinando la duración del proceso de optimización.
- **n_pop: 200** - Establece el tamaño de la población, es decir, el número de soluciones individuales en cada iteración.
- **heterogeneity** - Controla el porcentaje de la población seleccionada como las mejores soluciones frente a una selección aleatoria.

Configuración específica de NSGA-III:

- **ref_dirs:** (**metodo** = Riesz s-Energy [4], **muestras** = **n_pop**) - Las direcciones de referencia son clave en NSGA-III para mantener la diversidad en el frente de Pareto.

Estas adaptaciones permiten evaluar de manera efectiva el impacto de la heterogeneidad temporal en el rendimiento de los algoritmos en escenarios de optimización multi-objetivo, proporcionando una visión más clara de su comportamiento en condiciones variables.

Datos y ejecuciones

Para una evaluación exhaustiva del rendimiento de los algoritmos de optimización multi-objetivo, se emplearon dos indicadores clave: IGD+ [29] y HV [20]. Estas métricas son fundamentales para entender la efectividad de los algoritmos, particularmente en términos de calidad y diversidad de las soluciones generadas.

- **Inverted Generational Distance Plus (IGD+)** El cálculo de IGD+ en cada generación permite monitorear la evolución y aproximación de las soluciones al óptimo teórico aplicado a cada ($n_objectives$) propuestos en el problema.
- **Hypervolume (HV)** Debido a consideraciones computacionales, el cálculo del HV se realiza solo cuando el número de objetivos ($n_objectives$) es cinco o menos, ya que su complejidad aumenta significativamente con el número de objetivos.

En el proceso de recopilación de datos, almacenamos tres tablas de información. La primera tabla contiene información sobre toda la población por generación, que se guarda como un archivo externo y será útil en la fase de análisis. La segunda tabla almacena el valor del indicador obtenido por el Frente de Pareto. Por último, se registran los valores en el espacio de los objetivos de cada individuo perteneciente al Frente de Pareto. Todos estos datos se capturan en diferentes *DataFrames*.

Esta metodología de registro asegura que dispongamos de un conjunto de datos completo para el análisis posterior del rendimiento de los algoritmos bajo diferentes configuraciones y condiciones.

3.3. Análisis

Esta sección expone la metodología empleada para analizar los datos obtenidos de los experimentos realizados, con el objetivo de explorar el impacto de la heterogeneidad temporal en algoritmos evolutivos multi-objetivo. El análisis se orienta hacia la resolución de las siguientes interrogantes fundamentales:

- ¿Existe algún impacto estadísticamente significativo al variar los valores de heterogeneidad temporal?

- En caso de identificar diferencias estadísticas, ¿cuál sería el valor óptimo de heterogeneidad para maximizar la eficacia en la resolución de un problema específico de optimización?

Para abordar estas preguntas, se seleccionaron técnicas de estadística no paramétricas, dado que no se asume una distribución específica de los datos. A continuación se describe el procedimiento utilizado.

3.3.1. Esquema general

El enfoque experimental se centró en evaluar la influencia de la heterogeneidad temporal en los algoritmos evolutivos. Se partió con dos estrategias evolutivas extremas, representadas por los valores de heterogeneidad de 0 para la estrategia (μ, λ) y 1 para $(\mu + \lambda)$. Luego, se analizaron tres valores intermedios (0.25, 0.50, y 0.75) para investigar de manera más detallada el efecto de la heterogeneidad en los algoritmos, los tipos de problemas y las dimensiones evaluadas.

Los experimentos se ejecutaron en una computadora equipada con un procesador Threadripper de tercera generación modelo 3970X de 32 núcleos a 4 GHz y 32 GB de RAM en configuración de canal cuádruple. Se utilizó un enfoque de multi-procesamiento para realizar las veinte pruebas de forma paralela, como se muestra en el Algoritmo 2.

Algorithm 2 Proceso de Optimización Multi-Objetivo

Require: *algorithm, problem, n_objectives, h, n_test, n_population, maintenance = False*

- 1: *process* \leftarrow Lista vacía de procesos
- 2: *multi* \leftarrow Verdadero
- 3: **for** *id, n* en *enumerate(n_test)* **do**
- 4: **if** *n_objectives* ≥ 10 o *maintenance* **then**
- 5: TEST(*algorithm, problem, n_objectives, h, id, n, n_population*)
- 6: *multi* \leftarrow Falso
- 7: **else**
- 8: *process[id]* \leftarrow Nuevo proceso con *target* TEST y *args*
- 9: START(*process[id]*)
- 10: **end if**
- 11: **end for**
- 12: **if** *multi* **then**
- 13: **for** *proces* en *process* **do**
- 14: JOIN(*proces*)
- 15: **end for**
- 16: **end if**

Además, se utilizaron las siguientes versiones de bibliotecas relevantes para garantizar la reproducibilidad y precisión de los resultados:

- **Python:** 3.10

- **IOHexperimenter:** 0.3.10
- **Pymoo:** 0.6.0
- **Pandas:** 1.4.4
- **P-Flacco:** 1.1.0
- **SHAP:** 0.41.0
- **ScikitLearn:** 1.2.2
- **Matplotlib:** 3.6.3

Esta configuración experimental y de software nos proporciona una base sólida para el análisis de los datos recopilados y la evaluación del impacto de la heterogeneidad en los algoritmos de optimización multi-objetivo.

3.3.2. Base de datos

Para el análisis de los indicadores de desempeño, empleamos un enfoque detallado que involucra el examen de los datos recopilados para cada algoritmo, problema, dimensión y nivel de heterogeneidad.

Mono-objetivo En el contexto mono-objetivo, utilizamos los datos del archivo CSV de nombre Data, cargándolos en un *DataFrame*. Seleccionamos y analizamos los datos correspondientes a cada combinación específica de algoritmo, problema, dimensión y nivel de heterogeneidad. Este proceso resulta en veinte tablas distintas, cada una representando los resultados de una prueba particular.

Para cada conjunto de datos, realizamos un análisis estadístico detallado por generación, calculando la media, mediana y desviación estándar de los resultados obtenidos. Esto nos permite generar una tabla de análisis para cada instancia de problema, y al consolidar todas estas instancias, obtenemos una visión completa de los resultados empíricos del laboratorio mono-objetivo.

Multi-objetivo En el caso multi-objetivo, seguimos un procedimiento similar, utilizando los datos del archivo mencionado anteriormente. Este archivo contiene información detallada sobre el alcance potencial de la población en comparación con el frente de Pareto. Para analizar estos datos, normalizamos la información de los individuos para cada problema y dimensión, ajustándola a una escala de cero a uno.

Con los datos de los individuos normalizados, calculamos el hipervolumen para cada instancia de problema, utilizando la población completa. Esta nueva tabla de datos se somete al mismo análisis estadístico aplicado en el contexto mono-objetivo, generando así una tabla de análisis específica para el bi-objetivo.

Evaluación estadística

Estos análisis se complementan con pruebas estadísticas para evaluar la significancia de los resultados. Aplicamos pruebas como Kruskal-Wallis, Wilcoxon Rank Sum y Borda para examinar la significancia estadística de las diferencias observadas, lo que nos permite obtener una comprensión más profunda de las variaciones en los resultados y su relevancia en el contexto de la optimización multi-objetivo.

3.3.3. Gráficas de convergencia

Para visualizar la evolución de las métricas de rendimiento en nuestros experimentos de optimización, diseñamos una función específica para generar gráficos de convergencia. Estos gráficos representan cómo métricas clave el promedio de las veinte pruebas del valor de HV e IGD+ en Multi-Objetivo y Mejor Resultado en Mono-Objetivo, mostrando como cambian a lo largo de las generaciones para diferentes configuraciones de algoritmos, problemas, y parámetros de objetivos y heterogeneidad.

Procedimiento de generación de gráficos

Inicialmente, configuramos los parámetros de visualización y leemos los datos de rendimiento desde un archivo CSV denominado Data. Este archivo contiene métricas calculadas previamente para varias combinaciones de algoritmos, problemas y niveles de heterogeneidad.

- Realizamos una iteración exhaustiva sobre diversas combinaciones, incluyendo algoritmos, problemas y números de objetivos/dimensiones.
- Para cada combinación, iteramos sobre distintos valores de heterogeneidad (0.25, 0.5, 0.75, 1.0).

Creación y personalización de gráficos

Durante la ejecución, generamos gráficos que muestran la evolución de las métricas seleccionadas a lo largo de las generaciones. Dependiendo del número de objetivos, se crean diferentes tipos de gráficos en el caso de multi-objetivo:

- Para 5 objetivos o menos, generamos dos figuras, una para HV y otra para IGD+.
- Para más de 5 objetivos, generamos una sola figura para IGD+.

Utilizamos subplots de Matplotlib para estos gráficos, personalizándolos con títulos, etiquetas y leyendas. Cada gráfico refleja claramente la combinación de algoritmo, problema y número de objetivos, además de diferenciar las líneas en función de los valores de heterogeneidad.

Almacenamiento y Gestión de Recursos

Tras la creación, los gráficos se guardan en formato PDF. Cerramos cada figura después de guardarla para optimizar la gestión de memoria. Esta metodología nos permite obtener una representación visual clara del rendimiento de los algoritmos bajo diferentes configuraciones, facilitando la interpretación y el análisis de los resultados experimentales.

3.3.4. Modelos

En nuestro análisis, aplicamos modelos estadísticos para abordar tres aspectos clave de los resultados obtenidos en los experimentos de optimización:

Significancia estadística de los datos

Para determinar la significancia estadística de los datos, utilizamos la prueba de Kruskal-Wallis. La cual recibe el banco de datos generados en el archivo CSV denominado Data.

Evaluación de la significancia entre Valores de heterogeneidad

Para analizar la significancia entre distintos niveles de heterogeneidad, empleamos la prueba de Wilcoxon Rank Sum. Esta prueba compara los resultados obtenidos para diferentes valores de heterogeneidad en cada instancia específica de algoritmo, problema y dimensión. En contextos multi-objetivo, donde buscamos maximizar el valor del hipervolumen, la prueba evalúa el impacto de la heterogeneidad en la búsqueda de valores más grandes. En contextos mono-objetivo, donde el objetivo es minimizar los valores de los objetivos, examina el impacto en la búsqueda de valores más pequeños. Los resultados de esta prueba se documentan en una tabla dedicada.

Identificación del Valor de heterogeneidad más efectivo

Utilizando los resultados de las pruebas de Wilcoxon, generamos una tabla de conteo de Borda. En esta tabla, cada valor de heterogeneidad que demuestra un nivel de significancia menor a 0.05 recibe un punto. Este método nos permite identificar la frecuencia con la que cada nivel de heterogeneidad resulta ser el más efectivo para una instancia particular. La tabla se nombra de acuerdo con la prueba.

Estos modelos y análisis se realizaron utilizando el paquete de análisis de datos de Scipy, proporcionando una visión integral y detallada de la significancia y el impacto de la heterogeneidad en nuestros experimentos.

3.3.5. Análisis de los espacios de búsqueda

Tras los análisis iniciales, emprendemos un nuevo estudio centrado en la exploración de los espacios de búsqueda asociados a los problemas de optimización. El objetivo es adquirir una comprensión detallada de las características inherentes de cada problema y cómo estas interactúan con los valores de heterogeneidad y los algoritmos aplicados.

Exploración de características de los problemas

Utilizamos el paquete P-Flacco para generar funciones del conjunto BBOB y llevar a cabo un análisis de Exploratory Landscape Analysis (ELA) en profundidad. Esto nos permite recopilar datos valiosos sobre las peculiaridades de cada problema y entender mejor su naturaleza y complejidad.

Establecimiento de correlaciones

Nuestro enfoque se orienta a descubrir correlaciones significativas entre las características de los problemas y los valores de heterogeneidad. Buscamos predecir los valores óptimos de heterogeneidad para algoritmos específicos basándonos en estas características. Para esto, analizamos los resultados de ELA y generamos mapas de calor, facilitando la visualización de las relaciones entre distintas características del problema.

Aplicación de *machine learning*

Empleamos el modelo `RandomForestRegressor` de ScikitLearn para predecir el valor óptimo de heterogeneidad siendo esta nuestro valor de salida del modelo y utilizando las características obtenidas de los problemas de búsqueda identificadas como entradas. Para optimizar los parámetros del modelo, aplicamos `GridSearchCV`. Este enfoque nos ayuda a determinar la configuración más eficaz del modelo, incluyendo la profundidad óptima del modelo.

Identificación de características clave

Tras la optimización de hiperparámetros, usamos el paquete SHAP para identificar las características que ejercen un mayor impacto en el modelo `RandomForestRegressor` empleado. Esto nos permite comprender mejor qué factores son los más influyentes en la eficacia de la heterogeneidad en los espacios de búsqueda y cómo estos se relacionan con los resultados obtenidos en nuestras pruebas.

Este análisis detallado de los espacios de búsqueda es crucial para comprender la influencia de las características de los problemas en la efectividad de los algoritmos. Además, ofrece una base sólida para decisiones informadas sobre la selección de algoritmos y valores de heterogeneidad en futuras investigaciones y aplicaciones prácticas.

Capítulo 4

Resultados

Este capítulo se dedica a ofrecer un análisis detallado de estos resultados, subrayando la importancia de la heterogeneidad en la selección de algoritmos y en la configuración de problemas para alcanzar resultados óptimos. Examinaremos los datos recopilados desde varias perspectivas, incluyendo el desempeño de los algoritmos en diferentes niveles de heterogeneidad, la influencia de la heterogeneidad en las características de los problemas de optimización y las implicaciones de estos hallazgos para la práctica de la optimización en entornos complejos para contestar las preguntas planteadas en la investigación.

4.1. Mono-objetivo

Esta sección se dedica a desglosar y analizar los resultados obtenidos en problemas mono-objetivo, una faceta esencial de la optimización evolutiva. Se destacarán patrones emergentes y lecciones aprendidas para comprender mejor la influencia de la heterogeneidad en este tipo de problemas.

4.1.1. Análisis empírico

El análisis empírico de los datos recogidos en los experimentos mono-objetivo nos permite corroborar y reforzar las observaciones preliminares sobre la influencia de la heterogeneidad. Los resultados obtenidos y resumidos en la Tabla 4.1 de métricas Mono Objetivo indican una tendencia clara de la heterogeneidad, tanto en problemas de baja como de alta dimensionalidad.

Algoritmo	Problema	Dimensiones	h=0.0	h=0.25	h=0.5	h=0.75	h=1.0
GA	Sphere	2	7.9484E+01(3.7492E-03)	7.9480E+01(1.4254E-04)	7.9480E+01(4.9077E-05)	7.9480E+01(1.7433E-05)	7.9480E+01(7.5652E-06)
GA	Sphere	10	1.0429E+02(7.9521E+00)	7.9510E+01(1.4726E-02)	7.9495E+01(6.9573E-03)	7.9490E+01(5.0258E-03)	7.9489E+01(4.4419E-03)
GA	Sphere	40	3.6321E+02(2.5305E+01)	1.0540E+02(4.5501E+00)	1.0468E+02(2.9068E+00)	1.0354E+02(3.5920E+00)	1.0165E+02(2.1120E+00)
GA	Weierstrass	2	7.1399E+01(5.3488E-02)	7.1353E+01(2.7189E-03)	7.1354E+01(2.8201E-03)	7.1352E+01(1.9857E-03)	7.1351E+01(1.4899E-03)
GA	Weierstrass	10	9.1821E+01(4.7914E+00)	7.7086E+01(2.8347E+00)	7.5554E+01(1.5945E+00)	7.5316E+01(1.6731E+00)	7.5542E+01(1.9201E+00)
GA	Weierstrass	40	1.1414E+02(3.7013E+00)	9.8668E+01(2.6435E+00)	9.6950E+01(2.0189E+00)	9.5475E+01(2.6083E+00)	9.5487E+01(2.5075E+00)
GA	Gallagher21	2	-1.0000E+03(1.4708E-04)	-1.0000E+03(4.9135E-05)	-1.0000E+03(1.4645E-06)	-1.0000E+03(1.3635E-05)	-1.0000E+03(2.5357E-03)
GA	Gallagher21	10	-9.4362E+02(1.1383E+01)	-9.9657E+02(4.2071E+00)	-9.9806E+02(1.4543E+00)	-9.9847E+02(7.1157E-01)	-9.9867E+02(7.2902E-01)
GA	Gallagher21	40	-9.1588E+02(7.4863E-01)	-9.7967E+02(9.6866E+00)	-9.8345E+02(6.3943E+00)	-9.8453E+02(5.7575E+00)	-9.8601E+02(4.3706E+00)
DE	Sphere	2	7.9481E+01(2.2766E-03)	7.9480E+01(0.0000E+00)	7.9480E+01(0.0000E+00)	7.9480E+01(0.0000E+00)	7.9480E+01(0.0000E+00)
DE	Sphere	10	9.3710E+01(4.8424E+00)	7.9480E+01(0.0000E+00)	7.9480E+01(7.4964E-05)	7.9480E+01(3.8316E-05)	7.9480E+01(2.0214E-05)
DE	Sphere	40	2.9783E+02(2.2498E+01)	7.9712E+01(2.7139E-01)	7.9532E+01(8.4440E-02)	7.9534E+01(7.8243E-02)	7.9507E+01(7.8327E-02)
DE	Weierstrass	2	7.1370E+01(2.1977E-02)	7.1350E+01(1.4580E-14)	7.1350E+01(1.4580E-14)	7.1350E+01(1.4580E-14)	7.1350E+01(1.8318E-08)
DE	Weierstrass	10	7.9492E+01(1.4912E+00)	7.1525E+01(1.7680E-01)	7.1467E+01(9.7423E-02)	7.1477E+01(1.5281E-01)	7.1548E+01(2.2504E-01)
DE	Weierstrass	40	1.0094E+02(1.9037E+00)	9.3671E+01(9.5860E+00)	9.2507E+01(9.9780E+00)	8.6584E+01(1.2072E+01)	8.2399E+01(1.2101E+01)
DE	Gallagher21	2	-1.0000E+03(3.1635E-03)	-1.0000E+03(4.6054E-06)	-1.0000E+03(8.0655E-07)	-1.0000E+03(3.8874E-06)	-1.0000E+03(1.9084E-06)
DE	Gallagher21	10	-9.7427E+02(1.0155E+01)	-9.9821E+02(6.3563E-01)	-9.9811E+02(2.8273E-01)	-9.9812E+02(3.2623E-01)	-9.9814E+02(4.2861E-01)
DE	Gallagher21	40	-9.1880E+02(1.7186E+00)	-9.9031E+02(8.2209E+00)	-9.8895E+02(8.3852E+00)	-9.9229E+02(8.0253E+00)	-9.8987E+02(8.3824E+00)
ES	Sphere	2	7.9529E+01(1.0159E-01)	7.9480E+01(1.2267E-06)	7.9480E+01(3.8719E-07)	7.9480E+01(6.7425E-07)	7.9480E+01(4.0837E-07)
ES	Sphere	10	1.0193E+02(7.2278E+00)	7.9498E+01(3.9784E-03)	7.9498E+01(2.7493E-03)	7.9492E+01(3.5457E-03)	7.9493E+01(3.6095E-03)
ES	Sphere	40	3.6581E+02(1.9291E+01)	8.0312E+01(7.9824E-02)	8.0209E+01(8.0415E-02)	8.0244E+01(4.7132E-02)	8.0184E+01(3.6097E-02)
ES	Weierstrass	2	7.1363E+01(1.1574E-02)	7.1350E+01(1.1207E-04)	7.1350E+01(7.9706E-05)	7.1350E+01(1.0136E-04)	7.1350E+01(7.7261E-05)
ES	Weierstrass	10	7.9895E+01(1.5052E+00)	7.1973E+01(3.1153E-01)	7.1998E+01(5.5310E-01)	7.1831E+01(2.4448E-01)	7.1972E+01(2.7065E-01)
ES	Weierstrass	40	1.0260E+02(2.3803E+00)	7.8507E+01(1.4572E+00)	7.7809E+01(7.8254E-01)	7.7212E+01(4.9859E-01)	7.7208E+01(7.9584E-01)
ES	Gallagher21	2	-1.0000E+03(1.7170E-03)	-1.0000E+03(2.0020E-11)	-1.0000E+03(8.3777E-12)	-1.0000E+03(5.1342E-12)	-1.0000E+03(2.6860E-12)
ES	Gallagher21	10	-9.6047E+02(1.9401E+01)	-9.9514E+02(6.4983E+00)	-9.9607E+02(5.1229E+00)	-9.9609E+02(5.4220E+00)	-9.9587E+02(5.7568E+00)
ES	Gallagher21	40	-9.1650E+02(1.1241E+00)	-9.8991E+02(8.5294E+00)	-9.9159E+02(8.3005E+00)	-9.9159E+02(8.2984E+00)	-9.9160E+02(8.3058E+00)

Tabla 4.1: Resumen del análisis empírico mono-objetivo para algunos problemas en particular

Esta tabla proporciona una visión general de los datos recopilados de cada prueba realizada, permitiendo evaluar cómo la heterogeneidad afecta la búsqueda de soluciones óptimas.

Los hallazgos sugieren que en ciertos casos, la heterogeneidad tiene un impacto significativo en la identificación de soluciones óptimas. Es notable que, particularmente en valores de heterogeneidad mayores o iguales a 0.5, se observa una mayor eficacia en la obtención de resultados óptimos por parte de los individuos en los problemas propuestos. Además, con el incremento en la dimensionalidad de los problemas, los mejores resultados tienden a agruparse en un rango de heterogeneidad que varía entre el 1 % y el 0.5 %.

Esta tendencia nos ofrece perspectivas valiosas sobre la calibración de la heterogeneidad en algoritmos evolutivos aplicados a problemas mono-objetivo. Los datos indican que ajustes adecuados en los niveles de heterogeneidad pueden conducir a mejoras significativas en la eficacia de estos algoritmos.

4.1.2. Gráficos de convergencia

Para complementar y clarificar estos hallazgos, hemos generado gráficos de convergencia para cada algoritmo y problema. Estos gráficos visualizan cómo los valores de heterogeneidad afectan la convergencia de las soluciones en diferentes contextos.

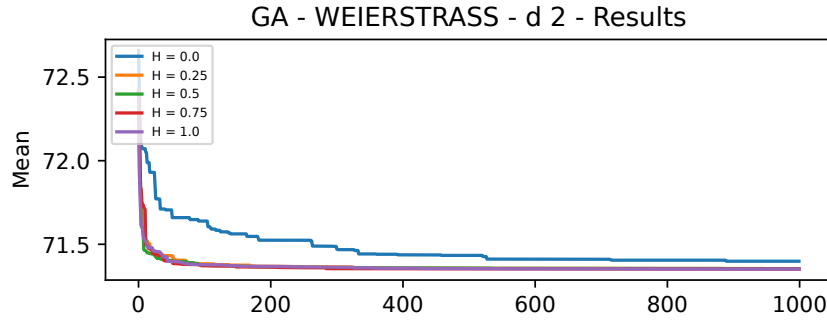


Figura 4.1: Gráfico de convergencia para el algoritmo genético en problemas de baja dimensionalidad.

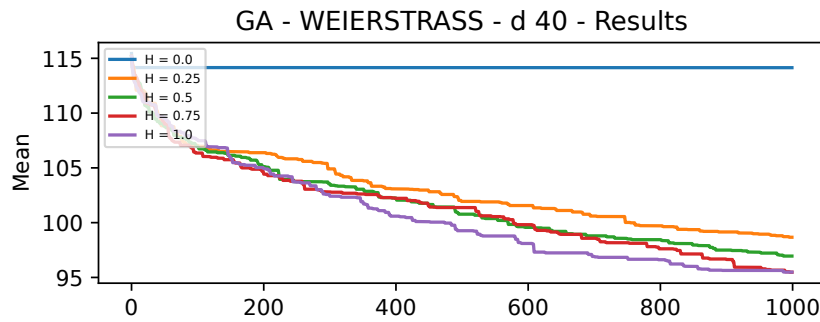


Figura 4.2: Gráfico de convergencia para el algoritmo genético en problemas de alta dimensionalidad.

En la Figura 4.1, correspondiente a un problema de baja dimensionalidad, se observa que los valores de heterogeneidad convergen y obtienen resultados en regiones similares. Por otro lado, la Figura 4.2, que representa un problema de alta dimensionalidad, muestra que para valores bajos de heterogeneidad la convergencia es menos rápida en comparación con valores altos. Patrones similares se observan en las Figuras 4.3, 4.4, 4.5 y 4.6 para otros algoritmos.

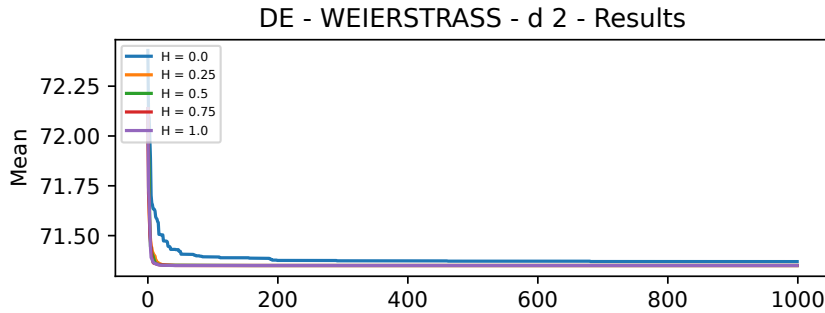


Figura 4.3: Evolución Diferencial - Dos dimensiones

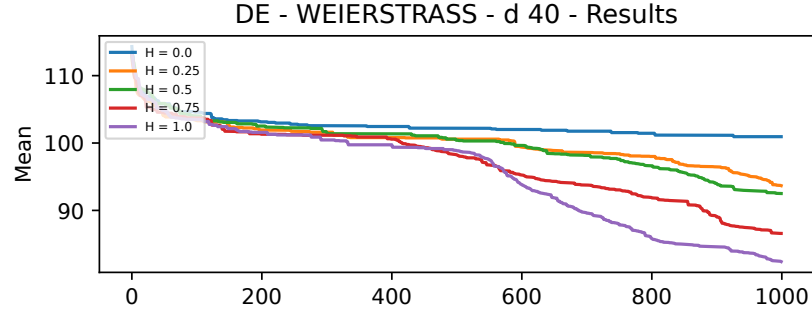


Figura 4.4: Evolución Diferencial - Cuarenta dimensiones

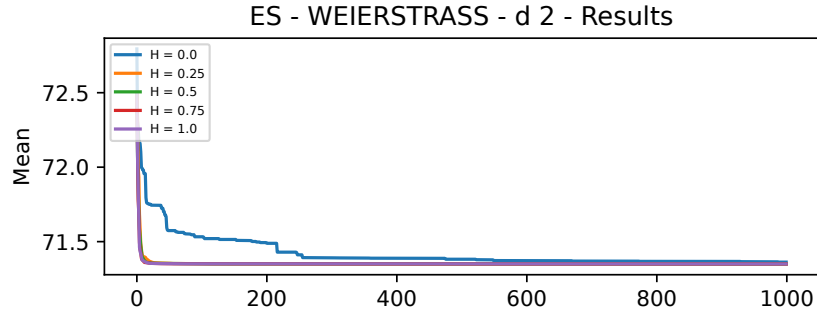


Figura 4.5: Estrategias Evolutivas - Dos dimensiones

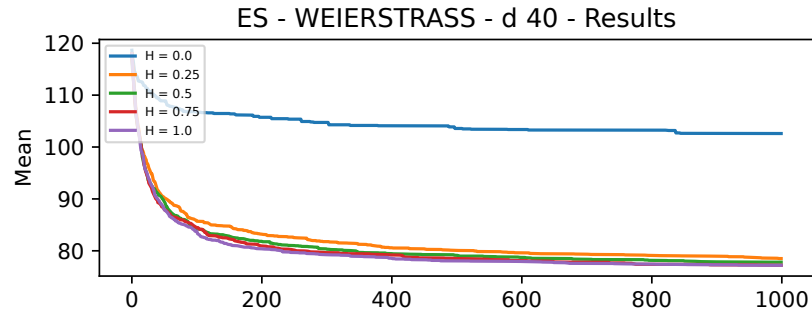


Figura 4.6: Estrategias Evolutivas - Cuarenta dimensiones

Estas representaciones visuales refuerzan los resultados obtenidos y proporcionan una comprensión más profunda de la influencia de la heterogeneidad en el proceso de optimización, dependiendo de la dimensionalidad del problema y el tipo de algoritmo utilizado.

4.1.3. Pruebas estadísticas

En esta subsección, abordamos el análisis estadístico exhaustivo realizado para respaldar y validar los hallazgos obtenidos en los experimentos de optimización mono-objetivo.

Análisis de Kruskal-Wallis

La prueba de Kruskal-Wallis se aplicó para evaluar la significancia estadística de la heterogeneidad en los resultados. Los resultados de esta prueba vistos en la Tabla 4.2, muestran que se rechaza la hipótesis nula de que los datos han sido generados de la misma distribución.

Algoritmo	Problema	Dimensiones	Valor - P
DE	Sphere	2	$3,546009 \times 10^{-20}$
DE	Sphere	10	$4,036662 \times 10^{-13}$
DE	Sphere	40	$1,172101 \times 10^{-12}$
DE	Weierstrass	2	$1,970363 \times 10^{-19}$
DE	Weierstrass	10	$7,260672 \times 10^{-10}$
DE	Weierstrass	40	$1,443381 \times 10^{-7}$
DE	Gallagher21	2	$3,922149 \times 10^{-13}$
DE	Gallagher21	10	$3,075205 \times 10^{-11}$
DE	Gallagher21	40	$3,906155 \times 10^{-11}$
ES	Sphere	2	$2,747200 \times 10^{-11}$
ES	Sphere	10	$5,291616 \times 10^{-14}$
ES	Sphere	40	$8,528497 \times 10^{-14}$
ES	Weierstrass	2	$2,075988 \times 10^{-11}$
ES	Weierstrass	10	$4,230093 \times 10^{-10}$
ES	Weierstrass	40	$1,239032 \times 10^{-11}$
ES	Gallagher21	2	$1,030112 \times 10^{-10}$
ES	Gallagher21	10	$1,106032 \times 10^{-8}$
ES	Gallagher21	40	$6,883057 \times 10^{-10}$
GA	Sphere	2	$5,427408 \times 10^{-14}$
GA	Sphere	10	$1,991402 \times 10^{-14}$
GA	Sphere	40	$1,002805 \times 10^{-11}$
GA	Weierstrass	2	$1,676884 \times 10^{-10}$
GA	Weierstrass	10	$3,367596 \times 10^{-10}$
GA	Weierstrass	40	$6,195023 \times 10^{-12}$
GA	Gallagher21	2	$8,651438 \times 10^{-8}$
GA	Gallagher21	10	$5,409370 \times 10^{-13}$
GA	Gallagher21	40	$2,851189 \times 10^{-11}$

Tabla 4.2: Resumen de la prueba de Kruskal Wallis para algunos problemas

Los valores de p-valor obtenidos en la prueba de Kruskal-Wallis están dentro del umbral crítico de 0.05, lo que indica que las diferencias observadas son significativas y no atribuibles al azar.

Prueba de Wilcoxon ranksum

Al aplicar la prueba de Wilcoxon ranksum, se muestran resultados consistentes en condiciones de heterogeneidad baja ($h=0.0$) con p-valores cercanos o iguales a 1.000000, indicando un rendimiento estable en entornos homogéneos para todos los problemas y dimensiones, mientras que en niveles de heterogeneidad más altos, los p-valores varían considerablemente, reflejando cómo los ajustes en la heterogeneidad afectan el desempeño del algoritmo.

Algoritmo	Problema	Dimensiones	Heterogeneidad	h=0.0	h=0.25	h=0.5	h=0.75	h=1.0
DE	Gallagher21	2	0.00	-	0.999999	1.000000	1.000000	1.000000
DE	Gallagher21	2	0.25	0.000001	-	0.616977	0.500000	0.424909
DE	Gallagher21	2	0.50	0.000000	0.383023	-	0.393387	0.298931
DE	Gallagher21	2	0.75	0.000000	0.500000	0.606613	-	0.414338
DE	Gallagher21	2	1.00	0.000000	0.575091	0.701069	0.585662	-
DE	Gallagher21	10	0.00	-	1.000000	1.000000	1.000000	1.000000
DE	Gallagher21	10	0.25	0.000000	-	0.019901	0.111754	0.032928
DE	Gallagher21	10	0.50	0.000000	0.980099	-	0.841551	0.543082
DE	Gallagher21	10	0.75	0.000000	0.888246	0.158449	-	0.208539
DE	Gallagher21	10	1.00	0.000000	0.967072	0.456918	0.791461	-
DE	Gallagher21	40	0.00	-	1.000000	1.000000	1.000000	1.000000
DE	Gallagher21	40	0.25	0.000000	-	0.510790	0.991353	0.982567
DE	Gallagher21	40	0.50	0.000000	0.489210	-	0.995295	0.987621
DE	Gallagher21	40	0.75	0.000000	0.008647	0.004705	-	0.500000
DE	Gallagher21	40	1.00	0.000000	0.017433	0.012379	0.500000	-
DE	Sphere	2	0.00	-	1.000000	1.000000	1.000000	1.000000
DE	Sphere	2	0.25	0.000000	-	0.500000	0.500000	0.500000
DE	Sphere	2	0.50	0.000000	0.500000	-	0.500000	0.500000
DE	Sphere	2	0.75	0.000000	0.500000	0.500000	-	0.500000
DE	Sphere	2	1.00	0.000000	0.500000	0.500000	0.500000	-
DE	Sphere	10	0.00	-	1.000000	1.000000	1.000000	1.000000
DE	Sphere	10	0.25	0.000000	-	0.007456	0.000219	0.001463
DE	Sphere	10	0.50	0.000000	0.992544	-	0.066642	0.271386
DE	Sphere	10	0.75	0.000000	0.999781	0.933358	-	0.810334
DE	Sphere	10	1.00	0.000000	0.998537	0.728614	0.189666	-
DE	Sphere	40	0.00	-	1.000000	1.000000	1.000000	1.000000
DE	Sphere	40	0.25	0.000000	-	0.990698	0.996853	0.999975
DE	Sphere	40	0.50	0.000000	0.009302	-	0.759066	0.999468
DE	Sphere	40	0.75	0.000000	0.003147	0.240934	-	0.981368
DE	Sphere	40	1.00	0.000000	0.000025	0.000532	0.018632	-
DE	Weierstrass	2	0.00	-	1.000000	1.000000	1.000000	1.000000
DE	Weierstrass	2	0.25	0.000000	-	0.500000	0.500000	0.393387
DE	Weierstrass	2	0.50	0.000000	0.500000	-	0.500000	0.393387
DE	Weierstrass	2	0.75	0.000000	0.500000	0.500000	-	0.393387
DE	Weierstrass	2	1.00	0.000000	0.606613	0.606613	0.606613	-
DE	Weierstrass	10	0.00	-	1.000000	1.000000	1.000000	1.000000
DE	Weierstrass	10	0.25	0.000000	-	0.767413	0.821136	0.467661
DE	Weierstrass	10	0.50	0.000000	0.232587	-	0.616977	0.178864
DE	Weierstrass	10	0.75	0.000000	0.178864	0.383023	-	0.178864
DE	Weierstrass	10	1.00	0.000000	0.532339	0.821136	0.821136	-
DE	Weierstrass	40	0.00	-	0.999870	0.999992	0.999995	0.999999
DE	Weierstrass	40	0.25	0.000130	-	0.775596	0.965034	0.997102
DE	Weierstrass	40	0.50	0.000008	0.224404	-	0.834923	0.987621
DE	Weierstrass	40	0.75	0.000005	0.034966	0.165077	-	0.938446
DE	Weierstrass	40	1.00	0.000001	0.002898	0.012379	0.061554	-

Tabla 4.3: Resumen de la prueba de Willcoxon ranksum en evolución diferencial para algunos problemas

Algoritmo	Problema	Dimensiones	Heterogeneidad	h=0.0	h=0.25	h=0.5	h=0.75	h=1.0
ES	Gallagher21	2	0.0	-	1.000000	1.000000	1.000000	1.000000
ES	Gallagher21	2	0.25	0.000000	-	0.933358	0.988048	0.990354
ES	Gallagher21	2	0.5	0.000000	0.066642	-	0.719574	0.737507
ES	Gallagher21	2	0.75	0.000000	0.011952	0.280426	-	0.467661
ES	Gallagher21	2	1.0	0.000000	0.009646	0.262493	0.532339	-
ES	Gallagher21	10	0.0	-	1.000000	1.000000	1.000000	1.000000
ES	Gallagher21	10	0.25	0.000000	-	0.724112	0.696357	0.877616
ES	Gallagher21	10	0.5	0.000000	0.275888	-	0.446207	0.696357
ES	Gallagher21	10	0.75	0.000000	0.303643	0.553793	-	0.715000
ES	Gallagher21	10	1.0	0.000000	0.122384	0.303643	0.285000	-
ES	Gallagher21	40	0.0	-	1.000000	1.000000	1.000000	1.000000
ES	Gallagher21	40	0.25	0.000000	-	0.813978	0.813978	0.898199
ES	Gallagher21	40	0.5	0.000000	0.186022	-	0.616977	0.677189
ES	Gallagher21	40	0.75	0.000000	0.186022	0.383023	-	0.543082
ES	Gallagher21	40	1.0	0.000000	0.101801	0.322811	0.456918	-
ES	Sphere	2	0.0	-	1.000000	1.000000	1.000000	1.000000
ES	Sphere	2	0.25	0.000000	-	0.995295	0.989999	0.999468
ES	Sphere	2	0.5	0.000000	0.004705	-	0.532339	0.834923
ES	Sphere	2	0.75	0.000000	0.010001	0.467661	-	0.877616
ES	Sphere	2	1.0	0.000000	0.000532	0.165077	0.122384	-
ES	Sphere	10	0.0	-	1.000000	1.000000	1.000000	1.000000
ES	Sphere	10	0.25	0.000000	-	0.616977	0.999969	0.999870
ES	Sphere	10	0.5	0.000000	0.383023	-	0.999994	0.999945
ES	Sphere	10	0.75	0.000000	0.000031	0.000006	-	0.258104
ES	Sphere	10	1.0	0.000000	0.000130	0.000055	0.741896	-
ES	Sphere	40	0.0	-	1.000000	1.000000	1.000000	1.000000
ES	Sphere	40	0.25	0.000000	-	0.999639	0.995295	0.999999
ES	Sphere	40	0.5	0.000000	0.000361	-	0.127957	0.982567
ES	Sphere	40	0.75	0.000000	0.004705	0.872043	-	0.999961
ES	Sphere	40	1.0	0.000000	0.000001	0.017433	0.000039	-
ES	Weierstrass	2	0.0	-	1.000000	1.000000	1.000000	1.000000
ES	Weierstrass	2	0.25	0.000000	-	0.985776	0.958292	0.999704
ES	Weierstrass	2	0.5	0.000000	0.014224	-	0.332580	0.916138
ES	Weierstrass	2	0.75	0.000000	0.041708	0.667420	-	0.980099
ES	Weierstrass	2	1.0	0.000000	0.000296	0.083862	0.019901	-
ES	Weierstrass	10	0.0	-	1.000000	1.000000	1.000000	1.000000
ES	Weierstrass	10	0.25	0.000000	-	0.715000	0.935089	0.478428
ES	Weierstrass	10	0.5	0.000000	0.285000	-	0.759066	0.186022
ES	Weierstrass	10	0.75	0.000000	0.064911	0.240934	-	0.049466
ES	Weierstrass	10	1.0	0.000000	0.521572	0.813978	0.950534	-
ES	Weierstrass	40	0.0	-	1.000000	1.000000	1.000000	1.000000
ES	Weierstrass	40	0.25	0.000000	-	0.944751	0.998403	0.997333
ES	Weierstrass	40	0.5	0.000000	0.055249	-	0.994500	0.981368
ES	Weierstrass	40	0.75	0.000000	0.001597	0.005500	-	0.564465
ES	Weierstrass	40	1.0	0.000000	0.002667	0.018632	0.435535	-

Tabla 4.4: Resumen de la prueba de Wilcoxon ranksum en estrategias evolutivas para algunos problemas

Algoritmo	Problema	Dimensiones	Heterogeneidad	h=0.0	h=0.25	h=0.5	h=0.75	h=1.0
GA	Gallagher21	2	0.0	-	0.999998	1.000000	0.999961	0.715000
GA	Gallagher21	2	0.25	0.000002	-	0.696357	0.313164	0.000643
GA	Gallagher21	2	0.5	0.000000	0.303643	-	0.083862	0.000130
GA	Gallagher21	2	0.75	0.000039	0.686836	0.916138	-	0.002667
GA	Gallagher21	2	1.0	0.285000	0.999357	0.999870	0.997333	-
GA	Gallagher21	10	0.0	-	1.000000	1.000000	1.000000	1.000000
GA	Gallagher21	10	0.25	0.000000	-	0.995295	0.999945	0.999990
GA	Gallagher21	10	0.5	0.000000	0.004705	-	0.938446	0.990698
GA	Gallagher21	10	0.75	0.000000	0.000055	0.061554	-	0.893306
GA	Gallagher21	10	1.0	0.000000	0.000010	0.009302	0.106694	-
GA	Gallagher21	40	0.0	-	1.000000	1.000000	1.000000	1.000000
GA	Gallagher21	40	0.25	0.000000	-	0.927952	0.989255	0.999149
GA	Gallagher21	40	0.5	0.000000	0.072048	-	0.854277	0.986726
GA	Gallagher21	40	0.75	0.000000	0.010745	0.145723	-	0.791461
GA	Gallagher21	40	1.0	0.000000	0.000851	0.013274	0.208539	-
GA	Sphere	2	0.0	-	1.000000	1.000000	1.000000	1.000000
GA	Sphere	2	0.25	0.000000	-	0.958292	0.998403	1.000000
GA	Sphere	2	0.5	0.000000	0.041708	-	0.834923	0.999956
GA	Sphere	2	0.75	0.000000	0.001597	0.165077	-	0.999758
GA	Sphere	2	1.0	0.000000	0.000000	0.000044	0.000242	-
GA	Sphere	10	0.0	-	1.000000	1.000000	1.000000	1.000000
GA	Sphere	10	0.25	0.000000	-	0.999673	0.999998	1.000000
GA	Sphere	10	0.5	0.000000	0.000327	-	0.983699	0.998403
GA	Sphere	10	0.75	0.000000	0.000002	0.016301	-	0.828118
GA	Sphere	10	1.0	0.000000	0.000000	0.001597	0.171882	-
GA	Sphere	40	0.0	-	1.000000	1.000000	1.000000	1.000000
GA	Sphere	40	0.25	0.000000	-	0.828118	0.965034	0.999673
GA	Sphere	40	0.5	0.000000	0.171882	-	0.821136	0.999561
GA	Sphere	40	0.75	0.000000	0.034966	0.178864	-	0.960648
GA	Sphere	40	1.0	0.000000	0.000327	0.000439	0.039352	-
GA	Weierstrass	2	0.0	-	1.000000	1.000000	1.000000	1.000000
GA	Weierstrass	2	0.25	0.000000	-	0.313164	0.975846	0.982567
GA	Weierstrass	2	0.5	0.000000	0.686836	-	0.993083	0.998537
GA	Weierstrass	2	0.75	0.000000	0.024154	0.006917	-	0.342463
GA	Weierstrass	2	1.0	0.000000	0.017433	0.001463	0.657537	-
GA	Weierstrass	10	0.0	-	1.000000	1.000000	1.000000	1.000000
GA	Weierstrass	10	0.25	0.000000	-	0.902926	0.962894	0.950534
GA	Weierstrass	10	0.5	0.000000	0.097074	-	0.791461	0.446207
GA	Weierstrass	10	0.75	0.000000	0.037106	0.208539	-	0.435535
GA	Weierstrass	10	1.0	0.000000	0.049466	0.553793	0.564465	-
GA	Weierstrass	40	0.0	-	1.000000	1.000000	1.000000	1.000000
GA	Weierstrass	40	0.25	0.000000	-	0.989999	0.999415	0.999673
GA	Weierstrass	40	0.5	0.000000	0.010001	-	0.960648	0.962894
GA	Weierstrass	40	0.75	0.000000	0.000585	0.039352	-	0.575091
GA	Weierstrass	40	1.0	0.000000	0.000327	0.037106	0.424909	-

Tabla 4.5: Resumen de la prueba de Willcoxon ranksum en el algoritmo genético para algunos problemas

Los resultados de esta prueba vistos en las Tablas 4.3, 4.4 y 4.5 sugieren que el elitismo no es necesariamente la mejor estrategia en todos los casos, evidenciando la importancia de ajustar el nivel de heterogeneidad en función del problema específico.

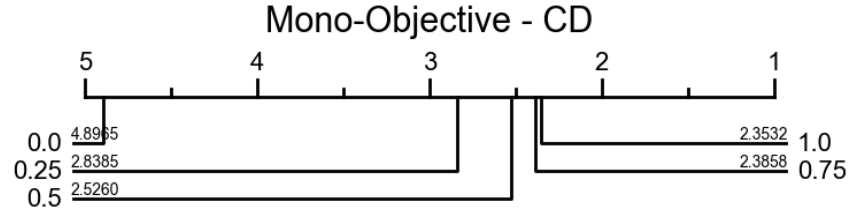


Figura 4.7: Gráfica de diferencias críticas mono-objetivo

Para ilustrar de manera más clara los resultados obtenidos, en la Figura 4.7 se presenta la gráfica de distancias críticas. Esta gráfica proporciona un ranking de los valores de heterogeneidad basados en su rendimiento medio en las pruebas, donde el valor 1 corresponde al mejor rendimiento, mientras que el valor 5 representa el peor.

Las líneas horizontales conectan aquellos valores de heterogeneidad cuyos rendimientos no presentan diferencias estadísticamente significativas. A partir de la gráfica, se pueden destacar los siguientes puntos:

- Los niveles de heterogeneidad 0.75 y 1.0 muestran un rendimiento similar, ya que están conectados por una línea, lo que indica que no existe una diferencia estadísticamente significativa entre ellos.
- El valor de heterogeneidad 0.5 , a pesar de mostrar un rendimiento intermedio, no evidencia diferencias significativas con respecto a los valores 0.25 y 0.0 , los cuales obtuvieron los peores resultados en el ranking.

En términos generales, los resultados sugieren que los niveles más altos de heterogeneidad (0.75 y 1.0) están asociados a un rendimiento superior en comparación con los niveles más bajos. Sin embargo, no se observa una diferencia significativa entre los niveles altos de heterogeneidad.

Conteo de Borda

Para reforzar aún más nuestros hallazgos, analizamos la Tabla 4.6 de conteo de Borda, que destaca la frecuencia con la que diferentes valores de heterogeneidad resultaron ser los más efectivos.

Algoritmo	Dimensiones	h=0.0	h=0.25	h=0.5	h=0.75	h=1.0
DE	2	25	63	65	60	48
DE	3	24	68	64	55	48
DE	5	24	63	53	48	49
DE	10	24	65	53	53	55
DE	20	24	56	56	60	61
DE	40	24	50	57	71	88
ES	2	24	49	58	64	62
ES	3	24	52	58	59	65
ES	5	24	56	60	61	62
ES	10	24	52	56	63	62
ES	20	24	52	56	59	69
ES	40	24	51	60	69	75
GA	2	24	50	54	61	58
GA	3	24	48	63	66	66
GA	5	24	47	54	66	74
GA	10	24	48	63	74	89
GA	20	24	48	63	78	92
GA	40	24	48	61	79	90
Total	-	433	966	1054	1146	1213

Tabla 4.6: Resumen de conteos de Borda

Se observó que, con el incremento de las dimensiones, existe una tendencia generalizada entre los algoritmos para mejorar su rendimiento relativo o su clasificación en comparación con otros bajo condiciones de alta heterogeneidad. Este fenómeno sugiere que las estrategias implementadas por dichos algoritmos se benefician de un mayor grado de elitismo en espacios de búsqueda más complejos. Esta tendencia es particularmente notable en el algoritmo genético (GA), el cual exhibe una alta predisposición hacia el elitismo. Sin embargo, se ha identificado que algoritmos como la Evolución Diferencial (DE) obtienen mayores beneficios en entornos de baja heterogeneidad.

4.1.4. Explicabilidad

Mediante el uso de P-Flacco, exploramos las características ELA de las funciones objetivo, las cuales proyectamos utilizando un mapa de calor presente en la Figura 4.8 para visualizar la correlación entre estas características. Identificamos que, a medida que aumenta la dimensionalidad de los problemas, ciertas características, como la distancia media y el ángulo medio, exhiben un impacto significativo en la necesidad de heterogeneidad. Específicamente, observamos que el ángulo medio alcanza valores absolutos de 1 en dimensiones de 40, lo que sugiere que estos factores son cruciales en la optimización de los algoritmos evolutivos.

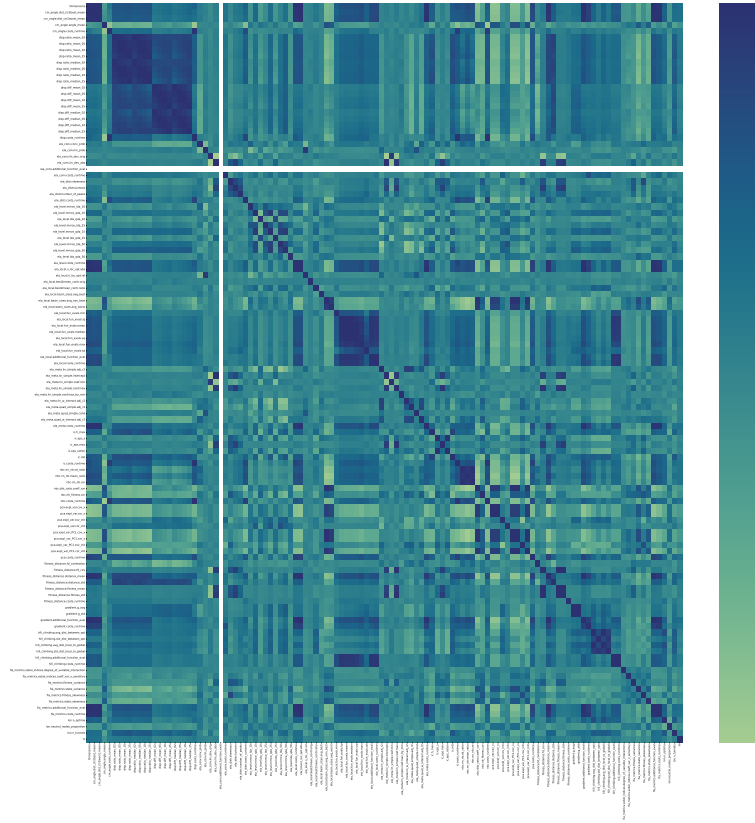


Figura 4.8: Mapa de correlación de características de los espacios mono-objetivo

Para validar nuestra hipótesis sobre la influencia de estas características en los resultados de optimización, recurrimos al análisis proporcionado por el paquete SHAP, enfocado en el modelo **RandomForestRegressor** aplicado a nuestros datos experimentales. Los resultados del análisis SHAP que se muestran en la Figura 4.9 confirman que las características más influyentes en los resultados son, efectivamente, los algoritmos utilizados y las métricas de distancia media y propagación.

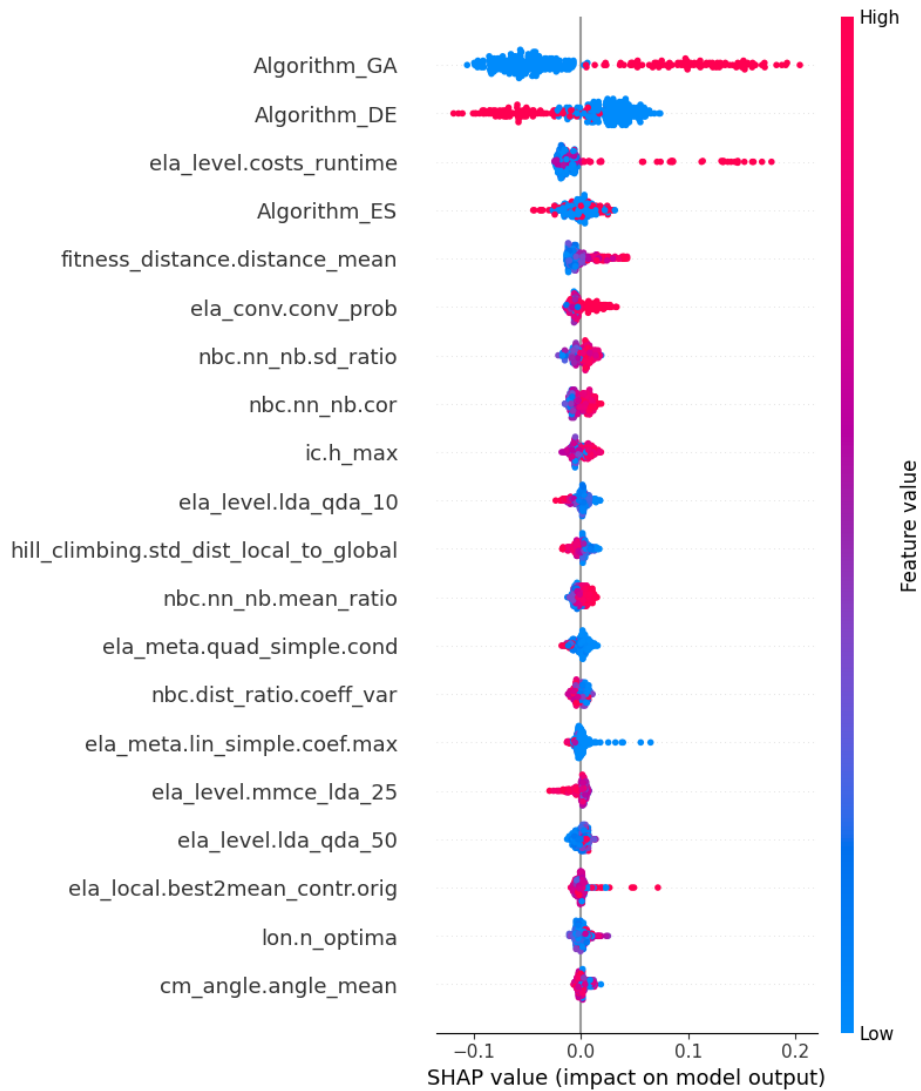


Figura 4.9: Análisis Mono-Objetivo de SHAP destacando las características más influyentes en la elección de heterogeneidad.

Estos hallazgos subrayan la importancia de analizar detenidamente las características de los espacios de búsqueda para comprender cómo influyen en la selección óptima de la heterogeneidad, dependiendo del algoritmo utilizado y la dimensionalidad del problema.

Este enfoque de explicabilidad nos permite no solo entender mejor los mecanismos detrás de los algoritmos de optimización sino también guiar la selección de parámetros de heterogeneidad de manera más informada y efectiva.

4.2. Multi-objetivo

En esta sección, desplazamos nuestro enfoque hacia el análisis de problemas multi-objetivo, con el propósito de explorar cómo la heterogeneidad afecta el comportamiento en los resultados para los conjuntos de prueba seleccionados.

4.2.1. Análisis empírico

Para proporcionar una visión detallada de los efectos de la heterogeneidad en contextos multi-objetivo, analizamos las Tablas 4.7 y 4.8. Estas tablas resumen los resultados recopilados de las pruebas realizadas, ilustrando cómo diferentes niveles de heterogeneidad impactan el rendimiento de los algoritmos en estos problemas complejos.

Profundizando en el análisis empírico de los resultados obtenidos en los experimentos de optimización multi-objetivo, examinando las tendencias observadas en las métricas de rendimiento y su implicación.

Análisis de hipervolumen (HV)

Los resultados obtenidos por NSGA-II, medidos a través del indicador de hipervolumen, se resumen en las Tablas 4.7 y 4.8.

Algoritmo	Problema	Dimensiones	h=0.0	h=0.25	h=0.5	h=0.75	h=1.0
NSGA-II	1	2	1.7632E-02(3.8790E-17)	1.1990E+00(2.7230E-03)	1.2016E+00(6.6671E-04)	1.2025E+00(1.5522E-05)	1.2025E+00(2.7910E-06)
NSGA-II	1	40	1.9257E-02(1.7656E-03)	1.1221E+00(5.5496E-03)	1.1492E+00(6.0130E-03)	1.1668E+00(4.4089E-03)	1.1992E+00(7.0035E-04)
NSGA-II	2	2	1.3883E-02(2.2272E-16)	1.2078E+00(9.3559E-04)	1.2085E+00(2.4423E-04)	1.2088E+00(3.2634E-06)	1.2088E+00(1.3674E-06)
NSGA-II	2	40	1.1478E-02(1.0120E-03)	1.1957E+00(3.3989E-03)	1.2017E+00(1.4748E-03)	1.2034E+00(1.4063E-03)	1.2094E+00(1.8934E-05)
NSGA-II	3	2	6.6582E-02(1.4238E-17)	1.2098E+00(1.2180E-04)	1.2099E+00(1.0630E-04)	1.2100E+00(6.4450E-06)	1.2100E+00(6.0390E-10)
NSGA-II	3	40	2.9621E-02(4.1218E-03)	1.1591E+00(5.7317E-03)	1.1785E+00(3.3430E-03)	1.1904E+00(3.2868E-03)	1.2059E+00(8.0616E-04)
NSGA-II	4	2	1.0000E-02(1.1368E-16)	1.2090E+00(7.8701E-04)	1.2095E+00(4.2270E-04)	1.2098E+00(1.6722E-06)	1.2098E+00(3.3603E-07)
NSGA-II	4	40	2.5099E-02(5.1962E-03)	1.1726E+00(6.1072E-03)	1.1794E+00(4.1628E-03)	1.1890E+00(4.8146E-03)	1.2076E+00(2.0231E-04)
NSGA-II	5	2	4.2280E-02(4.6819E-17)	1.1763E+00(2.0154E-02)	1.1868E+00(1.4926E-02)	1.2005E+00(1.9442E-03)	1.2039E+00(1.9354E-05)
NSGA-II	5	40	1.8789E-02(1.5293E-03)	1.0152E+00(1.2330E-02)	1.0632E+00(1.3717E-02)	1.0828E+00(1.2615E-02)	1.1710E+00(5.6621E-03)
NSGA-II	6	2	7.6170E-02(2.8689E-16)	1.1987E+00(2.1951E-03)	1.2008E+00(1.3876E-03)	1.2026E+00(1.4024E-05)	1.2027E+00(1.3013E-05)
NSGA-II	6	40	7.8328E-02(6.5815E-03)	1.1795E+00(8.1408E-03)	1.1931E+00(4.4673E-03)	1.1979E+00(2.4535E-03)	1.2092E+00(2.7847E-04)
NSGA-II	7	2	1.0000E-02(4.9252E-17)	1.2077E+00(2.0100E-03)	1.2091E+00(7.6545E-04)	1.2094E+00(4.1021E-04)	1.2100E+00(3.9491E-07)
NSGA-II	7	40	4.2232E-02(1.6620E-02)	1.1937E+00(3.9636E-03)	1.1991E+00(1.4237E-03)	1.2054E+00(1.8176E-04)	1.2096E+00(1.8176E-04)
NSGA-II	8	2	8.0758E-02(4.5757E-05)	1.2095E+00(1.1651E-04)	1.2098E+00(1.7864E-04)	1.2099E+00(6.9521E-05)	1.2100E+00(3.1512E-06)
NSGA-II	8	40	1.1283E-01(1.1402E-02)	1.2087E+00(4.9746E-04)	1.2098E+00(1.0254E-04)	1.2099E+00(4.3878E-05)	1.2100E+00(1.7562E-05)
NSGA-II	9	2	1.0000E-02(4.9239E-17)	1.2096E+00(3.1461E-04)	1.2099E+00(1.4002E-04)	1.2100E+00(1.7740E-06)	1.2100E+00(2.9764E-08)
NSGA-II	9	40	2.0952E-02(2.4162E-03)	1.1782E+00(3.5731E-03)	1.1925E+00(3.3406E-03)	1.2000E+00(1.6453E-03)	1.2086E+00(8.2078E-05)
NSGA-II	10	2	4.9920E-02(9.8986E-03)	1.2085E+00(2.4228E-03)	1.2088E+00(2.0325E-03)	1.2098E+00(5.0397E-06)	1.2098E+00(1.0588E-07)
NSGA-II	10	40	1.4526E-02(5.4919E-03)	7.3265E-01(9.1413E-02)	9.4849E-01(4.8695E-02)	1.0385E+00(3.8375E-02)	1.1469E+00(6.1918E-03)
NSGA-III	1	2	2.1189E-01(4.2572E-02)	1.1987E+00(4.0672E-03)	1.2004E+00(2.1320E-03)	1.2015E+00(1.0672E-03)	1.2025E+00(3.6680E-05)
NSGA-III	1	40	2.4929E-02(1.9641E-03)	1.1309E+00(7.1112E-03)	1.1464E+00(6.7775E-03)	1.1535E+00(5.9832E-03)	1.1771E+00(3.0917E-03)
NSGA-III	2	2	2.5915E-01(5.1953E-02)	1.2084E+00(4.8479E-04)	1.2086E+00(1.9297E-04)	1.2087E+00(6.8345E-05)	1.2087E+00(3.1652E-05)
NSGA-III	2	40	2.3999E-02(4.9295E-03)	1.1903E+00(4.0018E-03)	1.1974E+00(2.1016E-03)	1.1998E+00(1.8181E-03)	1.2080E+00(3.2929E-04)
NSGA-III	3	2	3.8244E-01(8.1350E-02)	1.2099E+00(1.2785E-04)	1.2099E+00(1.2010E-04)	1.2099E+00(1.9936E-04)	1.2100E+00(4.7475E-09)
NSGA-III	3	40	3.7200E-02(6.6329E-03)	1.1605E+00(6.9607E-03)	1.1760E+00(5.5053E-03)	1.1801E+00(6.2836E-03)	1.1917E+00(3.7321E-03)
NSGA-III	4	2	3.8150E-01(9.1889E-02)	1.2089E+00(1.4132E-03)	1.2096E+00(3.3094E-04)	1.2097E+00(8.5522E-05)	1.2098E+00(2.5915E-06)
NSGA-III	4	40	3.8852E-02(9.7052E-03)	1.1617E+00(5.6435E-03)	1.1710E+00(6.3789E-03)	1.1766E+00(2.0207E-03)	1.1963E+00(2.3895E-03)
NSGA-III	5	2	2.5182E-01(5.0982E-02)	1.1915E+00(1.2725E-02)	1.1889E+00(1.2453E-02)	1.1970E+00(6.6614E-03)	1.2038E+00(1.2030E-04)
NSGA-III	5	40	2.4949E-02(2.7451E-03)	1.0117E+00(1.6799E-02)	1.0447E+00(1.6543E-02)	1.0548E+00(9.9534E-03)	1.1120E+00(8.5769E-03)
NSGA-III	6	2	4.0083E-01(5.7904E-02)	1.2000E+00(2.5722E-03)	1.2013E+00(9.2849E-04)	1.2006E+00(1.1969E-03)	1.2026E+00(6.6429E-05)
NSGA-III	6	40	9.6836E-02(5.8516E-03)	1.1821E+00(6.7963E-03)	1.1872E+00(6.0552E-03)	1.1924E+00(3.1341E-03)	1.2021E+00(2.0071E-03)
NSGA-III	7	2	2.2438E-01(5.4920E-02)	1.2083E+00(1.7719E-03)	1.2094E+00(7.0602E-04)	1.2091E+00(8.5949E-04)	1.2100E+00(4.8978E-07)
NSGA-III	7	40	7.4073E-02(1.6934E-02)	1.1798E+00(4.9031E-03)	1.1980E+00(3.4947E-03)	1.2017E+00(2.0035E-03)	1.2044E+00(1.2711E-03)
NSGA-III	8	2	3.9594E-01(4.6685E-02)	1.2098E+00(2.1445E-04)	1.2098E+00(1.3934E-04)	1.2098E+00(1.2994E-04)	1.2100E+00(4.5162E-06)
NSGA-III	8	40	1.4146E-01(1.0887E-02)	1.2032E+00(2.5722E-03)	1.2093E+00(3.0847E-04)	1.2097E+00(1.2447E-04)	1.2096E+00(1.6055E-04)
NSGA-III	9	2	2.4416E-01(6.8229E-02)	1.2098E+00(1.2111E-04)	1.2099E+00(4.0012E-05)	1.2099E+00(2.6158E-05)	1.2100E+00(4.4230E-07)
NSGA-III	9	40	3.1027E-02(2.7627E-03)	1.1773E+00(4.7067E-03)	1.1880E+00(3.5784E-03)	1.1901E+00(3.3627E-03)	1.2043E+00(1.0419E-03)
NSGA-III	10	2	5.1514E-01(5.0906E-02)	1.2091E+00(1.2064E-03)	1.2095E+00(5.9072E-04)	1.2096E+00(4.5283E-04)	1.2096E+00(1.4459E-05)
NSGA-III	10	40	2.6006E-02(5.8383E-03)	7.2076E-01(8.2484E-02)	8.2769E-01(8.0112E-02)	8.8582E-01(6.8925E-02)	1.0681E+00(1.9711E-02)

Tabla 4.7: Resumen de resultados BiBBOB con hipervolumen

Algoritmo	Problema	K	h=0.0	h=0.25	h=0.5	h=0.75	h=1.0
NSGA-II	WFG1	3	1.0658E-14(0.0000E+00)	2.3432E+01(1.7814E+00)	2.5693E+01(1.3080E+00)	5.3147E+01(1.0799E+00)	5.7917E+01(1.6687E+00)
NSGA-II	WFG1	5	1.7053E-12(0.0000E+00)	2.2859E+03(1.6067E+02)	2.2153E+03(1.0036E+02)	4.4995E+03(7.5586E+01)	4.6695E+03(9.1086E+01)
NSGA-II	WFG2	3	5.0284E-01(1.1083E+00)	7.1435E+01(4.7282E+00)	8.4288E+01(4.9502E+00)	9.9183E+01(3.2257E+00)	9.8136E+01(5.4061E+00)
NSGA-II	WFG2	5	3.4414E+00(8.8640E+00)	7.2924E+03(8.3749E+02)	8.1449E+03(5.2702E+02)	1.0284E+04(1.1538E+01)	1.0292E+04(1.2327E+01)
NSGA-II	WFG3	3	4.6755E+00(3.5768E+00)	4.3496E+01(7.1499E+00)	4.2122E+01(2.4497E+00)	7.5480E+01(1.1179E+01)	7.5826E+01(1.0216E-01)
NSGA-II	WFG3	5	1.8182E+02(1.3077E+02)	4.1160E+03(2.4565E+02)	4.3343E+03(2.2694E+02)	7.1865E+03(5.6788E+01)	7.2426E+03(3.5297E+01)
NSGA-II	WFG4	3	8.7789E-01(2.3315E-01)	3.7005E+01(4.8181E+00)	4.7756E+01(3.8061E+00)	7.3964E+01(4.0269E-01)	7.4611E+01(2.7216E-01)
NSGA-II	WFG4	5	4.0334E+01(8.2917E+00)	2.1238E+03(2.8864E+02)	2.8380E+03(6.0052E+02)	7.2625E+03(3.4983E+02)	7.4471E+03(2.6836E+02)
NSGA-II	WFG5	3	1.7430E+00(8.9520E-01)	3.0558E+01(4.7061E+00)	2.9985E+01(5.1547E+00)	7.1675E+01(2.7048E-01)	7.1784E+01(4.2525E-01)
NSGA-II	WFG5	5	1.3096E+02(5.3685E+01)	1.4948E+03(2.6938E+02)	3.8280E+03(6.0893E+02)	6.8859E+03(1.5449E+02)	7.2818E+03(1.3192E+02)
NSGA-II	WFG6	3	2.0802E+00(1.3617E+00)	3.7578E+01(1.0423E+01)	4.3834E+01(9.8491E+00)	7.2047E+01(4.2122E-01)	7.2617E+01(3.6653E-01)
NSGA-II	WFG6	5	4.8432E+01(2.4592E+01)	1.8461E+03(5.1133E+02)	2.6766E+03(5.6379E+02)	7.3009E+03(2.3975E+02)	7.4327E+03(1.6349E+02)
NSGA-II	WFG7	3	5.5033E-03(1.2611E-02)	3.5590E+01(4.3197E+00)	4.2380E+01(2.7246E+00)	7.4778E+01(5.7333E-01)	7.5398E+01(4.2521E-01)
NSGA-II	WFG7	5	7.5651E+01(5.5494E+01)	2.8051E+03(1.9664E+02)	3.3382E+03(1.9345E+02)	7.2612E+03(2.0541E+02)	7.4388E+03(1.8143E+02)
NSGA-II	WFG8	3	1.0553E+00(8.5153E-01)	2.8619E+01(3.8099E+00)	3.4401E+01(3.6869E+00)	6.6635E+01(5.1002E-01)	6.8032E+01(2.8151E-01)
NSGA-II	WFG8	5	4.1957E+01(2.2711E+01)	1.6900E+03(2.4758E+02)	2.3718E+03(1.8411E+02)	6.0115E+03(1.8918E+02)	6.1372E+03(1.0655E+02)
NSGA-II	WFG9	3	6.6021E-02(4.9340E-03)	4.5999E+01(3.8896E+00)	5.3707E+01(3.1752E+00)	6.7996E+01(1.6031E+00)	6.7908E+01(1.5281E+00)
NSGA-II	WFG9	5	9.9691E+00(5.0953E+00)	3.1386E+03(4.3593E+02)	3.8913E+03(4.0748E+02)	6.3100E+03(1.9283E+02)	6.5530E+03(1.1814E+02)
NSGA-III	WFG1	3	1.3530E-01(4.4051E-02)	2.4269E+01(1.2415E+00)	3.5767E+01(1.6013E+00)	5.5443E+01(1.7266E+00)	6.3523E+01(1.2975E+00)
NSGA-III	WFG1	5	1.2666E+01(3.4929E+00)	2.2122E+03(1.3100E+02)	3.1042E+03(1.1458E+02)	4.4234E+03(1.1443E+02)	4.7417E+03(1.1579E+02)
NSGA-III	WFG2	3	1.6700E+01(5.1059E+00)	7.6305E+01(2.8631E+00)	7.8478E+01(4.8778E+00)	9.7611E+01(5.3311E+00)	9.8471E+01(5.4989E+00)
NSGA-III	WFG2	5	6.9919E+02(3.5615E+02)	7.7175E+03(4.8185E+02)	7.9954E+03(1.0237E+03)	1.0042E+04(3.7903E+02)	1.0319E+04(1.2694E+01)
NSGA-III	WFG3	3	1.0389E+01(3.6995E+00)	4.3638E+01(4.0713E+00)	5.9539E+01(2.6997E+00)	7.3729E+01(2.5634E-01)	7.3995E+01(2.1104E-01)
NSGA-III	WFG3	5	7.4726E+02(2.8657E+02)	4.3454E+03(3.2418E+02)	6.8410E+03(8.6320E+01)	6.9922E+03(3.0004E+01)	7.0032E+03(3.4378E+01)
NSGA-III	WFG4	3	3.0494E+00(7.2583E-01)	4.1379E+01(5.1188E+00)	6.4367E+01(9.9168E-01)	7.2747E+01(3.6697E-01)	7.7353E+01(6.9158E-02)
NSGA-III	WFG4	5	1.7978E+02(3.3397E+01)	2.9421E+03(2.2752E+02)	6.8878E+03(1.4693E+02)	7.9101E+03(8.1412E+01)	8.9753E+03(3.8997E+01)
NSGA-III	WFG5	3	5.4455E+00(1.0381E+00)	3.2694E+01(4.1825E+00)	6.7061E+01(5.6137E-01)	7.1448E+01(2.6398E-01)	7.4106E+01(2.6197E-01)
NSGA-III	WFG5	5	3.6621E+02(7.7810E+01)	2.4523E+03(4.5823E+02)	7.3802E+03(1.6664E+02)	8.1795E+03(4.4347E+01)	8.7270E+03(1.7067E+01)
NSGA-III	WFG6	3	5.2386E+00(8.2209E-01)	4.1836E+01(6.9853E+00)	3.3885E+01(8.1587E+00)	7.1930E+01(3.4415E-01)	7.4726E+01(4.0188E-01)
NSGA-III	WFG6	5	3.1508E+02(6.0188E+01)	2.4704E+03(2.9399E+02)	6.8310E+03(1.8621E+02)	7.9604E+03(7.1953E+01)	8.6857E+03(8.9656E+01)
NSGA-III	WFG7	3	3.0437E+00(1.4228E+00)	4.2109E+01(2.8706E+00)	6.5899E+01(1.2202E+00)	7.4211E+01(2.0979E-01)	7.7589E+01(3.2094E-02)
NSGA-III	WFG7	5	3.6738E+02(1.1086E+02)	3.1897E+03(1.9176E+02)	7.3697E+03(1.1350E+02)	8.3113E+03(6.4070E+01)	9.1296E+03(1.0447E+01)
NSGA-III	WFG8	3	2.6648E+00(1.1001E+00)	2.9975E+01(4.5465E+00)	2.7976E+01(4.4372E+00)	6.7076E+01(5.0398E-01)	7.0959E+01(1.8238E-01)
NSGA-III	WFG8	5	1.5130E+02(3.9786E+01)	2.2096E+03(2.5941E+02)	5.6445E+03(8.6754E+01)	6.4981E+03(3.9869E+02)	8.5705E+03(2.5747E+02)
NSGA-III	WFG9	3	6.4964E-01(8.5365E-01)	4.5294E+01(3.9249E+00)	6.6277E+01(8.9142E-01)	7.0315E+01(1.5077E+00)	7.2209E+01(1.9136E+00)
NSGA-III	WFG9	5	6.5621E+01(7.3099E+01)	3.3359E+03(2.4258E+02)	6.8451E+03(8.2306E+01)	7.2766E+03(3.6846E+01)	7.5449E+03(2.6381E+01)

Tabla 4.8: Resumen de resultados WFG con hipervolumen

En estas se revela que, en la mayoría de los casos, un valor de heterogeneidad $H = 1$ obtuvo los mejores resultados, con ciertas excepciones en múltiples problemas como F1 en BiBBOB para NSGA-III y para WFG2, WFG5, y WFG9 en $k = 3$ con NSGA-III.

Análisis de IGD+

A continuación, presentamos los resultados utilizando el indicador IGD+ en las Tablas 4.9 y 4.10, que respaldan las observaciones hechas con el hipervolumen.

Algoritmo	Problema	K	0.0	0.25	0.5	0.75	1.0
NSGA-II	WFG1	3	2.5711E+00(1.9797E-03)	1.7376E+00(5.3006E-02)	1.6912E+00(3.4967E-02)	1.1556E+00(3.0270E-02)	1.0066E+00(4.5573E-02)
NSGA-II	WFG1	5	3.0209E+00(2.5336E-03)	2.0788E+00(7.9665E-02)	2.1405E+00(3.3975E-02)	1.6866E+00(2.6430E-02)	1.6361E+00(4.0216E-02)
NSGA-II	WFG1	7	3.3319E+00(6.4601E-03)	2.2536E+00(8.4244E-02)	2.4086E+00(4.5030E-02)	2.0549E+00(3.4088E-02)	2.0372E+00(3.6994E-02)
NSGA-II	WFG1	10	3.5900E+00(1.3900E-02)	2.2673E+00(8.8005E-02)	2.4601E+00(1.4297E-01)	2.0374E+00(7.2278E-02)	1.8602E+00(1.3415E-01)
NSGA-II	WFG2	3	5.4570E+00(2.5688E-01)	7.1601E-01(2.2800E-01)	3.3905E-01(1.0615E-01)	7.8740E-02(5.2292E-02)	9.3808E-02(9.6728E-02)
NSGA-II	WFG2	5	1.0591E+01(8.1382E-01)	1.6604E+00(6.6736E-01)	1.1946E+00(2.8881E-01)	4.3194E-01(4.4673E-02)	4.5440E-01(5.6088E-02)
NSGA-II	WFG2	7	1.5095E+01(1.3631E+00)	3.0111E+00(9.6862E-01)	1.9325E+00(3.5199E-01)	6.1885E-01(8.8013E-02)	6.1940E-01(7.8406E-02)
NSGA-II	WFG2	10	2.2137E+01(2.1841E+00)	4.8418E+00(1.5096E+00)	3.2001E+00(7.9964E-01)	7.1074E-01(6.8347E-02)	7.7418E-01(1.2179E-01)
NSGA-II	WFG3	3	3.3061E+00(4.8218E-01)	8.8339E-01(2.4647E-01)	9.3878E-01(1.1505E-01)	4.2239E-02(5.7073E-03)	3.0032E-02(3.3062E-03)
NSGA-II	WFG3	5	3.8878E+00(3.1722E-01)	1.5073E+00(2.3186E-01)	1.3647E+00(1.1946E-01)	1.7643E-01(4.5637E-02)	1.5044E-01(3.2992E-02)
NSGA-II	WFG3	7	3.7347E+00(3.5185E-01)	2.2462E+00(3.2905E-01)	1.9734E+00(3.0050E-01)	2.6062E-01(5.2532E-02)	2.2788E-01(6.4730E-02)
NSGA-II	WFG3	10	3.7005E+00(3.0268E-01)	3.5068E+00(5.4478E-01)	2.8591E+00(5.9588E-01)	4.0515E-01(1.7181E-01)	3.1128E-01(1.2257E-01)
NSGA-II	WFG4	3	4.1709E+00(2.5634E-02)	1.0941E+00(3.2375E-01)	6.3661E-01(1.0876E-01)	1.3147E-01(6.6108E-03)	1.1629E-01(5.3263E-03)
NSGA-II	WFG4	5	6.9204E+00(1.1157E-01)	2.8612E+00(4.4235E-01)	2.4533E+00(6.0301E-01)	7.1513E-01(8.6957E-02)	6.5064E-01(6.0923E-02)
NSGA-II	WFG4	7	9.3485E+00(1.3834E-01)	4.6810E+00(6.2427E-01)	2.6884E+00(2.3983E-01)	1.6811E+00(9.8703E-02)	1.6413E+00(1.1102E-01)
NSGA-II	WFG4	10	1.3982E+01(2.5190E-01)	8.6270E+00(7.5325E-01)	4.7654E+00(4.0291E-01)	2.5365E+00(2.0661E-01)	2.4354E+00(1.4324E-01)
NSGA-II	WFG5	3	4.0013E+00(3.0123E-01)	1.6555E+00(4.6349E-01)	1.8566E+00(4.1917E-01)	1.4558E-01(3.5866E-03)	1.4064E-01(5.1831E-03)
NSGA-II	WFG5	5	7.1411E+00(2.0347E-01)	3.9584E+00(4.6049E-01)	1.6549E+00(4.2519E-01)	7.1619E-01(2.9858E-02)	6.2436E-01(2.9982E-02)
NSGA-II	WFG5	7	9.7053E+00(2.3448E-01)	5.8200E+00(8.2717E-01)	2.3022E+00(1.6122E-01)	1.6633E+00(6.3039E-02)	1.5911E+00(9.4298E-02)
NSGA-II	WFG5	10	1.2940E+01(3.5983E-01)	6.9064E+00(1.2047E+00)	3.2866E+00(2.5342E-01)	2.5650E+00(9.1577E-02)	2.5866E+00(6.9790E-02)
NSGA-II	WFG6	3	3.7049E+00(5.4911E-01)	1.3509E+00(7.6988E-01)	1.0673E+00(5.9792E-01)	1.6503E-01(7.4377E-03)	1.4963E-01(6.2517E-03)
NSGA-II	WFG6	5	7.7200E+00(4.3990E-02)	3.6673E+00(6.0460E-01)	3.3535E+00(3.9256E-01)	7.1257E-01(4.8358E-02)	6.7196E-01(3.3843E-02)
NSGA-II	WFG6	7	1.1475E+01(9.1237E-02)	7.1138E+00(9.8394E-01)	3.3953E+00(3.2811E-01)	1.9765E+00(2.4928E-01)	1.8630E+00(2.0226E-01)
NSGA-II	WFG6	10	1.8180E+01(1.5511E-01)	1.1187E+01(9.7825E-01)	6.0899E+00(6.6231E-01)	3.2042E+00(4.8109E-01)	3.2386E+00(5.9550E-01)
NSGA-II	WFG7	3	4.2390E+00(1.1905E-02)	1.2287E+00(2.7270E-01)	1.0359E+00(1.5403E-01)	1.1263E-01(6.6988E-03)	9.9310E-02(5.6760E-03)
NSGA-II	WFG7	5	7.5895E+00(1.3727E-01)	2.8074E+00(2.1589E-01)	2.4586E+00(1.9758E-01)	7.2286E-01(4.6944E-02)	6.8184E-01(3.6317E-02)
NSGA-II	WFG7	7	1.1388E+01(2.3889E-01)	5.2151E+00(6.2695E-01)	4.7580E+00(2.4821E-01)	1.9498E+00(1.6834E-01)	1.9143E+00(1.4625E-01)
NSGA-II	WFG7	10	1.7613E+01(1.9474E-01)	9.9074E+00(1.2108E+00)	9.1549E+00(5.4191E-01)	3.7396E+00(1.4263E-01)	3.7940E+00(1.1934E-01)
NSGA-II	WFG8	3	4.1584E+00(1.5656E-01)	1.8773E+00(3.5751E-01)	1.5021E+00(3.0692E-01)	2.7566E-01(1.0130E-02)	2.7566E-01(1.0130E-02)
NSGA-II	WFG8	5	7.6838E+00(6.0392E-02)	4.4499E+00(4.5177E-01)	4.1620E+00(4.3615E-01)	1.1572E+00(4.4360E-02)	1.1246E+00(3.3745E-02)
NSGA-II	WFG8	7	1.1596E+01(6.0502E-02)	7.5885E+00(7.7176E-01)	3.5480E+00(3.6449E-01)	2.2956E+00(9.2373E-02)	2.2475E+00(7.5685E-02)
NSGA-II	WFG8	10	1.7786E+01(1.3683E-01)	9.9649E+00(1.6385E+00)	4.6442E+00(5.1340E-01)	3.5489E+00(1.7904E-01)	3.6473E+00(1.2502E-01)
NSGA-II	WFG9	3	4.2723E+00(1.9197E-02)	7.2840E-01(1.0069E-01)	5.3917E-01(8.1846E-02)	2.1080E-01(3.0429E-02)	2.1214E-01(2.9456E-02)
NSGA-II	WFG9	5	8.0862E+00(4.5759E-02)	7.1222E+00(2.6982E-01)	2.2303E+00(1.5347E-01)	1.0887E+00(7.8197E-02)	1.0240E+00(4.9375E-02)
NSGA-II	WFG9	7	1.2152E+01(6.1741E-02)	5.3377E+00(3.1207E-01)	4.7968E+00(2.2066E-01)	2.7193E+00(2.0743E-01)	2.6372E+00(1.7433E-01)
NSGA-II	WFG9	10	1.8281E+01(1.4350E-01)	9.6028E+00(4.2034E-01)	8.5471E+00(5.3825E-01)	4.4317E+00(2.6263E-01)	4.3547E+00(2.1179E-01)

Tabla 4.9: Resumen de resultados WFG con NSGA-II en IGD+

Algoritmo	Problema	K	0.0	0.25	0.5	0.75	1.0
NSGA-III	WFG1	3	2.5651E+00(3.3162E-03)	1.7393E+00(4.1661E-02)	1.4418E+00(5.0657E-02)	9.9824E-01(4.3217E-02)	7.9781E-01(3.4752E-02)
NSGA-III	WFG1	5	3.0132E+00(4.4126E-03)	2.1151E+00(4.4773E-02)	1.8376E+00(4.1284E-02)	1.4475E+00(2.4165E-02)	1.3697E+00(1.8544E-02)
NSGA-III	WFG1	7	3.3241E+00(7.3818E-03)	2.3655E+00(8.6522E-02)	2.0441E+00(3.2609E-02)	1.0866E+00(9.7425E-02)	9.9030E-01(1.3468E-01)
NSGA-III	WFG1	10	3.5785E+00(2.0214E-02)	2.3346E+00(1.3869E-01)	2.1466E+00(6.6710E-02)	3.6119E-01(1.1763E-01)	3.4648E-01(1.3737E-01)
NSGA-III	WFG2	3	2.7707E+00(6.6230E-01)	5.3183E-01(9.7474E-02)	1.8040E-01(1.7863E-01)	8.3480E-02(9.7369E-02)	6.7318E-02(1.0512E-01)
NSGA-III	WFG2	5	7.3629E+00(1.3511E+00)	1.4999E+00(3.9749E-01)	1.2984E+00(7.2456E-01)	2.2469E-01(1.6432E-01)	1.4850E-01(9.2468E-03)
NSGA-III	WFG2	7	1.0230E+01(2.2566E+00)	2.7013E+00(6.1776E-01)	1.0166E+00(9.1554E-01)	5.4664E-01(6.6241E-01)	3.8939E-01(5.0665E-01)
NSGA-III	WFG2	10	1.4769E+01(3.0893E+00)	4.8041E+00(9.4512E-01)	1.5191E+00(1.3380E+00)	1.2089E+00(1.3296E+00)	8.1144E-01(1.1171E+00)
NSGA-III	WFG3	3	2.6833E+00(4.4791E-01)	9.4427E-01(1.7890E-01)	3.8043E-01(8.3742E-02)	1.0092E-01(1.2085E-02)	8.9849E-02(6.9000E-03)
NSGA-III	WFG3	5	2.9222E+00(4.4048E-01)	1.3503E+00(2.0965E-01)	3.3909E-01(4.5098E-02)	2.7276E-01(2.7712E-02)	2.7297E-01(4.1225E-02)
NSGA-III	WFG3	7	3.3426E+00(3.2071E-01)	1.5454E+00(3.1901E-01)	8.6914E-01(2.7470E-01)	8.5301E-01(2.1614E-01)	9.1217E-01(1.6500E-01)
NSGA-III	WFG3	10	3.3727E+00(5.1651E-01)	1.9557E+00(4.0201E-01)	9.9693E-01(8.5384E-01)	1.4970E+00(8.8789E-01)	1.9174E+00(8.3381E-01)
NSGA-III	WFG4	3	3.8805E+00(1.1055E-01)	9.0336E-01(3.1212E-01)	2.7166E-01(1.7835E-02)	1.3132E-01(8.3219E-03)	3.3199E-02(2.5742E-03)
NSGA-III	WFG4	5	6.5594E+00(1.1621E-01)	2.3324E+00(2.6834E-01)	7.4675E-01(5.2361E-02)	5.4002E-01(2.9418E-02)	3.3549E-01(1.1949E-02)
NSGA-III	WFG4	7	8.8384E+00(1.8590E-01)	3.5507E+00(3.1124E-01)	1.4778E+00(9.4947E-02)	1.1737E+00(7.8060E-02)	7.3848E-01(1.7963E-02)
NSGA-III	WFG4	10	1.3438E+01(5.2346E-01)	6.9154E+00(6.4913E-01)	3.1017E+00(3.6351E-01)	1.9967E+00(1.9670E-01)	8.3465E-01(4.4447E-02)
NSGA-III	WFG5	3	3.2890E+00(3.4051E-01)	1.5337E+00(3.4715E-01)	2.0880E-01(9.5185E-03)	1.2593E-01(3.1441E-03)	9.8951E-02(7.6800E-03)
NSGA-III	WFG5	5	5.9536E+00(5.3304E-01)	2.6776E+00(5.0690E-01)	5.3819E-01(3.1047E-02)	3.8848E-01(1.2437E-02)	2.7019E-01(6.9017E-03)
NSGA-III	WFG5	7	8.6570E+00(6.3787E-01)	3.5706E+00(5.2227E-01)	9.8476E-01(5.1659E-02)	7.8174E-01(2.9257E-02)	6.5399E-01(1.7837E-02)
NSGA-III	WFG5	10	1.1028E+01(1.3196E+00)	4.0270E+00(8.7077E-01)	1.6720E+00(1.0976E-01)	1.3956E+00(4.8206E-02)	1.2058E+00(8.8050E-02)
NSGA-III	WFG6	3	3.1849E+00(3.5244E-01)	1.0324E+00(3.9390E-01)	1.6690E+00(5.6321E-01)	1.4260E-01(6.8752E-03)	7.6298E-02(9.0571E-03)
NSGA-III	WFG6	5	6.3635E+00(5.6749E-01)	2.9987E+00(4.5346E-01)	7.2468E-01(4.3458E-02)	4.7470E-01(1.6686E-02)	2.6128E-01(1.7163E-02)
NSGA-III	WFG6	7	1.0207E+01(5.6036E-01)	4.9292E+00(4.7477E-01)	1.1098E+00(5.1231E-02)	8.0052E-01(2.8959E-02)	4.9010E-01(2.5988E-02)
NSGA-III	WFG6	10	1.6827E+01(1.0924E+00)	9.1831E+00(7.9924E-01)	1.4427E+00(1.1493E-01)	1.1405E+00(8.5842E-02)	6.7957E-01(6.3734E-02)
NSGA-III	WFG7	3	3.7578E+00(2.8314E-01)	9.5448E-01(1.3721E-01)	2.5044E-01(2.5753E-02)	9.4691E-02(4.1695E-03)	2.0722E-02(1.2582E-03)
NSGA-III	WFG7	5	6.3775E+00(4.1762E-01)	2.5931E+00(2.0811E-01)	5.9523E-01(2.6231E-02)	4.0499E-01(1.6902E-02)	2.0127E-01(6.1812E-03)
NSGA-III	WFG7	7	9.8063E+00(6.9371E-01)	5.0710E+00(2.6822E-01)	1.2946E+00(9.7827E-02)	9.2140E-01(5.1178E-02)	4.8722E-01(1.4471E-02)
NSGA-III	WFG7	10	1.5924E+01(1.1678E+00)	9.3861E+00(4.8256E-01)	3.0607E+00(2.0020E-01)	1.7724E+00(1.1758E-01)	8.8690E-01(2.0901E-02)
NSGA-III	WFG8	3	3.9639E+00(1.7629E-01)	1.7502E+00(4.0881E-01)	2.0611E+00(3.3070E-01)	2.4828E-01(1.1321E-02)	1.8425E-01(3.1886E-03)
NSGA-III	WFG8	5	7.4259E+00(1.2091E-01)	3.7711E+00(6.9422E-01)	1.1251E+00(4.5826E-02)	8.5309E-01(1.1315E-01)	3.3884E-01(6.8668E-02)
NSGA-III	WFG8	7	1.1078E+01(7.6632E-01)	5.7595E+00(8.8009E-01)	1.5516E+00(7.7562E-02)	1.0150E+00(5.3721E-02)	5.4594E-01(1.4287E-02)
NSGA-III	WFG8	10	1.7510E+01(2.9643E-01)	1.0225E+01(1.1303E+00)	1.7301E+00(1.4559E-01)	1.3890E+00(5.6890E-02)	8.7749E-01(1.7255E-02)
NSGA-III	WFG9	3	4.1743E+00(1.9044E-01)	7.5797E-01(1.3803E-01)	2.3155E-01(1.6065E-02)	1.5283E-01(3.1253E-02)	1.9730E-01(3.8842E-02)
NSGA-III	WFG9	5	7.9835E+00(2.8389E-01)	2.6014E+00(2.0562E-01)	9.5563E-01(4.2611E-02)	7.5523E-01(1.7622E-02)	5.2160E-01(1.4191E-02)
NSGA-III	WFG9	7	1.1923E+01(1.3888E-01)	4.9629E+00(3.0276E-01)	1.8717E+00(1.5585E-01)	1.3077E+00(1.6884E-01)	9.1350E-01(1.5195E-01)
NSGA-III	WFG9	10	1.7935E+01(3.0478E-01)	9.2131E+00(3.2086E-01)	2.9929E+00(2.2111E-01)	2.0545E+00(1.1753E-01)	1.4839E+00(1.0958E-01)

Tabla 4.10: Resumen de resultados WFG con NSGA-III en IGD+

Los resultados demuestran una preferencia por $H = 1$, seguido de cerca por $H = 0,75$, reforzando la idea de que una alta heterogeneidad favorece la convergencia hacia soluciones óptimas.

Los datos sugieren que, en general, niveles altos de heterogeneidad tienden a favorecer la obtención de mejores resultados en contextos multi-objetivo. No obstante, es interesante observar que, similar a los hallazgos en el análisis mono-objetivo, existen casos donde niveles más bajos de heterogeneidad también conducen a resultados óptimos. Esta variabilidad subraya la importancia de considerar el nivel adecuado de heterogeneidad para cada problema específico, en función de sus características y la dimensionalidad del espacio de búsqueda.

4.2.2. Gráficos de convergencia

Los gráficos de convergencia mostrados en las Figuras 4.10 y 4.11 ilustran cómo los niveles de heterogeneidad afectan la convergencia de los algoritmos.

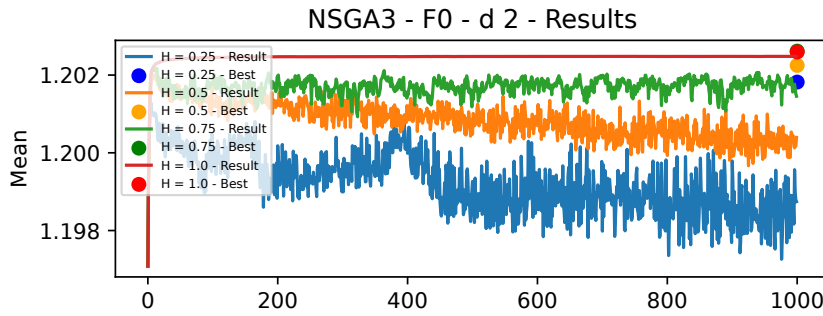


Figura 4.10: NSGA-III - Hipervolumen - Dos dimensiones

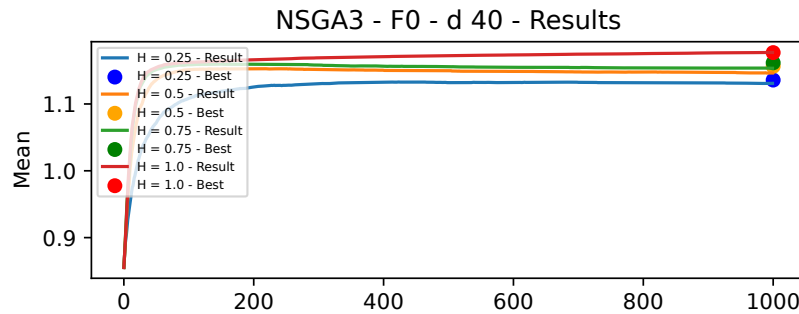


Figura 4.11: NSGA-III - Hipervolumen - Cuarenta dimensiones

Estos patrones se confirman en los gráficos de cajas de las Figuras 4.12 y 4.13, que examinan la distribución de los resultados y su variabilidad en función de los niveles de heterogeneidad.

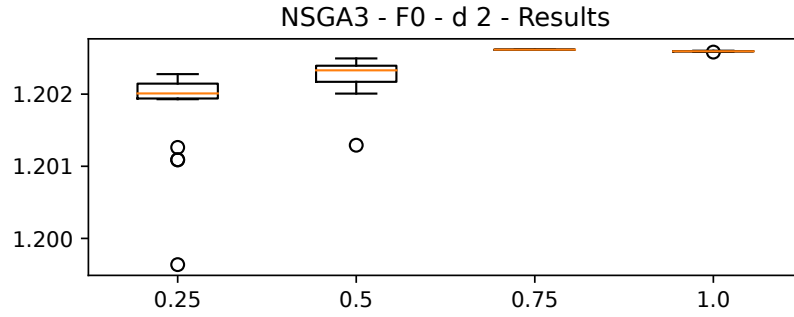


Figura 4.12: NSGA-III - Boxplots - Dos dimensiones

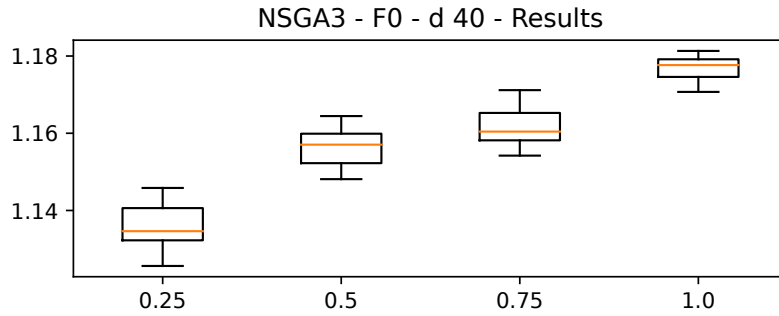


Figura 4.13: NSGA-III - Boxplots - Cuarenta dimensiones

4.2.3. Pruebas estadísticas

A continuación, se presentan los análisis aplicados para evaluar el impacto de la heterogeneidad temporal en los algoritmos estudiados.

Análisis de Kruskal-Wallis

La prueba de Kruskal-Wallis se utilizó para evaluar la significancia estadística de los datos obtenidos en los resultados para mostrar los efectos de la heterogeneidad. Los datos obtenidos, presentados en las Tablas 4.11 y 4.12, indican que todos los valores de p-valor superan el umbral crítico de 0.05, lo que confirma la relevancia estadística de nuestros resultados en las pruebas BiBBOB.

Algoritmo	Problema	Dimensiones	Valor - P
NSGA-II	1	2	1.161788E-19
NSGA-II	1	40	1.161788E-19
NSGA-II	2	2	1.161788E-19
NSGA-II	2	40	1.161788E-19
NSGA-II	3	2	1.161788E-19
NSGA-II	3	40	1.161788E-19
NSGA-II	4	2	1.161788E-19
NSGA-II	4	40	1.161788E-19
NSGA-II	5	2	1.161788E-19
NSGA-II	5	40	1.161788E-19
NSGA-II	6	2	1.161788E-19
NSGA-II	6	40	1.161788E-19
NSGA-II	7	2	1.161788E-19
NSGA-II	7	40	1.161788E-19
NSGA-II	8	2	1.161788E-19
NSGA-II	8	40	1.161788E-19
NSGA-II	9	2	1.161788E-19
NSGA-II	9	40	1.161788E-19
NSGA-II	10	2	1.161788E-19
NSGA-II	10	40	1.161788E-19

Tabla 4.11: Resumen de la prueba BiBBOB de Kruskal Wallis con NSGA-II en hipervolumen

Algoritmo	Problema	Dimensiones	Valor - P
NSGA-III	1	2	3.524316E-19
NSGA-III	1	40	3.524316E-19
NSGA-III	2	2	3.524316E-19
NSGA-III	2	40	3.524316E-19
NSGA-III	3	2	3.524316E-19
NSGA-III	3	40	3.524316E-19
NSGA-III	4	2	3.524316E-19
NSGA-III	4	40	3.524316E-19
NSGA-III	5	2	3.524316E-19
NSGA-III	5	40	3.524316E-19
NSGA-III	6	2	3.524316E-19
NSGA-III	6	40	3.524316E-19
NSGA-III	7	2	3.524316E-19
NSGA-III	7	40	3.524316E-19
NSGA-III	8	2	3.524316E-19
NSGA-III	8	40	3.524316E-19
NSGA-III	9	2	3.524316E-19
NSGA-III	9	40	3.524316E-19
NSGA-III	10	2	3.524316E-19
NSGA-III	10	40	3.524316E-19

Tabla 4.12: Resumen de la prueba BiBBOB de Kruskal Wallis con NSGA-III en hipervolumen

Mostrando de igual forma los mismos resultados en las Tablas 4.13 y 4.14 para los problemas de WFG tomando en cuenta de la limitante del calculo del hipervolumen para mas de cinco dimensiones.

Algoritmo	Problema	K	HV	IGD+
NSGA-II	WFG1	3	2.868101E-19	6.290090E-19
NSGA-II	WFG1	5	1.373986E-18	2.219591E-18
NSGA-II	WFG1	7	-	2.421277E-18
NSGA-II	WFG1	10	-	6.249993E-19
NSGA-II	WFG2	3	1.889840E-18	2.752182E-18
NSGA-II	WFG2	5	3.269007E-18	4.998850E-18
NSGA-II	WFG2	7	-	2.115462E-18
NSGA-II	WFG2	10	-	2.465708E-18
NSGA-II	WFG3	3	1.173661E-18	1.255335E-18
NSGA-II	WFG3	5	2.804691E-18	6.129765E-18
NSGA-II	WFG3	7	-	5.151214E-18
NSGA-II	WFG3	10	-	2.519298E-16
NSGA-II	WFG4	3	2.445484E-19	1.869243E-19
NSGA-II	WFG4	5	2.547132E-18	4.338122E-18
NSGA-II	WFG4	7	-	9.551041E-19
NSGA-II	WFG4	10	-	9.959799E-19
NSGA-II	WFG5	3	1.072532E-17	4.045596E-18
NSGA-II	WFG5	5	1.272764E-19	1.429720E-19
NSGA-II	WFG5	7	-	7.555988E-19
NSGA-II	WFG5	10	-	1.104602E-18
NSGA-II	WFG6	3	2.688698E-18	1.835627E-18
NSGA-II	WFG6	5	2.795256E-18	5.327367E-18
NSGA-II	WFG6	7	-	8.424612E-19
NSGA-II	WFG6	10	-	1.107109E-18
NSGA-II	WFG7	3	1.025247E-18	1.276894E-18
NSGA-II	WFG7	5	9.569499E-19	1.511281E-18
NSGA-II	WFG7	7	-	3.592501E-18
NSGA-II	WFG7	10	-	5.302982E-18
NSGA-II	WFG8	3	4.452856E-19	6.065023E-19
NSGA-II	WFG8	5	8.900907E-19	6.015865E-18
NSGA-II	WFG8	7	-	9.364689E-19
NSGA-II	WFG8	10	-	8.879647E-19
NSGA-II	WFG9	3	1.835296E-18	1.891716E-18
NSGA-II	WFG9	5	9.141590E-19	1.705343E-18
NSGA-II	WFG9	7	-	2.377231E-18
NSGA-II	WFG9	10	-	1.828495E-18

Tabla 4.13: Resumen de la prueba WFG de Kruskal Wallis con NSGAII

Algoritmo	Problema	K	HV	IGD+
NSGA-III	WFG1	3	1.112309E-19	1.112309E-19
NSGA-III	WFG1	5	1.458365E-19	1.112309E-19
NSGA-III	WFG1	7	-	7.185443E-19
NSGA-III	WFG1	10	-	2.556141E-18
NSGA-III	WFG2	3	5.096463E-18	5.450033E-18
NSGA-III	WFG2	5	1.424388E-18	1.219169E-18
NSGA-III	WFG2	7	-	6.427244E-17
NSGA-III	WFG2	10	-	4.158801E-16
NSGA-III	WFG3	3	5.420651E-19	5.402717E-19
NSGA-III	WFG3	5	4.192228E-18	1.175711E-17
NSGA-III	WFG3	7	-	6.836976E-15
NSGA-III	WFG3	10	-	2.639723E-11
NSGA-III	WFG4	3	1.112309E-19	1.112309E-19
NSGA-III	WFG4	5	1.112309E-19	1.112309E-19
NSGA-III	WFG4	7	-	1.192118E-19
NSGA-III	WFG4	10	-	1.112309E-19
NSGA-III	WFG5	3	1.112309E-19	1.112309E-19
NSGA-III	WFG5	5	1.112309E-19	1.112309E-19
NSGA-III	WFG5	7	-	1.112309E-19
NSGA-III	WFG5	10	-	1.138433E-19
NSGA-III	WFG6	3	5.707854E-19	4.719595E-19
NSGA-III	WFG6	5	1.112309E-19	1.112309E-19
NSGA-III	WFG6	7	-	1.112309E-19
NSGA-III	WFG6	10	-	1.305375E-19
NSGA-III	WFG7	3	1.112309E-19	1.112309E-19
NSGA-III	WFG7	5	1.112309E-19	1.112309E-19
NSGA-III	WFG7	7	-	1.112309E-19
NSGA-III	WFG7	10	-	1.112309E-19
NSGA-III	WFG8	3	9.671707E-19	7.036632E-19
NSGA-III	WFG8	5	1.112309E-19	1.112309E-19
NSGA-III	WFG8	7	-	1.112309E-19
NSGA-III	WFG8	10	-	1.165034E-19
NSGA-III	WFG9	3	5.325995E-19	6.196537E-19
NSGA-III	WFG9	5	1.112309E-19	1.112309E-19
NSGA-III	WFG9	7	-	1.566267E-19
NSGA-III	WFG9	10	-	1.112309E-19

Tabla 4.14: Resumen de la prueba WFG de Kruskal Wallis NSGAIII

Prueba de Wilcoxon ranksum

Posteriormente, la prueba de Wilcoxon ranksum fue aplicada, cuyos resultados se detallan en la Tabla Wilcoxon BiBBOB. Esta prueba subraya que, aunque el elitismo es

predominante en varios escenarios, los resultados más significativos se sitúan entre los valores de 0.75 y 1.00, sugiriendo que el elitismo puro no es siempre la estrategia más efectiva para abordar problemas bi-objetivo.

Algoritmo	Problema	Dimensiones	Heterogeneidad	h=0.0	h=0.25	h=0.5	h=0.75	h=1.0
NSGA2	1	2	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA2	1	2	0.25	3.150924e-08	-	1.000000e+00	1.000000e+00	1.000000e+00
NSGA2	1	2	0.5	3.150924e-08	4.256164e-08	-	1.000000e+00	1.000000e+00
NSGA2	1	2	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	1.000000e+00
NSGA2	1	2	1.0	3.150924e-08	3.150924e-08	3.150924e-08	3.150924e-08	-
NSGA2	2	2	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA2	2	2	0.25	3.150924e-08	-	1.000000e+00	1.000000e+00	1.000000e+00
NSGA2	2	2	0.5	3.150924e-08	4.256164e-08	-	1.000000e+00	1.000000e+00
NSGA2	2	2	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	1.000000e+00
NSGA2	2	2	1.0	3.150924e-08	3.150924e-08	3.150924e-08	3.150924e-08	-
NSGA2	3	2	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA2	3	2	0.25	3.150924e-08	-	1.000000e+00	1.000000e+00	1.000000e+00
NSGA2	3	2	0.5	3.150924e-08	4.256164e-08	-	1.000000e+00	1.000000e+00
NSGA2	3	2	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	1.000000e+00
NSGA2	3	2	1.0	3.150924e-08	3.150924e-08	3.150924e-08	3.150924e-08	-
NSGA2	4	2	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA2	4	2	0.25	3.150924e-08	-	1.000000e+00	1.000000e+00	1.000000e+00
NSGA2	4	2	0.5	3.150924e-08	4.256164e-08	-	1.000000e+00	1.000000e+00
NSGA2	4	2	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	1.000000e+00
NSGA2	5	2	1.0	3.150924e-08	3.150924e-08	3.150924e-08	3.150924e-08	-
NSGA2	5	2	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA2	5	2	0.25	3.150924e-08	-	1.000000e+00	1.000000e+00	1.000000e+00
NSGA2	5	2	0.5	3.150924e-08	4.256164e-08	-	1.000000e+00	1.000000e+00
NSGA2	5	2	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	1.000000e+00
NSGA2	5	2	1.0	3.150924e-08	3.150924e-08	3.150924e-08	3.150924e-08	-
NSGA2	6	2	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA2	6	2	0.25	3.150924e-08	-	1.000000e+00	1.000000e+00	1.000000e+00
NSGA2	6	2	0.5	3.150924e-08	4.256164e-08	-	1.000000e+00	1.000000e+00
NSGA2	6	2	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	1.000000e+00
NSGA2	6	2	1.0	3.150924e-08	3.150924e-08	3.150924e-08	3.150924e-08	-
NSGA2	7	2	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA2	7	2	0.25	3.150924e-08	-	1.000000e+00	1.000000e+00	1.000000e+00
NSGA2	7	2	0.5	3.150924e-08	4.256164e-08	-	1.000000e+00	1.000000e+00
NSGA2	7	2	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	1.000000e+00
NSGA2	7	2	1.0	3.150924e-08	3.150924e-08	3.150924e-08	3.150924e-08	-
NSGA2	8	2	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA2	8	2	0.25	3.150924e-08	-	1.000000e+00	1.000000e+00	1.000000e+00
NSGA2	8	2	0.5	3.150924e-08	4.256164e-08	-	1.000000e+00	1.000000e+00
NSGA2	8	2	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	1.000000e+00
NSGA2	8	2	1.0	3.150924e-08	3.150924e-08	3.150924e-08	3.150924e-08	-
NSGA2	9	2	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA2	9	2	0.25	3.150924e-08	-	1.000000e+00	1.000000e+00	1.000000e+00
NSGA2	9	2	0.5	3.150924e-08	4.256164e-08	-	1.000000e+00	1.000000e+00
NSGA2	9	2	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	1.000000e+00
NSGA2	9	2	1.0	3.150924e-08	3.150924e-08	3.150924e-08	3.150924e-08	-
NSGA2	10	2	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA2	10	2	0.25	3.150924e-08	-	1.000000e+00	1.000000e+00	1.000000e+00
NSGA2	10	2	0.5	3.150924e-08	4.256164e-08	-	1.000000e+00	1.000000e+00
NSGA2	10	2	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	1.000000e+00
NSGA2	10	2	1.0	3.150924e-08	3.150924e-08	3.150924e-08	3.150924e-08	-

Tabla 4.15: Resumen de la prueba BiBBOB de Wilcoxon NSGAII en 2 dimensiones

[illegible]

Tabla 4.16: Resumen de la prueba BiBBOB de Wilcoxon NSGAI en 40 dimensiones

Algoritmo	Problema	Dimensiones	Heterogeneidad	h=0.0	h=0.25	h=0.5	h=0.75	h=1.0
NSGA3	1	2	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA3	1	2	0.25	3.150924e-08	-	9.99388e-01	1.000000e+00	1.000000e+00
NSGA3	1	2	0.5	3.150924e-08	6.123895e-05	-	1.000000e+00	1.000000e+00
NSGA3	1	2	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	3.150924e-08
NSGA3	1	2	1.0	3.150924e-08	3.150924e-08	3.150924e-08	1.000000e+00	-
NSGA3	2	2	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA3	2	2	0.25	3.150924e-08	-	9.99388e-01	1.000000e+00	1.000000e+00
NSGA3	2	2	0.5	3.150924e-08	6.123895e-05	-	1.000000e+00	1.000000e+00
NSGA3	2	2	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	3.150924e-08
NSGA3	2	2	1.0	3.150924e-08	3.150924e-08	3.150924e-08	1.000000e+00	-
NSGA3	3	2	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA3	3	2	0.25	3.150924e-08	-	9.99388e-01	1.000000e+00	1.000000e+00
NSGA3	3	2	0.5	3.150924e-08	6.123895e-05	-	1.000000e+00	1.000000e+00
NSGA3	3	2	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	3.150924e-08
NSGA3	3	2	1.0	3.150924e-08	3.150924e-08	3.150924e-08	1.000000e+00	-
NSGA3	4	2	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA3	4	2	0.25	3.150924e-08	-	9.99388e-01	1.000000e+00	1.000000e+00
NSGA3	4	2	0.5	3.150924e-08	6.123895e-05	-	1.000000e+00	1.000000e+00
NSGA3	4	2	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	3.150924e-08
NSGA3	4	2	1.0	3.150924e-08	3.150924e-08	3.150924e-08	1.000000e+00	-
NSGA3	5	2	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA3	5	2	0.25	3.150924e-08	-	9.99388e-01	1.000000e+00	1.000000e+00
NSGA3	5	2	0.5	3.150924e-08	6.123895e-05	-	1.000000e+00	1.000000e+00
NSGA3	5	2	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	3.150924e-08
NSGA3	5	2	1.0	3.150924e-08	3.150924e-08	3.150924e-08	1.000000e+00	-
NSGA3	6	2	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA3	6	2	0.25	3.150924e-08	-	9.99388e-01	1.000000e+00	1.000000e+00
NSGA3	6	2	0.5	3.150924e-08	6.123895e-05	-	1.000000e+00	1.000000e+00
NSGA3	6	2	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	3.150924e-08
NSGA3	6	2	1.0	3.150924e-08	3.150924e-08	3.150924e-08	1.000000e+00	-
NSGA3	7	2	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA3	7	2	0.25	3.150924e-08	-	9.99388e-01	1.000000e+00	1.000000e+00
NSGA3	7	2	0.5	3.150924e-08	6.123895e-05	-	1.000000e+00	1.000000e+00
NSGA3	7	2	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	3.150924e-08
NSGA3	7	2	1.0	3.150924e-08	3.150924e-08	3.150924e-08	1.000000e+00	-
NSGA3	8	2	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA3	8	2	0.25	3.150924e-08	-	9.99388e-01	1.000000e+00	1.000000e+00
NSGA3	8	2	0.5	3.150924e-08	6.123895e-05	-	1.000000e+00	1.000000e+00
NSGA3	8	2	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	3.150924e-08
NSGA3	8	2	1.0	3.150924e-08	3.150924e-08	3.150924e-08	1.000000e+00	-
NSGA3	9	2	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA3	9	2	0.25	3.150924e-08	-	9.99388e-01	1.000000e+00	1.000000e+00
NSGA3	9	2	0.5	3.150924e-08	6.123895e-05	-	1.000000e+00	1.000000e+00
NSGA3	9	2	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	3.150924e-08
NSGA3	9	2	1.0	3.150924e-08	3.150924e-08	3.150924e-08	1.000000e+00	-
NSGA3	10	2	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA3	10	2	0.25	3.150924e-08	-	9.99388e-01	1.000000e+00	1.000000e+00
NSGA3	10	2	0.5	3.150924e-08	6.123895e-05	-	1.000000e+00	1.000000e+00
NSGA3	10	2	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	3.150924e-08
NSGA3	10	2	1.0	3.150924e-08	3.150924e-08	3.150924e-08	1.000000e+00	-

Tabla 4.17: Resumen de la prueba BiBBOB de Wilcoxon NSGAIII en 2 dimensiones

Algoritmo	Problema	Dimensiones	Heterogeneidad	h=0.0	h=0.25	h=0.5	h=0.75	h=1.0
NSGA3	1	40	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA3	1	40	0.25	3.150924e-08	-	9.99388e-01	1.000000e+00	1.000000e+00
NSGA3	1	40	0.5	3.150924e-08	6.123895e-05	-	1.000000e+00	1.000000e+00
NSGA3	1	40	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	3.150924e-08
NSGA3	1	40	1.0	3.150924e-08	3.150924e-08	3.150924e-08	1.000000e+00	-
NSGA3	2	40	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA3	2	40	0.25	3.150924e-08	-	9.99388e-01	1.000000e+00	1.000000e+00
NSGA3	2	40	0.5	3.150924e-08	6.123895e-05	-	1.000000e+00	1.000000e+00
NSGA3	2	40	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	3.150924e-08
NSGA3	2	40	1.0	3.150924e-08	3.150924e-08	3.150924e-08	1.000000e+00	-
NSGA3	3	40	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA3	3	40	0.25	3.150924e-08	-	9.99388e-01	1.000000e+00	1.000000e+00
NSGA3	3	40	0.5	3.150924e-08	6.123895e-05	-	1.000000e+00	1.000000e+00
NSGA3	3	40	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	3.150924e-08
NSGA3	3	40	1.0	3.150924e-08	3.150924e-08	3.150924e-08	1.000000e+00	-
NSGA3	4	40	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA3	4	40	0.25	3.150924e-08	-	9.99388e-01	1.000000e+00	1.000000e+00
NSGA3	4	40	0.5	3.150924e-08	6.123895e-05	-	1.000000e+00	1.000000e+00
NSGA3	4	40	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	3.150924e-08
NSGA3	4	40	1.0	3.150924e-08	3.150924e-08	3.150924e-08	1.000000e+00	-
NSGA3	5	40	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA3	5	40	0.25	3.150924e-08	-	9.99388e-01	1.000000e+00	1.000000e+00
NSGA3	5	40	0.5	3.150924e-08	6.123895e-05	-	1.000000e+00	1.000000e+00
NSGA3	5	40	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	3.150924e-08
NSGA3	5	40	1.0	3.150924e-08	3.150924e-08	3.150924e-08	1.000000e+00	-
NSGA3	6	40	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA3	6	40	0.25	3.150924e-08	-	9.99388e-01	1.000000e+00	1.000000e+00
NSGA3	6	40	0.5	3.150924e-08	6.123895e-05	-	1.000000e+00	1.000000e+00
NSGA3	6	40	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	3.150924e-08
NSGA3	6	40	1.0	3.150924e-08	3.150924e-08	3.150924e-08	1.000000e+00	-
NSGA3	7	40	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA3	7	40	0.25	3.150924e-08	-	9.99388e-01	1.000000e+00	1.000000e+00
NSGA3	7	40	0.5	3.150924e-08	6.123895e-05	-	1.000000e+00	1.000000e+00
NSGA3	7	40	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	3.150924e-08
NSGA3	7	40	1.0	3.150924e-08	3.150924e-08	3.150924e-08	1.000000e+00	-
NSGA3	8	40	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA3	8	40	0.25	3.150924e-08	-	9.99388e-01	1.000000e+00	1.000000e+00
NSGA3	8	40	0.5	3.150924e-08	6.123895e-05	-	1.000000e+00	1.000000e+00
NSGA3	8	40	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	3.150924e-08
NSGA3	8	40	1.0	3.150924e-08	3.150924e-08	3.150924e-08	1.000000e+00	-
NSGA3	9	40	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA3	9	40	0.25	3.150924e-08	-	9.99388e-01	1.000000e+00	1.000000e+00
NSGA3	9	40	0.5	3.150924e-08	6.123895e-05	-	1.000000e+00	1.000000e+00
NSGA3	9	40	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	3.150924e-08
NSGA3	9	40	1.0	3.150924e-08	3.150924e-08	3.150924e-08	1.000000e+00	-
NSGA3	10	40	0.0	-	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00
NSGA3	10	40	0.25	3.150924e-08	-	9.99388e-01	1.000000e+00	1.000000e+00
NSGA3	10	40	0.5	3.150924e-08	6.123895e-05	-	1.000000e+00	1.000000e+00
NSGA3	10	40	0.75	3.150924e-08	3.150924e-08	3.150924e-08	-	3.150924e-08
NSGA3	10	40	1.0	3.150924e-08	3.150924e-08	3.150924e-08	1.000000e+00	-

Tabla 4.18: Resumen de la prueba BiBBOB de Wilcoxon NSGAIII en 40 dimensiones

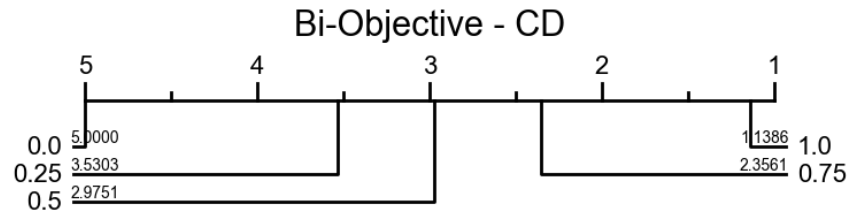


Figura 4.14: Gráfica de diferencias críticas bi-objetivo

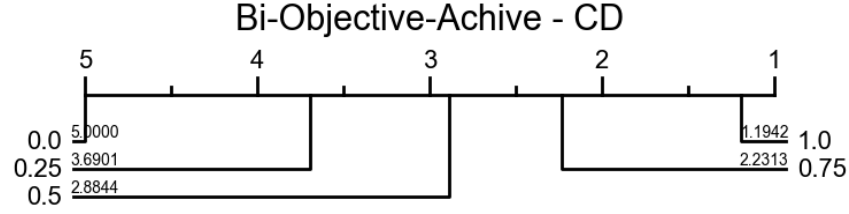


Figura 4.15: Gráfica de diferencias críticas bi-objetivo con archivo externo

Al igual que en las pruebas mono-objetivo, se generaron las gráficas de diferencias críticas para las pruebas multi-objetivo, tanto en las pruebas bi-objetivo sin el uso de un archivo externo, como se muestra en la Figura 4.14, así como con el uso de un archivo externo, como se presenta en la Figura 4.15.

En ambos escenarios, los niveles de heterogeneidad más altos, 0.75 y 1.0 , muestran un rendimiento superior, con diferencias estadísticamente significativas respecto a los niveles más bajos. Los resultados sugieren una clara mejora en el rendimiento al incrementar la heterogeneidad, tanto en los problemas bi-objetivo que hacen uso de un archivo de soluciones como en aquellos que no lo utilizan.

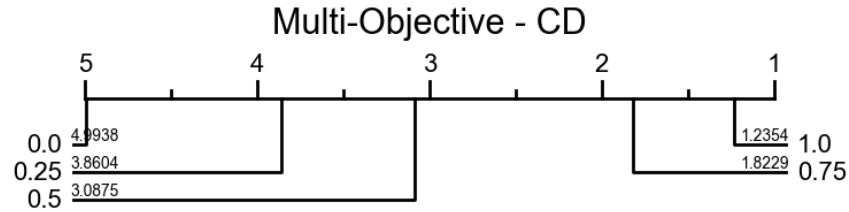


Figura 4.16: Gráfica de diferencias críticas multi-objetivo

Por otro lado, los resultados multi-objetivo presentados en la Figura 4.16 confirman que, al igual que en los análisis previos, un aumento en la heterogeneidad, específicamente con valores 0.75 y 1.0 , conduce a una mejora significativa en el rendimiento de los problemas multi-objetivo. Los valores bajos de heterogeneidad, como 0.0 y 0.25 , exhiben un rendimiento considerablemente inferior, sin diferencias relevantes entre ellos.

Conteo de Borda

El análisis mediante el conteo de Borda, presentado en la Tabla 4.19, ofrece una perspectiva adicional, destacando casos en los cuales valores de heterogeneidad de 0.75 resultan ser particularmente efectivos, especialmente en el algoritmo NSGA-III y en problemas de dimensiones reducidas.

Algoritmo	Dimensiones	$h = 0,0$	$h = 0,25$	$h = 0,5$	$h = 0,75$	$h = 1,0$
NSGA-II	2	55	110	131	218	275
NSGA-II	3	55	110	131	218	275
NSGA-II	5	55	110	131	218	275
NSGA-II	10	55	110	131	218	275
NSGA-II	20	55	110	131	218	275
NSGA-II	40	55	110	131	218	275
NSGA-III	2	55	110	164	199	259
NSGA-III	3	55	110	164	199	259
NSGA-III	5	55	110	164	199	259
NSGA-III	10	55	110	164	199	259
NSGA-III	20	55	110	164	199	259
NSGA-III	40	55	110	164	199	259
Total	-	660	1320	1768	2414	3034

Tabla 4.19: Tabla de resultados del conteo de Borda en BiBBOB

Además de ello mostrando notables resultados por la parte de WFG como se muestran en las Tablas 4.20 y 4.21

Algoritmo	K	$h = 0,0$	$h = 0,25$	$h = 0,5$	$h = 0,75$	$h = 1,0$
NSGA-II	3	9	18	25	36	43
NSGA-II	5	9	18	26	36	44
NSGA-III	3	9	19	25	36	45
NSGA-III	5	8	16	23	32	39
Total	-	35	71	99	140	171

Tabla 4.20: Tabla de resultados del conteo de Borda en WFG con hipervolumen

Algoritmo	K	$h = 0,0$	$h = 0,25$	$h = 0,5$	$h = 0,75$	$h = 1,0$
NSGA-II	3	9	18	24	36	44
NSGA-II	5	9	19	26	37	44
NSGA-II	7	9	19	26	36	39
NSGA-II	10	9	18	26	37	38
NSGA-III	3	9	20	24	36	45
NSGA-III	5	9	18	26	36	44
NSGA-III	7	9	18	27	35	42
NSGA-III	10	8	16	25	30	36
Total	-	71	146	204	283	332

Tabla 4.21: Tabla de resultados del conteo de Borda en WFG con IGD+

En conjunto, estos análisis estadísticos refuerzan la noción de que la heterogeneidad tiene un impacto significativo en la optimización mono-objetivo y multi-objetivo. Además, revelan que una selección cuidadosa del nivel de heterogeneidad puede optimizar el rendimiento de los algoritmos en estos complejos espacios de búsqueda, desafiando la suposición de que una estrategia de elitismo puro es universalmente preferible.

4.2.4. Explicabilidad

Al igual que en los experimentos mono-objetivo, realizamos un análisis exhaustivo de estas características ELA para todos los problemas evaluados.

Estos resultados se muestran en las Figuras 4.17. En ellas se muestran con relevancia las métricas de dispersión y los ángulos como factores que impactan el rendimiento y la eficacia del uso de la heterogeneidad. Mientras que características como `ela_meta.quad_simple.adj_r2` y las métricas asociadas con la escalada de colinas `hill_climbing` se presentan con variabilidad significativa, lo que sugiere su impacto relevante en la salida del modelo.

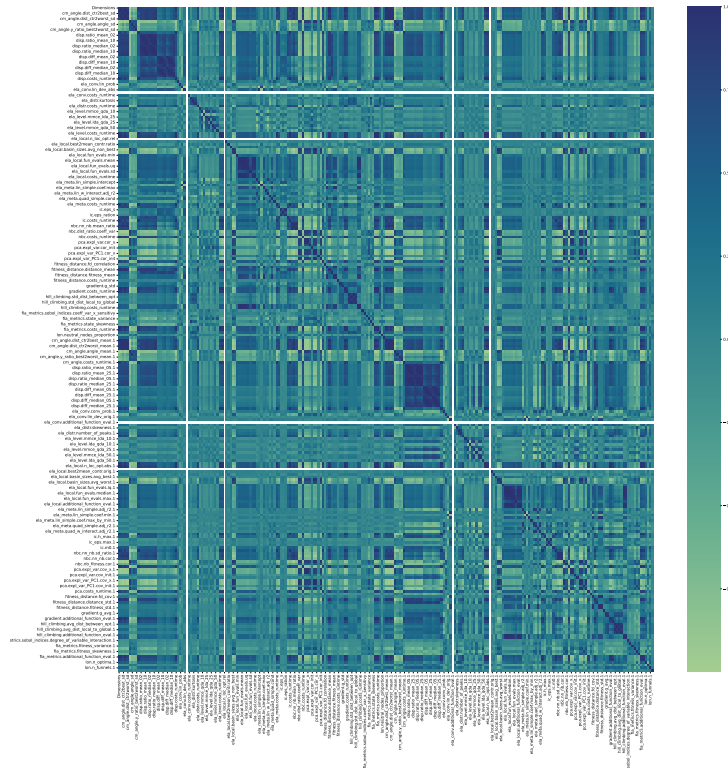


Figura 4.17: Mapa de correlación de caracterisitcas de los espacios Bi-Objetivo

El análisis de SHAP en bi-objetivo que se muestra en la Figura 4.18 confirman que las

características relacionadas con la metaheurística cuadrática ajustada impacta en un rango amplio en la salida del modelo. Esto sugiere que tales características son cruciales para el ajuste del modelo, evidenciando una fuerte dependencia del modelo en estas variables para la generación de resultados precisos.

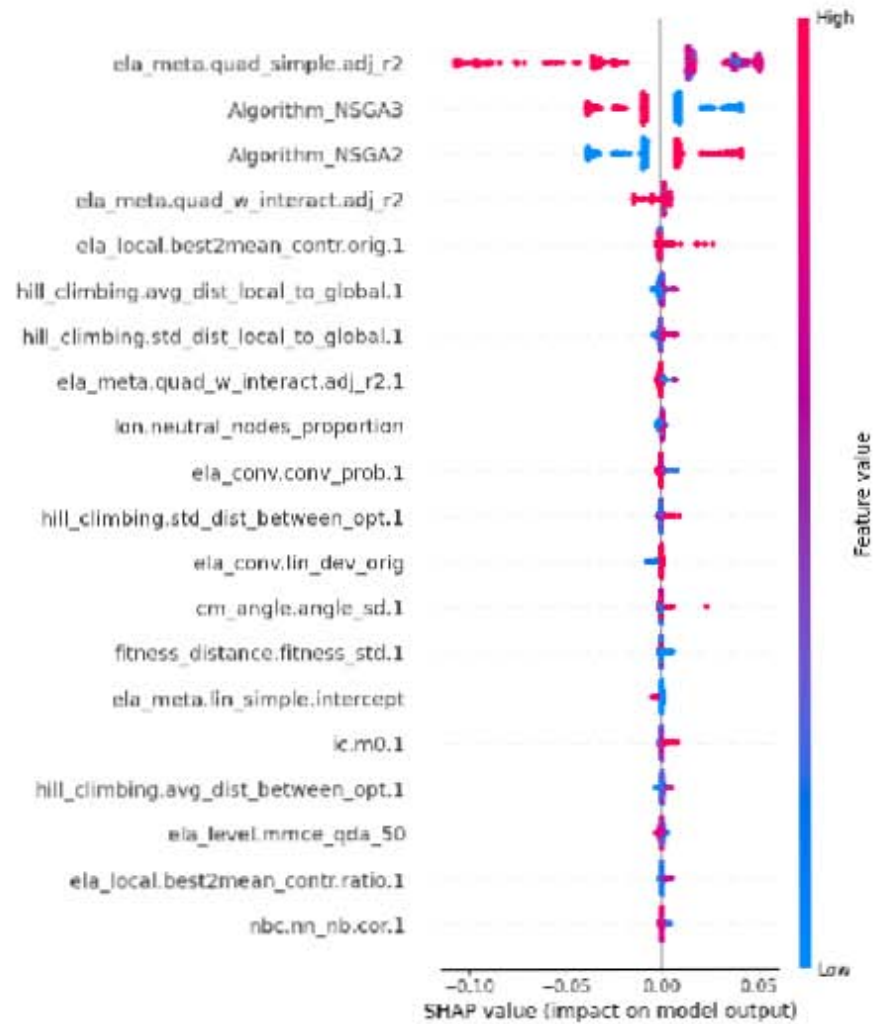


Figura 4.18: Análisis Bi-Objetivo de SHAP destacando las características más influyentes en la elección de heterogeneidad.

Por otro lado, las métricas asociadas a la escalada de colinas, como la distancia promedio y la desviación estándar de la distancia desde local a global, así como el algoritmo aplicado, también mostraron impactos considerables. Estos resultados indican que el

ajuste espacial y la exploración de soluciones locales frente a globales y el su aplicación respectiva en cada algoritmo juegan roles significativos con el uso de la heterogeneidad.

4.3. Discusión

Esta investigación se centró inicialmente en evaluar si la heterogeneidad temporal puede influir de manera significativa en el balance entre la exploración y la explotación. Los resultados obtenidos confirman que la heterogeneidad temporal contribuye a este balance, particularmente se denota más en entornos de baja dimensionalidad y en problemas con un solo objetivo.

Como se denotó en el análisis, se observó un aumento consistente en la efectividad de los algoritmos conforme aumentaba la heterogeneidad, lo que indica una mejor adaptación en escenarios más complejos. Este fenómeno puede interpretarse como un incremento en la capacidad de explotación sin sacrificar la eficacia en la exploración, especialmente en configuraciones donde los desafíos del problema se intensifican.

Sin embargo, es relevante destacar que, a pesar de la tendencia general hacia una mejor adaptabilidad en condiciones heterogéneas, los algoritmos evolutivos como el Algoritmo Genético (GA) mostraron preferencias hacia estrategias elitistas en situaciones de alta complejidad y dimensionalidad al igual que en los casos para los algoritmos en múltiples objetivos. Este enfoque elitista puede ser beneficioso para preservar individuos de alto rendimiento, pero también puede limitar la diversidad genética necesaria para una exploración efectiva a largo plazo.

Además, se encontraron evidencias de que en ciertos casos, especialmente en problemas gestionados por el algoritmo de Evolución Diferencial (DE), el elitismo no siempre produce los mejores resultados. Aportando a la importancia de adaptar la estrategia de balance entre exploración y explotación según las características específicas del problema y la heterogeneidad del entorno. Los resultados indican que la heterogeneidad puede ser un factor determinante para mitigar los efectos limitantes del elitismo, promoviendo un enfoque más equilibrado que pueda adaptarse eficazmente a la variabilidad del espacio de búsqueda.

Capítulo 5

Conclusiones y trabajo futuro

En este estudio exhaustivo sobre la influencia de la heterogeneidad en algoritmos evolutivos, se han obtenido resultados significativos que arrojan luz sobre su impacto en problemas mono-objetivo y multi-objetivo. Dentro de este capítulo, resumiremos los principales hallazgos del mismo.

5.1. Resultados en problemas mono-objetivo

- La heterogeneidad juega un papel relevante en la optimización mono-objetivo, con resultados interesantes en la búsqueda de soluciones con buena calidad.
- Se ha observado que, en problemas de baja dimensionalidad, valores moderados de heterogeneidad, alrededor del 0.5, tienden a ofrecer los mejores resultados. A medida que la dimensionalidad aumenta, los valores de heterogeneidad entre el 0.75 y 1.0 se vuelven más efectivos.
- Las gráficas de convergencia respaldan estos hallazgos al mostrar cómo los valores de heterogeneidad influyen en la velocidad de convergencia de los algoritmos.
- Los análisis estadísticos, incluyendo Kruskal-Wallis y Wilcoxon Ranksum, confirman la relevancia de estos resultados, con una clara tendencia hacia la importancia de la heterogeneidad en la optimización mono-objetivo.

5.2. Resultados en problemas multi-objetivo

- En el contexto bi-objetivo, la alta heterogeneidad generalmente conduce a mejores resultados, pero no se puede descartar la efectividad de la baja heterogeneidad como lo son en las pruebas con NSGA-III, mientras que en el contexto multi-objetivo, encontramos resultados favorables para altas dimensiones como se puede ver en WFG3 para NSGA-III con siete y diez dimensiones.
- Las gráficas de convergencia muestran que, en muchos casos, los valores de heterogeneidad más altos resultan en una convergencia más rápida hacia el frente de

Pareto. Sin embargo, existen excepciones notables como lo es el experimento de WFG3 con NSGA-III en diez dimensiones.

- Los análisis estadísticos, tanto Kruskal-Wallis como Wilcoxon Ranksum, respaldan la relevancia estadística de estos hallazgos, subrayando la influencia de la heterogeneidad en la optimización bi-objetivo.

5.3. Explicabilidad de los resultados

La interpretación de los resultados obtenidos mediante el gráfico SHAP revela patrones distintivos en la influencia de las características sobre las predicciones del modelo. En particular, la característica denotada como *Algorithm_GA* muestra una agrupación densa de valores SHAP cercanos a cero en la parte superior del gráfico, lo que indica una influencia estable y homogénea en las predicciones. Este hallazgo es notable ya que, a pesar de ser la característica de mayor importancia, su efecto sobre el modelo es consistente a través de las observaciones, sugiriendo que el Algoritmo Genético (GA) ejerce un efecto uniforme en las predicciones.

Por su parte, *Algorithm_DE* (Algoritmo de Evolución Diferencial) presenta una influencia predominantemente negativa, como se evidencia por la concentración de puntos hacia el lado izquierdo del eje cero en el gráfico SHAP, indicando que generalmente disminuye la probabilidad de las predicciones del modelo.

En contraste, características como *ela_level.costs_runtime* y *Algorithm_ES* (Algoritmo de Estrategia Evolutiva) exhiben una distribución de valores SHAP tanto positivos como negativos, aunque con una variabilidad menos marcada que las características anteriormente mencionadas.

Otras características, como *fitness_distance.distance_mean* y *ela_conv.conv_prob*, destacan por su notable dispersión de valores SHAP, señalando que el impacto de estas sobre el modelo varía significativamente entre las observaciones.

Las características situadas en la mitad inferior del gráfico, incluyendo:

- *ela_level*: *lda_qda_50*,
- *ela_local*: *best2mean_contr.orig*,
- *lon_n_optima*.

Se caracterizan por valores SHAP mayoritariamente agrupados cerca de cero, sugiriendo una contribución menor y más uniforme a las predicciones del modelo.

cm_angle: *angle_mean* se identifica como la característica de menor importancia relativa, ubicándose al final de la lista de características evaluadas.

La coloración de los puntos en el gráfico, ilustra cómo algunas características, tales como *Algorithm_DE* y *fitness_distance: distance_mean*, están asociadas con un amplio rango de valores y cambios significativos en la salida del modelo. En contraposición, *Algorithm_GA* muestra un rango más estrecho de valores de características, subrayando su efecto robusto en diversos escenarios.

Este análisis proporciona una jerarquía detallada de la influencia y la variabilidad en la contribución de cada característica a la salida del modelo. Se reveló que características específicas, como *ela_meta: quad_simple.adj_r2*, *Algorithm_NSGA2*, y *ela_meta: quad_w_interact.adj_r2*, tienen una preponderancia significativa, con una dispersión de sus valores SHAP que indica una variabilidad considerable en su contribución, sugiriendo un impacto marcado en las predicciones finales.

Contrastantemente, características situadas en el centro del espectro, como *ela_conv: conv_prob.1* y *hill_climbing: std_dist_between_opt.1*, ejercieron una influencia moderada, con una menor variabilidad en sus valores SHAP, apuntando a un efecto más homogéneo en las observaciones.

Las características con una posición más baja en el gráfico, incluidas:

- *ela_local: best2mean_contr.ratio.1*,
- *nbc: nn_nb.cor.1*.

Se caracterizaron por una importancia marginal, con valores SHAP agrupados cerca de cero que reflejan un impacto reducido en la salida del modelo.

La distribución de los valores SHAP subraya una relación compleja entre el valor de las características y su influencia en las predicciones del modelo. Este patrón, inferido a partir del color asignado a cada punto, sugiere cómo ciertas características aumentan o disminuyen la probabilidad de la predicción del modelo conforme varían sus valores.

En conclusión, este estudio subraya la heterogeneidad como un elemento crítico en la optimización evolutiva, cuyo efecto fluctúa según el tipo de problema y la dimensionalidad. Los hallazgos, apoyados por análisis estadísticos y exploración de las características de los espacios de búsqueda, proporcionan una base sólida para la selección de estrategias de heterogeneidad óptimas en distintos contextos de optimización.

5.4. Trabajo futuro

Estos hallazgos son valiosos tanto para investigadores como para profesionales que trabajan en la optimización de problemas, ya que generan un gran precedente en el estado del arte para investigar más a fondo en dos puntos claves:

- Generación de mecanismos de ajuste en línea para el parámetro de heterogeneidad para generar una mayor adaptabilidad en variaciones de problemas.

- Aplicación de técnicas de aprendizaje por refuerzo para el ajuste en línea [características - valor de heterogeneidad]
- Exploración exhaustiva de las características de la población con el archivo externo para un ajuste fino
- Pruebas en diferentes algoritmos de optimización para multi-objetivo.
 - MOEA/D: Descomposición
 - SMS-EMOA: Basado en Indicadores
- Exploración más exhaustiva para un detallado más profundo en características de mayor impacto para los valores de heterogeneidad.
- Pruebas exhaustivas en problemas que asemejen muestras del mundo real.

Estos aspectos destacan la necesidad de continuar explorando la heterogeneidad dentro de los algoritmos evolutivos, no solo para mejorar el entendimiento teórico y práctico de estos sistemas, sino también para desarrollar herramientas más robustas y eficaces que puedan ser aplicadas a una gama más amplia de problemas de optimización. La adaptabilidad y la capacidad de realizar pruebas en escenarios que reflejen desafíos del mundo real serán cruciales para el avance de esta área de investigación.

Bibliografía

- [1] Ajith Abraham, Lakhmi C. Jain y Robert Goldberg. *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications (Advanced Information and Knowledge Processing)*. Berlin, Heidelberg: Springer-Verlag, 2005. ISBN: 1852337877.
- [2] Hans-Georg Beyer y Hans-Paul Schwefel. «Evolution strategies - A comprehensive introduction». En: *Natural Computing* 1 (mar. de 2002), págs. 3-52. DOI: 10.1023/A:1015059928466.
- [3] J. Blank y K. Deb. «pymoo: Multi-Objective Optimization in Python». En: *IEEE Access* 8 (2020), págs. 89497-89509.
- [4] Julian Blank et al. «Generating Well-Spaced Points on a Unit Simplex for Evolutionary Many-Objective Optimization». En: *IEEE Transactions on Evolutionary Computation* 25.1 (2021), págs. 48-60. DOI: 10.1109/TEVC.2020.2992387.
- [5] Dimo Brockhoff et al. «Using Well-Understood Single-Objective Functions in Multiobjective Black-Box Optimization Test Suites». En: *Evolutionary Computation* 30.2 (2022). ArXiv e-prints, arXiv:1604.00359, págs. 165-193. DOI: 10.1162/evco_a_00298. URL: <https://inria.hal.science/hal-01296987>.
- [6] Carlos A. Coello Coello, Gary B. Lamont y David A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, 2007.
- [7] Pedro A. F. Cruz y Delfim F. M. Torres. «Evolution strategies in optimization problems». En: *Proc. Estonian Acad. Sci. Phys. Math.* 56.4 (2007), págs. 299-309.
- [8] G. Cuccu, F. Gomez y T. Glasmachers. «Novelty-based restarts for evolution strategies». En: *2011 IEEE Congress of Evolutionary Computation (CEC)*. IEEE, 2011, págs. 158-163.
- [9] Debzani Deb y Russell M. Smith. «Application of Random Forest and SHAP Tree Explainer in Exploring Spatial (In)Justice to Aid Urban Planning». En: *ISPRS International Journal of Geo-Information* 10.9 (2021). ISSN: 2220-9964. DOI: 10.3390/ijgi10090629. URL: <https://www.mdpi.com/2220-9964/10/9/629>.
- [10] Kalyanmoy Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, 2001.

- [11] Kalyanmoy Deb e Himanshu Jain. «An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints». En: *IEEE Transactions on Evolutionary Computation* 18.4 (2014), págs. 577-601. DOI: 10.1109/TEVC.2013.2281535.
- [12] Kalyanmoy Deb et al. «A fast and elitist multiobjective genetic algorithm: NSGA-II». En: *IEEE transactions on evolutionary computation* 6.2 (2002), págs. 182-197.
- [13] «Differential Evolution». En: *Differential Evolution: In Search of Solutions*. Boston, MA: Springer US, 2006, págs. 1-24. ISBN: 978-0-387-36896-2. DOI: 10.1007/978-0-387-36896-2_1. URL: https://doi.org/10.1007/978-0-387-36896-2_1.
- [14] Carola Doerr et al. *IOHprofiler: A Benchmarking and Profiling Tool for Iterative Optimization Heuristics*. 2018. arXiv: 1810.05281 [cs.NE].
- [15] 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.
- [16] M. Fleischer. «The Measure of Pareto Optima Applications to Multi-objective Metaheuristics». En: *Evolutionary Multi-Criterion Optimization*. Ed. por Carlos M. Fonseca et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, págs. 519-533. ISBN: 978-3-540-36970-7.
- [17] Huiru Gao, Haifeng Nie y Ke Li. *Visualisation of Pareto Front Approximation: A Short Survey and Empirical Comparisons*. 2019. arXiv: 1903.01768 [cs.NE].
- [18] Marcos Gestal. «Computación evolutiva para el proceso de selección de variables en espacios de búsqueda multimodales». Tesis doct. Universidade da Coruña. Departamento de Tecnoloxías da Información e as Comunicacóns, 2009. ISBN: 978-84-693-5493-3. URL: <http://hdl.handle.net/2183/7173>.
- [19] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
- [20] Andreia P Guerreiro, Carlos M Fonseca y Luís Paquete. «The hypervolume indicator: Computational problems and algorithms». En: *ACM Computing Surveys (CSUR)* 54.6 (2021), págs. 1-42.
- [21] Nikolaus Hansen et al. «COCO: a platform for comparing continuous optimizers in a black-box setting». En: *Optimization Methods and Software* 36.1 (ago. de 2020), págs. 114-144. ISSN: 1029-4937. DOI: 10.1080/10556788.2020.1808977. URL: <http://dx.doi.org/10.1080/10556788.2020.1808977>.
- [22] Nikolaus Hansen et al. *COMparing Continuous Optimizers: numbbbo/COCO on Github*. Ver. v2.3. Mar. de 2019. DOI: 10.5281/zenodo.2594848. URL: <https://doi.org/10.5281/zenodo.2594848>.
- [23] Nikolaus Hansen et al. «Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions». Tesis doct. INRIA, 2009.

- [24] John H. Holland. «Genetic Algorithms». En: *Scientific American* 267.1 (1992), págs. 66-73. ISSN: 00368733, 19467087. URL: <http://www.jstor.org/stable/24939139> (visitado 11-01-2024).
- [25] John H. Holland. «Genetic Algorithms». En: *Scientific American* 267.1 (1992), págs. 66-73. ISSN: 00368733, 19467087. URL: <http://www.jstor.org/stable/24939139> (visitado 03-03-2024).
- [26] V. L. Huang et al. «Multi-objective optimization using self-adaptive differential evolution algorithm». En: *2009 IEEE Congress on Evolutionary Computation*. 2009, págs. 190-194. DOI: 10.1109/CEC.2009.4982947.
- [27] S. Huband et al. «A review of multiobjective test problems and a scalable test problem toolkit». En: *IEEE Transactions on Evolutionary Computation* 10.5 (2006), págs. 477-506. DOI: 10.1109/TEVC.2005.861417.
- [28] Simon Huband et al. «A Scalable Multi-objective Test Problem Toolkit». En: *Evolutionary Multi-Criterion Optimization*. Ed. por Carlos A. Coello Coello, Arturo Hernández Aguirre y Eckart Zitzler. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, págs. 280-295. ISBN: 978-3-540-31880-4.
- [29] Hisao Ishibuchi et al. «Modified Distance Calculation in Generational Distance and Inverted Generational Distance». En: *Evolutionary Multi-Criterion Optimization*. Ed. por António Gaspar-Cunha, Carlos Henggeler Antunes y Carlos Coello Coello. Cham: Springer International Publishing, 2015, págs. 110-125.
- [30] Y. Jiang y D. Vargas Vasconcellos. «Preliminary Results for Subpopulation Algorithm Based on Novelty (SAN) Compared with the State of the Art». En: *2021 5th IEEE International Conference on Cybernetics (CYBCONF)*. IEEE. 2021, págs. 067-072.
- [31] etall Katie R. Mollan. «Precise and accurate power of the rank-sum test for a continuous outcome». En: *Journal of Biopharmaceutical Statistics* 30.4 (2020). PMID: 32126888, págs. 639-648. DOI: 10.1080/10543406.2020.1730866. eprint: <https://doi.org/10.1080/10543406.2020.1730866>. URL: <https://doi.org/10.1080/10543406.2020.1730866>.
- [32] J. Kennedy y R. Eberhart. «Particle swarm optimization». En: *Proceedings of ICNN'95 - International Conference on Neural Networks*. Vol. 4. 1995, 1942-1948 vol.4. DOI: 10.1109/ICNN.1995.488968.
- [33] Pascal Kerschke y Heike Trautmann. «Comprehensive Feature-Based Landscape Analysis of Continuous and Constrained Optimization Problems Using the R-package flacco». En: *Applications in Statistical Computing – From Music Data Analysis to Industrial Quality Improvement*. Ed. por Nadja Bauer et al. Studies in Classification, Data Analysis, and Knowledge Organization. Springer, 2019, págs. 93-123. DOI: 10.48550/arXiv.1708.05258. URL: <https://doi.org/10.48550/arXiv.1708.05258>.

- [34] Pascal Kerschke y Heike Trautmann. «Comprehensive feature-based landscape analysis of continuous and constrained optimization problems using the R-package flacco». En: *Applications in Statistical Computing: From Music Data Analysis to Industrial Quality Improvement* (2019), págs. 93-123.
- [35] B. Li et al. «An improved two archive algorithm for many-objective optimization». En: *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE. 2014, págs. 2869-2876.
- [36] M. Li, S. Yang y X. Liu. «Bi-goal evolution for many-objective optimization problems». En: *Artificial Intelligence* 228 (2015), págs. 45-65.
- [37] Adriana Lara López. «Un estudio de las Estrategias Evolutivas para problemas de Optimización Multiobjetivo, su convergencia y operadores». Tesis doct. INSTITUTO POLITÉCNICO NACIONAL, 2003.
- [38] Scott M Lundberg y Su-In Lee. «A Unified Approach to Interpreting Model Predictions». En: *Advances in Neural Information Processing Systems*. Ed. por I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017.
- [39] Carlos Martinez y Juan Velásquez. «Conceptual Developments in Genetic Programming for Time Series Forecasting». En: *IEEE Latin America Transactions* 13 (ago. de 2015), págs. 2728-2733. DOI: 10.1109/TLA.2015.7332156.
- [40] Y. Ismael Martínez-Arévalo. «La heterogeneidad temporal mejora los algoritmos de búsqueda». Tesis de mtría. Universidad Nacional Autónoma de México, ene. de 2021.
- [41] Yoshio Ismael Martínez-Arévalo, Katya Rodríguez-Vazquez y Carlos Gershenson. «Temporal Heterogeneity Improves Speed and Convergence in Genetic Algorithms». arXiv:2203.13194. 2020. URL: <http://arxiv.org/abs/2203.13194>.
- [42] Patrick E. McKight y Julius Najab. «Kruskal-Wallis Test». En: *The Corsini Encyclopedia of Psychology*. John Wiley y Sons, Ltd, 2010, págs. 1-1. ISBN: 9780470479216. DOI: <https://doi.org/10.1002/9780470479216.corpsy0491>.
- [43] Kaisa Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, 1999.
- [44] Pablo Chacón Montes. «Algoritmos Evolutivos: algoritmos genéticos y de cuasi-especie». En: *Vida artificial* 10 (1995), págs. 87.
- [45] Maysam Orouskhani, Mohammad Teshnehlab y Mohammad Ali Nekoui. «Evolutionary dynamic multi-objective optimization algorithm based on Borda count method». En: *International Journal of Machine Learning and Cybernetics* 10 (2019), págs. 1931-1959.
- [46] F. Pedregosa et al. «Scikit-learn: Machine Learning in Python». En: *Journal of Machine Learning Research* 12 (2011), págs. 2825-2830.

- [47] Victor Manuel Sanchez Sanchez, Carlos Gershenson Garcia y Carlos Ignacio Hernandez Castellanos. «On the Effect of Temporal Heterogeneity on Selection Pressure of Evolutionary Algorithms». En: *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*. GECCO '23 Companion. Lisbon, Portugal: Association for Computing Machinery, 2023, págs. 479-482. DOI: 10.1145/3583133.3590749. URL: <https://doi.org/10.1145/3583133.3590749>.
- [48] Fernanda Sánchez-Puig et al. «Heterogeneity extends criticality». En: *Frontiers in Complex Systems* 1 (2023). ISSN: 2813-6187. DOI: 10.3389/fcpxs.2023.1111486. URL: <https://doi.org/10.3389/fcpxs.2023.1111486>.
- [49] Hans-Paul Schwefel. *Evolution and Optimum Seeking*. Ene. de 1995. ISBN: 978-0-471-57148-3.
- [50] Ke Shang et al. «A Survey on the Hypervolume Indicator in Evolutionary Multi-objective Optimization». En: *IEEE Transactions on Evolutionary Computation* 25.1 (2021), págs. 1-20. DOI: 10.1109/TEVC.2020.3013290.
- [51] Yanan Sun, Gary G. Yen y Zhang Yi. «IGD Indicator-Based Evolutionary Algorithm for Many-Objective Optimization Problems». En: *IEEE Transactions on Evolutionary Computation* 23.2 (2019), págs. 173-187. DOI: 10.1109/TEVC.2018.2791283.
- [52] Ryoji Tanabe e Hisao Ishibuchi. «Non-Elitist Evolutionary Multi-Objective Optimizers Revisited». En: *Proceedings of the Genetic and Evolutionary Computation Conference*. GECCO '19. Prague, Czech Republic: Association for Computing Machinery, 2019, págs. 612-619. ISBN: 9781450361118. DOI: 10.1145/3321707.3321754. URL: <https://doi.org/10.1145/3321707.3321754>.
- [53] Ryoji Tanabe e Hisao Ishibuchi. «Non-elitist evolutionary multi-objective optimizers revisited». En: *Proceedings of the Genetic and Evolutionary Computation Conference*. 2019, págs. 612-619.
- [54] Risto Trajanov et al. *Explainable Landscape Analysis in Automated Algorithm Performance Prediction*. 2022. arXiv: 2203.11828 [cs.LG].
- [55] Thomas Willett y Paul Flacco. «The Reallocation Effects on Exchange Rate Fluctuations Under Uncertainty Where Speculation is Efficient». En: (1984).
- [56] Wei Zeng, Azadeh Davoodi y Rasit Onur Topaloglu. «Explainable DRC Hotspot Prediction with Random Forest and SHAP Tree Explainer». En: *2020 Design, Automation and Test in Europe Conference and Exhibition*. 2020, págs. 1151-1156. DOI: 10.23919/DATE48585.2020.9116488.
- [57] Hu Zhang et al. «Balancing exploration and exploitation in multiobjective evolutionary optimization». En: *Information Sciences* 497 (2019), págs. 129-148. ISSN: 0020-0255. URL: <https://doi.org/10.1016/j.ins.2019.05.046>.
- [58] 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. DOI: 10.1109/TEVC.2007.892759.

- [59] Eckart Zitzler, Kalyanmoy Deb y Lothar Thiele. «Comparison of Multiobjective Evolutionary Algorithms: Empirical Results». En: *Evolutionary Computation* 8.2 (2000), págs. 173-195. DOI: 10.1162/106365600568202.