



**Laboratorio Nacional de Informática Avanzada,
A.C**

**SELECCIÓN DE ATRIBUTOS CON UN
ENFOQUE EVOLUTIVO MULTI OBJETIVO**

TESIS

Que presenta:

José Alfredo Mendoza Mota

Para obtener el grado de:

Maestro en Computación Aplicada

Director:

Dr. Efrén Mezura Montes

Co-Director:

Dr. Rafael Rivera López

Xalapa, Veracruz, México

Diciembre 2021

Dedicatoria

Dedicado a mis padres que siempre me han apoyado y a mis hijos los cuales siempre me han impulsado a cumplir todas mis metas.

Agradecimientos

Agradezco a Dios por permitirme terminar este reto en mi vida.

Agradezco a mis asesores de tesis, el Dr. Efrén Mezura Montes, así como al Dr. Rafael Rivera López, por su tiempo, paciencia, apoyo y dedicación que me brindaron para la realización de este trabajo de tesis.

A mis padres que me han apoyado siempre en las decisiones y retos que he tomado en mi vida.

A mis hijos, que me han impulsado siempre a superarme y ser una mejor persona.

A mis profesores, que compartieron sus conocimientos conmigo durante este tiempo.

A mis compañeros y amigos que me apoyaron durante esta etapa de mi vida.

Por último, agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT), por el apoyo económico otorgado para la realización de la Maestría en Computación Aplicada (MCA) mediante el número de becario 953990.

A todos, muchas gracias

Resumen

La selección de características es un proceso fundamental dentro del preprocesamiento de datos, el cual consiste en seleccionar un subconjunto de características relevantes de un conjunto de datos original, esto con el fin de mejorar el rendimiento de los algoritmos a utilizar en la fase de minería de datos. Para llevar a cabo este proceso existen 3 métodos: Filtrado, Envoltura y Embebido los cuales se diferencian por la forma de colaboración entre el proceso de selección de características y el algoritmo de clasificación.

En el presente proyecto de tesis se implementa el método de selección de características de Envoltura utilizando el algoritmo evolutivo multiobjetivo de Evolución Diferencial Generalizada versión 3 (GDE 3, por sus siglas en inglés), enfocado en minimizar el error de clasificación utilizando el menor número de características seleccionadas posibles y utilizado una representación basada en permutaciones con el objetivo de obtener mejores resultados que los algoritmos tradicionales y evolutivos mono objetivo de selección de características.

Los resultados obtenidos en esta investigación utilizando 25 conjuntos de datos, nos muestran que el algoritmo propuesto (FSGDE 3) mejora considerablemente la reducción del error de clasificación utilizando un menor número de características comparado con los algoritmos tradicionales, sin embargo comparado con el algoritmo evolutivo mono objetivo de Evolución Diferencial, en su mayoría la precisión es menor, esto debido a que el algoritmo propuesto considera la reducción del número de características, lo que implica que a un menor número de características seleccionadas, el error aumente. De igual manera al comparar los resultados del algoritmo FSGDE 3 contra otro algoritmo evolutivo multiobjetivo (FSNSGA II), se demostró que FSGDE 3 es más competitivo para la solución de problemas de selección de características. Lo cual concluye que FSGDE 3 cumplió el objetivo propuesto para este proyecto de tesis.

Índice general

Dedicatoria	I
Agradecimientos	II
Resumen	III
Índice general	V
Índice de figuras	IX
Índice de tablas	X
Índice de algoritmos	XI
1. Introducción	1
1.1. Planteamiento de Problema	3
1.2. Objetivos	4
1.2.1. Objetivo general	4
1.2.2. Objetivos específicos	4
1.3. Hipótesis	4
1.4. Justificación	5
1.5. Trabajo relacionado	5
1.6. Organización del documento	6
2. Aprendizaje Automático	7
2.1. Tipos de Algoritmos	7
2.1.1. Aprendizaje Supervisado	8
2.1.2. Algoritmos de aprendizaje supervisado	8
2.1.3. Aprendizaje No Supervisado	10
2.1.4. Algoritmos de agrupamiento	11
2.2. Selección de características	11
2.2.1. Estrategias de selección de características	12
2.2.2. Algoritmos tradicionales	13

2.2.3.	Ventajas y desventajas de las estrategias de selección de características	14
3.	Cómputo Evolutivo	15
3.1.	Algoritmos evolutivos	16
3.1.1.	Algoritmos Genéticos (AG)	17
3.1.2.	Programación Genética (PG)	17
3.1.3.	Estrategias Evolutivas (EE)	18
3.1.4.	Programación Evolutiva (PE)	18
3.1.5.	Evolución Diferencial (ED)	19
3.2.	Optimización Multiobjetivo	19
3.2.1.	Algoritmos Evolutivos en la Optimización Multiobjetivo	22
3.2.2.	Indicadores de Desempeño	24
4.	Evolución Diferencial Multiobjetivo	31
4.1.	Generalized Differential Evolution (GDE)	31
4.2.	Feature Selection Generalized Differential Evolution 3 (FSGDE 3)	34
4.2.1.	Representación	34
4.2.2.	Funciones Objetivo	35
4.2.3.	Población de Soluciones	35
4.2.4.	Mecanismo de Selección Padres	36
4.2.5.	Operadores de variación	37
4.2.6.	Mecanismo de Reemplazo	39
5.	Experimentación y Resultados	41
5.1.	Marco Experimental	41
5.2.	Selección de datos de prueba	42
5.3.	Sintonización de parámetros	42
5.3.1.	Resultados de IRACE	43
5.4.	Diseño Experimental	45
5.4.1.	Experimento 1	45
5.4.2.	Experimento 2	46
5.5.	Resultados	46
5.5.1.	Experimento 1	46
5.5.2.	Experimento 2	59
6.	Conclusiones y Trabajo futuro	65
6.1.	Conclusiones	65
6.2.	Trabajo Futuro	66
	Anexos	67
	A. Anexos	68
	B. Anexos-Continuación	108
	C. Anexos-microarreglos	149

Índice de figuras

1.1. Proceso de descubrimiento de conocimiento en bases de datos	2
2.1. Tipos de aprendizaje.	8
2.2. Proceso de Aprendizaje Supervisado.	9
2.3. Estructura de un árbol de decisión.	9
2.4. Estructura de una Red Neuronal.	9
2.5. Ejemplo de una red Bayesiana.	10
2.6. Ejemplo de una Máquina de Soporte Vectorial.	10
2.7. Proceso de Aprendizaje No supervisado	11
2.8. Ejemplo del algoritmo K-means.	11
2.9. Enfoques de Selección de Características.	12
3.1. Metáfora Evolutiva.	15
3.2. Modelo General de un Algoritmo Evolutivo.	16
3.3. Frente de Pareto	22
4.1. Método Envoltorio.	34
4.2. Esquema de codificación.	35
4.3. Población de Soluciones.	36
4.4. Selección de Vectores.	36
4.5. Operadores de Variación del algoritmo FSGDE 3.	37
4.6. Procedimiento del operador de Mutación.	38
4.7. Procedimiento de reparación.	39
5.1. Resultados de FSGDE 3 utilizando el conjunto de datos m-libras.	51
5.2. Resultados de FSNSGA II utilizando el conjunto de datos m-libras.	52
5.3. Resultados de FSGDE 3 utilizando el conjunto de datos semeion.	53
5.4. Frente acumulado del conjunto de datos Semeion	54
5.5. Resultados de FSNSGA II utilizando el conjunto de datos semeion.	55
5.6. Resultados de FSGDE 3 utilizando el micro arreglo SRBCT.	56
5.7. Resultados de FSNSGA II utilizando el micro arreglo SRBCT.	57
5.8. Comparación del frente acumulado de FSGDE 3 y FSNSGA II.	58

A.1. Frentes de Pareto del Conjunto de datos arrhythmia con FSGDE 3. (1 - 2) . . .	68
A.2. Frentes de Pareto del Conjunto de datos arrhythmia con FSGDE 3. (2 - 2) . . .	69
A.3. Frentes de Pareto del Conjunto de datos arrhythmia con FSNSGA II. (1 - 2) . . .	70
A.4. Frentes de Pareto del Conjunto de datos arrhythmia con FSNSGA II. (2 - 2) . . .	71
A.5. Frentes de Pareto del Conjunto de datos audiology con FSGDE3. (1 - 2) . . .	72
A.6. Frentes de Pareto del Conjunto de datos audiology con FSGDE3. (2 - 2) . . .	73
A.7. Frentes de Pareto del Conjunto de datos audiology con FSNSGA II. (1 - 2) . . .	74
A.8. Frentes de Pareto del Conjunto de datos audiology con FSNSGA II. (2 - 2) . . .	75
A.9. Frentes de Pareto del Conjunto de datos australian con FSGDE3. (1 - 2) . . .	76
A.10. Frentes de Pareto del Conjunto de datos australian con FSGDE3. (2 - 2) . . .	77
A.11. Frentes de Pareto del Conjunto de datos australian con FSNSGA II. (1 - 2) . . .	78
A.12. Frentes de Pareto del Conjunto de datos australian con FSNSGA II. (2 - 2) . . .	79
A.13. Frentes de Pareto del Conjunto de datos crx con FSGDE3. (1 - 2)	80
A.14. Frentes de Pareto del Conjunto de datos crx con FSGDE3. (2 - 2)	81
A.15. Frentes de Pareto del Conjunto de datos crx con FSNSGA II. (1 - 2)	82
A.16. Frentes de Pareto del Conjunto de datos crx con FSNSGA II. (2 - 2)	83
A.17. Frentes de Pareto del Conjunto de datos cylinder-bands con FSGDE3. (1 - 2) . . .	84
A.18. Frentes de Pareto del Conjunto de datos cylinder-bands con FSGDE3. (2 - 2) . . .	85
A.19. Frentes de Pareto del Conjunto de datos cylinder-bands con FSNSGA II. (1 - 2)	86
A.20. Frentes de Pareto del Conjunto de datos cylinder-bands con FSNSGA II. (2 - 2)	87
A.21. Frentes de Pareto del Conjunto de datos dermatology con FSGDE3. (1 - 2) . . .	88
A.22. Frentes de Pareto del Conjunto de datos dermatology con FSGDE3. (2 - 2) . . .	89
A.23. Frentes de Pareto del Conjunto de datos dermatology con FSNSGA II. (1 - 2) . . .	90
A.24. Frentes de Pareto del Conjunto de datos dermatology con FSNSGA II. (2 - 2) . . .	91
A.25. Frentes de Pareto del Conjunto de datos german-c con FSGDE3. (1 - 2) . . .	92
A.26. Frentes de Pareto del Conjunto de datos german-c con FSGDE3. (2 - 2) . . .	93
A.27. Frentes de Pareto del Conjunto de datos german-c con FSNSGA II. (1 - 2) . . .	94
A.28. Frentes de Pareto del Conjunto de datos german-c con FSNSGA II. (2 - 2) . . .	95
A.29. Frentes de Pareto del Conjunto de datos h-valley con FSGDE3. (1 - 2)	96
A.30. Frentes de Pareto del Conjunto de datos h-valley con FSGDE3. (2 - 2)	97
A.31. Frentes de Pareto del Conjunto de datos h-valley con FSNSGA II. (1 - 2)	98
A.32. Frentes de Pareto del Conjunto de datos h-valley con FSNSGA II. (2 - 2)	99
A.33. Frentes de Pareto del Conjunto de datos ionosphere con FSGDE3. (1 - 2)	100
A.34. Frentes de Pareto del Conjunto de datos ionosphere con FSGDE3. (2 - 2)	101
A.35. Frentes de Pareto del Conjunto de datos ionosphere con FSNSGA II. (1 - 2)	102
A.36. Frentes de Pareto del Conjunto de datos ionosphere con FSNSGA II. (2 - 2)	103
A.37. Frentes de Pareto del Conjunto de datos m-libras con FSGDE3. (1 - 2)	104
A.38. Frentes de Pareto del Conjunto de datos m-libras con FSGDE3. (2 - 2)	105
A.39. Frentes de Pareto del Conjunto de datos m-libras con FSNSGA II. (1 - 2)	106
A.40. Frentes de Pareto del Conjunto de datos m-libras con FSNSGA II. (2 - 2)	107
B.1. Frentes de Pareto del Conjunto de datos musk-1 con FSGDE3. (1 - 2)	109
B.2. Frentes de Pareto del Conjunto de datos musk-1 con FSGDE3. (2 - 2)	110

B.3. Frentes de Pareto del Conjunto de datos musk-1 con FSNSGA II. (1 - 2) . . .	111
B.4. Frentes de Pareto del Conjunto de datos musk-1 con FSNSGA II. (2 - 2) . . .	112
B.5. Frentes de Pareto del Conjunto de datos parkinsons con FSGDE3. (1 - 2) . . .	113
B.6. Frentes de Pareto del Conjunto de datos parkinsons con FSGDE3. (2 - 2) . . .	114
B.7. Frentes de Pareto del Conjunto de datos parkinsons con FSNSGA II. (1 - 2)	115
B.8. Frentes de Pareto del Conjunto de datos parkinsons con FSNSGA II. (2 - 2)	116
B.9. Frentes de Pareto del Conjunto de datos semeion con FSGDE3. (1 - 2) . . .	117
B.10. Frentes de Pareto del Conjunto de datos semeion con FSGDE3. (2 - 2) . . .	118
B.11. Frentes de Pareto del Conjunto de datos semeion con FSNSGA II. (1 - 2) . . .	119
B.12. Frentes de Pareto del Conjunto de datos semeion con FSNSGA II. (2 - 2) . . .	120
B.13. Frentes de Pareto del Conjunto de datos sonar con FSGDE3. (1 - 2)	121
B.14. Frentes de Pareto del Conjunto de datos sonar con FSGDE3. (2 - 2)	122
B.15. Frentes de Pareto del Conjunto de datos sonar con FSNSGA II. (1 - 2)	123
B.16. Frentes de Pareto del Conjunto de datos sonar con FSNSGA II. (2 - 2)	124
B.17. Frentes de Pareto del Conjunto de datos soybean con FSGDE3. (1 - 2)	125
B.18. Frentes de Pareto del Conjunto de datos soybean con FSGDE3. (2 - 2)	126
B.19. Frentes de Pareto del Conjunto de datos soybean con FSNSGA II. (1 - 2)	127
B.20. Frentes de Pareto del Conjunto de datos soybean con FSNSGA II. (2 - 2)	128
B.21. Frentes de Pareto del Conjunto de datos spectf con FSGDE3. (1 - 2)	129
B.22. Frentes de Pareto del Conjunto de datos spectf con FSGDE3. (2 - 2)	130
B.23. Frentes de Pareto del Conjunto de datos spectf con FSNSGA II. (1 - 2)	131
B.24. Frentes de Pareto del Conjunto de datos spectf con FSNSGA II. (2 - 2)	132
B.25. Frentes de Pareto del Conjunto de datos splice con FSGDE3. (1 - 2)	133
B.26. Frentes de Pareto del Conjunto de datos splice con FSGDE3. (2 - 2)	134
B.27. Frentes de Pareto del Conjunto de datos splice con FSNSGA II. (1 - 2)	135
B.28. Frentes de Pareto del Conjunto de datos splice con FSNSGA II. (2 - 2)	136
B.29. Frentes de Pareto del Conjunto de datos vehicle con FSGDE3. (1 - 2)	137
B.30. Frentes de Pareto del Conjunto de datos vehicle con FSGDE3. (2 - 2)	138
B.31. Frentes de Pareto del Conjunto de datos vehicle con FSNSGA II. (1 - 2)	139
B.32. Frentes de Pareto del Conjunto de datos vehicle con FSNSGA II. (2 - 2)	140
B.33. Frentes de Pareto del Conjunto de datos vote con FSGDE3. (1 - 2)	141
B.34. Frentes de Pareto del Conjunto de datos vote con FSGDE3. (2 - 2)	142
B.35. Frentes de Pareto del Conjunto de datos vote con FSNSGA II. (1 - 2)	143
B.36. Frentes de Pareto del Conjunto de datos vote con FSNSGA II. (2 - 2)	144
B.37. Frentes de Pareto del Conjunto de datos wdbc con FSGDE3. (1 - 2)	145
B.38. Frentes de Pareto del Conjunto de datos wdbc con FSGDE3. (2 - 2)	146
B.39. Frentes de Pareto del Conjunto de datos wdbc con FSNSGA II. (1 - 2)	147
B.40. Frentes de Pareto del Conjunto de datos wdbc con FSNSGA II. (2 - 2)	148
C.1. Frentes de Pareto del Conjunto de datos all-aml con FSGDE3. (1 - 2)	150
C.2. Frentes de Pareto del Conjunto de datos all-aml con FSGDE3. (2 - 2)	151
C.3. Frentes de Pareto del Conjunto de datos all-aml con FSNSGA II. (1 - 2)	152
C.4. Frentes de Pareto del Conjunto de datos all-aml con FSNSGA II. (2 - 2)	153
C.5. Frentes de Pareto del Conjunto de datos colon con FSGDE3. (1 - 2)	154
C.6. Frentes de Pareto del Conjunto de datos colon con FSGDE3. (2 - 2)	155

C.7. Frentes de Pareto del Conjunto de datos colon con FSNSGA II. (1 - 2) . . .	156
C.8. Frentes de Pareto del Conjunto de datos colon con FSNSGA II. (2 - 2) . . .	157
C.9. Frentes de Pareto del Conjunto de datos leukemia con FSGDE3. (1 - 2) . . .	158
C.10. Frentes de Pareto del Conjunto de datos leukemia con FSGDE3. (2 - 2) . . .	159
C.11. Frentes de Pareto del Conjunto de datos leukemia con FSNSGA II. (1 - 2) .	160
C.12. Frentes de Pareto del Conjunto de datos leukemia con FSNSGA II. (2 - 2) .	161
C.13. Frentes de Pareto del Conjunto de datos lymphoma con FSGDE3. (1 - 2) . .	162
C.14. Frentes de Pareto del Conjunto de datos lymphoma con FSGDE3. (2 - 2) . .	163
C.15. Frentes de Pareto del Conjunto de datos lymphoma con FSNSGA II. (1 - 2)	164
C.16. Frentes de Pareto del Conjunto de datos lymphoma con FSNSGA II. (2 - 2)	165
C.17. Frentes de Pareto del Conjunto de datos srbct con FSGDE3. (1 - 2)	166
C.18. Frentes de Pareto del Conjunto de datos srbct con FSGDE3. (2 - 2)	167
C.19. Frentes de Pareto del Conjunto de datos srbct con FSNSGA II. (1 - 2) . . .	168
C.20. Frentes de Pareto del Conjunto de datos srbct con FSNSGA II. (2 - 2) . . .	169

Índice de tablas

2.1. Comparación de las 3 estrategias de selección de características.	13
3.1. Relaciones de preferencia para comparación de vectores objetivo.	21
5.1. Parámetros utilizados en los experimentos. Para ambos algoritmos, los parámetros fueron calibrados mediante IRACE	42
5.2. Descripción de los datasets utilizados en los experimentos	43
5.3. Clasificación de los conjuntos de datos basado en su número de características	44
5.4. Conjunto de datos Utilizados para la calibración de parámetros utilizando IRACE.	44
5.5. Parámetros y rangos de valores para calibración con IRACE.	45
5.6. Tabla de resultados considerando la precisión de la mejor solución del frente acumulado.	47
5.7. Tabla de resultados considerando el tamaño del subconjunto de la mejor solución del conjunto de soluciones no dominadas.	48
5.8. Prueba de Wilcoxon utilizando el Indicador de Desempeño Spacing.	49
5.9. Prueba de Wilcoxon utilizando el Indicador de Desempeño Hipervolumen.	50
5.10. Conjunto de datos utilizados en el segundo experimento	59
5.11. Tabla de resultados obtenidos utilizando la mejor solución	60
5.12. Reducción de dimensionalidad de los conjuntos de datos.	60
5.13. Tabla de resultados obtenidos utilizando la peor solución	61
5.14. Reducción de dimensionalidad de los conjuntos de datos.	62
5.15. Tabla de resultados obtenidos utilizando la solución intermedia o punto rodilla	63
5.16. Reducción de dimensionalidad de los conjuntos de datos.	63

Índice de algoritmos

1.	Algoritmo Genético Simple	17
2.	Algoritmo de Estrategias Evolutivas	18
3.	Algoritmo de Programación Evolutiva	18
4.	Algoritmo de Evolución Diferencial	19
5.	Evolución Diferencial Generalizada 3	33
6.	Evolución Diferencial Generalizada 3 para selección de características	40

Capítulo 1

Introducción

El Descubrimiento de conocimiento en base de datos (KDD, por sus siglas en Inglés) es una metodología que intenta explotar la enorme cantidad de información existente hoy en día, extrayendo conocimiento que pueda auxiliar a los seres humanos para llevar a cabo tareas de forma más eficiente [22].

KDD corresponde a un proceso que itera sobre sí mismo para explorar grandes volúmenes de datos y así poder encontrar o determinar relaciones entre ellos. Este proceso extrae información para ser utilizada en la toma de decisiones basadas en modelos de datos[26]. Esta metodología se divide en 6 etapas principales como se muestra en la Fig. 1.1.

Etapas de la Metodología KDD

1. **Obtención de datos.** Integración de información de distintas fuentes en un mismo conjunto de datos.
2. **Selección de datos.** Etapa donde los datos relevantes para el análisis son extraídos del conjunto universal.
3. **Preprocesamiento.** Esta etapa consiste en la preparación y limpieza de los datos extraídos y necesarios para las fases siguientes. En esta etapa se utilizan diversas estrategias para manipular datos faltantes o inconsistentes, obteniéndose al final una estructura de datos adecuada para su posterior transformación.
4. **Transformación.** Consiste en el tratamiento preliminar de los datos, transformación y generación de nuevas variables, a partir de las ya existentes, con una estructura de datos apropiada.
5. **Minería de Datos.** Es la fase en donde métodos inteligentes son aplicados con el objetivo de extraer patrones potencialmente útiles y comprensibles para generar modelos que ayuden a comprender la información obtenida de los datos.
6. **Interpretación.** Esta fase evalúa los patrones y modelos obtenidos, ya que, antes de ser interpretados para la obtención de conocimiento, debe de comprobarse que tienen la calidad suficiente para poder realizar dicha interpretación.

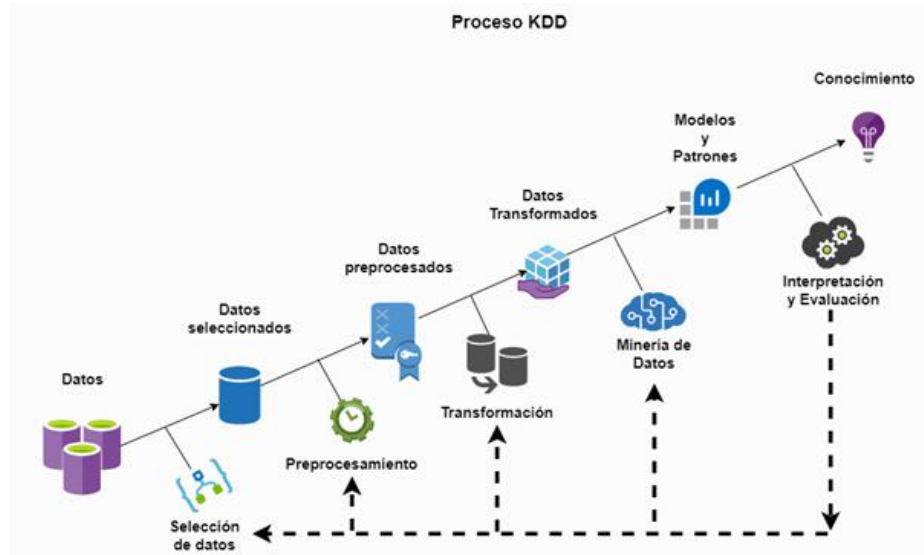


Figura 1.1: Proceso de descubrimiento de conocimiento en bases de datos

Debido al tamaño de los conjuntos de datos, a la presencia de ruido, datos inconsistentes, redundantes, etc., se hace necesaria la aplicación de técnicas de preprocesamiento sobre dichos conjuntos.

La preparación o preprocesamiento de datos es una de las tareas que suele consumir tiempo dentro del proceso KDD. La importancia de la preparación de los datos se debe principalmente a los aspectos siguientes, los cuales son definidos en [22]:

- Los datos crudos suelen estar incompletos, ser inconsistentes o simplemente presentar ruido.
- La preparación genera conjuntos de datos que son menores que el conjunto original, lo cual puede mejorar significativamente el rendimiento de los algoritmos en la fase de minería de datos.
- La preparación da lugar a datos de calidad, al recuperar instancias incompletas o corregir errores.

El preprocesamiento de datos se lleva a cabo siguiendo ciertas estrategias establecidas [21]:

- La limpieza de datos se utiliza para aumentar su calidad al nivel requerido mediante técnicas de análisis selectivo.
- La reducción de datos consiste en decidir qué datos deben ser utilizados para el análisis. El criterio que se sigue para su selección se basa en la relevancia con respecto a los objetivos que se establezcan para el modelo a generar.
- La integración de datos se basa en combinar múltiples registros para crear nuevos valores. Esta tarea comprende la producción de características derivadas o transformaciones de los valores de las características ya existentes.

- La transformación de datos consiste principalmente en modificaciones llevadas a cabo sobre los datos existentes, sin perjudicar el rendimiento de la fase de minería de datos.

Debido a que las técnicas de minería de datos generan los modelos basándose precisamente en los datos disponibles, con frecuencia se generan modelos complejos conforme los datos crecen en número. Lo anterior puede provocar inconvenientes adicionales tales como son el aumento en el tiempo de respuesta de los modelos, el aumento en la sensibilidad y posibilidad de sobreajuste, así como modelos difíciles de comprender. Buscando disminuir los efectos anteriores, se hace necesario un preprocesamiento adecuado que seleccione la información relevante del conjunto de datos, ya que, debido al efecto negativo de características irrelevantes en la mayoría de los esquemas de aprendizaje automático, es común llevar a cabo un proceso de selección de características previo al aprendizaje.

La selección de características es el proceso con el cual se eligen las variables más relevantes de un conjunto de datos, con el objetivo de mejorar el rendimiento de predicción de los algoritmos utilizados en la fase de la minería de datos [54]. De acuerdo a la forma de evaluación, el proceso de selección de características se divide en tres categorías:

1. **Filtrado.** Los métodos de filtrado realizan una evaluación de la correlación entre cada característica y el atributo de destino, esto mediante una medida estadística que designa una puntuación a cada característica, de esta forma se clasifican según su puntuación, lo cual funciona como una condición para mantener o eliminar una característica específica. Una de las técnicas más utilizadas es el algoritmo ReliefF [23].
2. **Envoltura (Wrapper).** Este esquema hace uso de un algoritmo de aprendizaje automático para la selección del mejor subconjunto de características, utilizando su rendimiento como criterio de evaluación. Algunos ejemplos de estos métodos son los de selección secuencial hacia adelante [28], eliminación secuencial hacia atrás [29] y eliminación recursiva de características [17].
3. **Embebido.** En esta categoría se combinan los esquemas anteriores. Es decir, el método de aprendizaje involucra una selección de características, como lo es el caso de los árboles de decisión. Otros ejemplos de estos son los métodos de regresión de LASSO y RIDGE, que tienen funciones de penalización para reducir el sobreajuste.

Existen algoritmos de aprendizaje automático que están diseñados para aprender aquellas características que son más apropiadas para la toma de decisiones. Un ejemplo de ellos son los árboles de decisión, los cuales eligen el atributo más prometedor, para llevar a cabo la división en cada nodo interno. Sin embargo, estos llegan a fallar cuando la cantidad de características aumenta, debido a que conforme se va profundizando en el árbol, menor es la cantidad de datos disponibles para decidir.

1.1. Planteamiento de Problema

En la literatura especializada se han propuesto diferentes métodos para la selección de características, tales como SFS (Sequential Forward Selection) y SBE (Sequential Backward Elimination). Incluso se tienen opciones como el análisis de componentes principales

(PCA, por sus siglas en Inglés) que además de reducir la cantidad de características, puede unir varios de ellos y crear nuevos. Sin embargo, su desempeño puede ser limitado en conjuntos de datos con grandes cantidades de características, en presencia de ruido y otras fuentes de dificultad.

Por lo anterior, en años recientes se han propuesto técnicas alternativas para la selección de características basadas en algoritmos evolutivos [56]. Si bien los primeros esfuerzos se centraron en algoritmos genéticos con representación binaria, también se han propuesto adaptaciones de algoritmos evolutivos para espacios continuos, como lo es el algoritmo de Evolución Diferencial, usando representaciones basadas en permutaciones, para resolver el problema de selección de características [36]. Sin embargo, este tipo de propuesta con representaciones alternas se ha adoptado para un solo objetivo (Porcentaje de Clasificación) y no se han extendido del todo a enfoques multi-objetivo. Por ello, en este trabajo se busca realizar una optimización al proceso de selección de características utilizando un algoritmo evolutivo multi-objetivo, el cual encuentre compromisos entre la minimización del error de clasificación y el número de características seleccionadas.

1.2. Objetivos

1.2.1. Objetivo general

Diseñar un algoritmo evolutivo multi-objetivo con representación basada en permutaciones para la selección de características con un método de envoltura, minimizando el error de clasificación, así como el número de características.

1.2.2. Objetivos específicos

- Analizar cualitativamente los algoritmos evolutivos multi-objetivo para seleccionar el algoritmo más adecuado.
- Diseñar el algoritmo evolutivo multi-objetivo previamente seleccionado para la selección de características que reduzca el error en la clasificación de los datos, así como el número de características.
- Comparar los resultados del algoritmo diseñado con los métodos tradicionales de selección de características, incluyendo algoritmos evolutivos con enfoques mono-objetivo.

1.3. Hipótesis

Es posible, mediante un algoritmo evolutivo multi-objetivo con representación basada en permutaciones, y con un método de envoltura, obtener mejores resultados de clasificación con conjuntos de características de menor tamaño, que los obtenidos por métodos tradicionales para selección de características.

1.4. Justificación

Con el impacto que actualmente tiene la Inteligencia Artificial en el tema del descubrimiento de conocimiento en base de datos, particularmente la computación evolutiva, se busca poder hacer uso de ella para encontrar una solución altamente competitiva, que nos ayude a resolver el problema planteado. Por ello, se propone trabajar con este tipo de algoritmos para optimizar el proceso de selección de características, encontrando un compromiso entre el número de características seleccionadas y el error de clasificación.

1.5. Trabajo relacionado

En muchas ocasiones, el número de características en un conjunto de datos puede ser muy alto, lo que representa un enorme espacio de búsqueda para poder buscar el mejor subconjunto de ellas para llevar a cabo un exitoso proceso de clasificación. En dichas circunstancias puede complicarse la generación de modelos con buen desempeño, más aún cuando algunas de las características son irrelevantes para tal fin. Por ello, es muy común reducir la dimensionalidad de los datos mediante una transformación de características o una selección de características, dicha selección puede realizarse de tres maneras:

1. Conociendo a priori el número de características necesarias.
2. Seleccionando las características para que el modelo de aprendizaje sea mejor.
3. Seleccionando el menor número de características para que el modelo tenga un mejor rendimiento.

Lo anterior puede realizarse por métodos no supervisados, como lo son la selección de subconjuntos por columna [14], el cual consiste en minimizar el error de reconstrucción de una matriz de datos mediante la selección de un subconjunto de características. Aunque la reconstrucción y selección de características son considerados dos métodos independientes para reducir la dimensionalidad, se han realizado trabajos para utilizar los dos métodos en conjunto, donde uno de ellos lo realiza mediante una proyección selectiva de características lineales no supervisadas para optimizar la relación entre los datos y la reducción del ruido [45].

Por otro lado, los algoritmos evolutivos han sido de mucha ayuda cuando se trata de la selección y construcción de características. La construcción de características se utiliza para precisamente generar nuevas y con mayor afinidad al modelo, y que se derivan de las características ya existentes. Esta construcción puede realizarse utilizando un algoritmo de programación genética [48]. Sin embargo, en conjuntos de datos con muchas características, este algoritmo utiliza una segmentación de datos para formar conjuntos con una dimensión más pequeña, lo cual mejora su desempeño. Por otro lado, la selección de características puede realizarse con algoritmos como el de Evolución Diferencial (ED) [18] y el algoritmo de optimización por cúmulo de partículas (PSO, por sus siglas en Inglés)[40][55][50], siendo este último el más popular.

Uno de los métodos utilizados para la selección de características con el algoritmo PSO, es mediante una discretización previa a la selección [47], ya que cada característica influye

de manera independiente. Otro método utilizado para la selección de características en un conjunto de datos de gran tamaño es utilizando el método de filtrado y envoltura en conjunto con el algoritmo PSO [49], logrando obtener subconjuntos de características más pequeños con un rendimiento de clasificación alto y una reducción en el tiempo de ejecución. Por otro lado, en [32] se propone un método que realiza un agrupamiento jerárquico donde se construyen varios conjuntos sustitutos utilizando el algoritmo PSO, el cual adopta un mecanismo de búsqueda que utiliza precisamente estos conjuntos logrando así una reducción en el tiempo de ejecución. Lo anterior mejora el rendimiento y ayuda a reducir el sobreajuste, y de esta manera el algoritmo puede elegir automáticamente conjuntos sustitutos para mejorar la precisión del algoritmo de clasificación. Otro método propuesto utiliza la selección y discretización de características de forma simultánea [46], esto por que en la etapa de discretización no se considera la interacción entre las características, por lo que dicho proceso simultáneo aumenta de manera significativa el rendimiento del porcentaje en la clasificación.

1.6. Organización del documento

El presente documento de tesis se organiza de la siguiente manera: en el capítulo 2 se presentan los conceptos básicos de aprendizaje automático, sus principales algoritmos y particularmente aquellos para la selección de características. En el capítulo 3 se muestran los conceptos sobre algoritmos evolutivos, centrándose en aquellos que resuelven problemas de optimización multi-objetivo. En el capítulo 4 se detallan los algoritmos evolutivos utilizados para la selección de características, así como la selección del algoritmo a utilizar. En el capítulo 5 se realiza la fase de experimentación, presentación y discusión de resultados. Finalmente, en el capítulo 6 se muestran las conclusiones generales de este trabajo, así como los trabajos futuros que derivan de la presente investigación.

Capítulo 2

Aprendizaje Automático

El aprendizaje automático es una subárea de la Inteligencia Artificial (IA), que se enfoca en desarrollar sistemas que puedan aprender en función de los datos que consumen, es decir, simular inteligencia humana que convierte experiencias en conocimiento.

El proceso de aprendizaje automático comparte características con la minería de datos, ya que ambos procesos buscan encontrar conocimiento dentro de un conjunto de datos. La minería de datos, por su parte, busca extraer información basada en patrones, con la finalidad de generar modelos que ayuden a la comprensión humana. Por otro lado, el aprendizaje automático realiza comparaciones entre los modelos generados por el proceso de minería de datos y la información nueva, esto con el fin de ser utilizado para la toma de decisiones.

Se define al aprendizaje automático como un proceso de dos fases: una fase de selección y una fase de adaptación [31]. En la primera fase se seleccionan las características más relevantes del conjunto y se comparan con las existentes a través de algún proceso de cotejo, ya que si presentan algunas diferencias significativas, se pasa a la fase dos, la cual realiza una adaptación al modelo, con la finalidad de mejorar su desempeño y obtener mejores resultados.

2.1. Tipos de Algoritmos

El aprendizaje automático consiste en diseñar algoritmos que permitan que una computadora aprenda, es decir, que emule inteligencia [1]. El tipo de aprendizaje más utilizado en IA es el inductivo, el cual genera modelos a partir de un conjunto de datos (Experiencia). Se clasifican generalmente en dos tipos principales: aprendizaje supervisado y aprendizaje no supervisado, como se observa en Fig. 2.1. La diferencia entre ellos está dada por su funcionamiento; el aprendizaje supervisado utiliza los conjuntos de datos etiquetados por un experto para generar un modelo de clasificación o predicción. En cambio el aprendizaje no supervisado, genera un modelo basado en los patrones existentes en los datos que no están etiquetados.

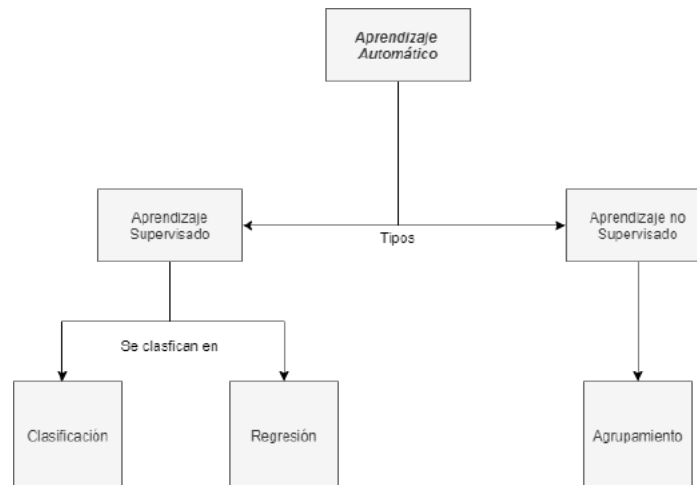


Figura 2.1: Tipos de aprendizaje.

2.1.1. Aprendizaje Supervisado

El aprendizaje Supervisado genera modelos basados en conjuntos de datos etiquetados, es decir, presenta una relación entre los datos de entrada y los datos de salida. Este tipo de aprendizaje divide el conjunto de datos en dos etapas: entrenamiento y prueba. La etapa de entrenamiento utiliza un subconjunto de los datos para generar un modelo basado en las salidas etiquetadas y utilizando el segundo subconjunto en la etapa de prueba, se valida el modelo y se comprueba la precisión. Este proceso se puede observar en la Fig. 2.2.

Basado en la salida del modelo, los algoritmos de aprendizaje supervisado se pueden dividir en 2 categorías: sí el valor de la salida es un dato continuo, es decir, un valor numérico con punto decimal, se dice que es un modelo de regresión, en cambio sí la salida es un valor nominal o discreto (categórico), se dice que se trata de un modelo de clasificación.

2.1.2. Algoritmos de aprendizaje supervisado

Los algoritmos de clasificación son los más utilizados en la práctica, de los cuales destacan los siguientes.

Árboles de decisión

En [31] se define un árbol de decisión como un método analítico que a través de una representación esquemática, muestra los posibles resultados de una serie de decisiones relacionadas. Esto se puede observar en la Fig. 2.3

Redes Neuronales

Las redes neuronales son un modelo inspirado en el funcionamiento del cerebro [1]. Están formadas por nodos llamados neuronas que se conectan entre sí. Su principal objetivo es entender, procesar y clasificar o etiquetar datos. La estructura de una red neuronal se puede observar en la Fig. 2.4

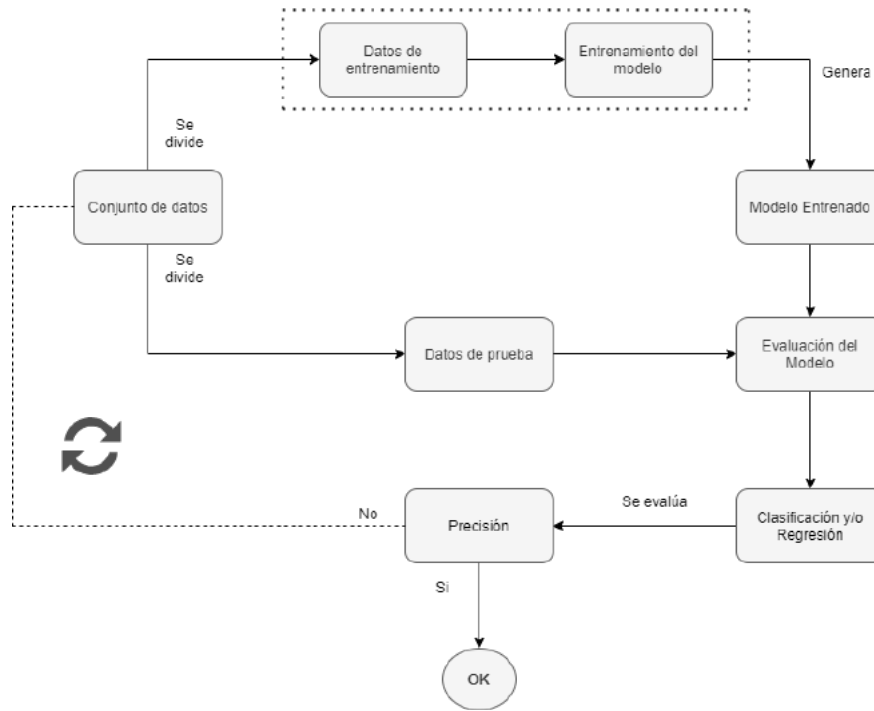


Figura 2.2: Proceso de Aprendizaje Supervisado.

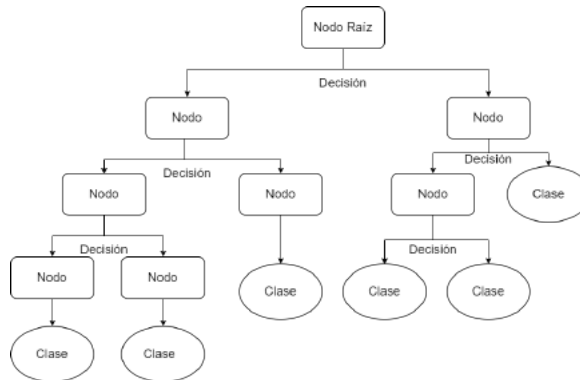


Figura 2.3: Estructura de un árbol de decisión.

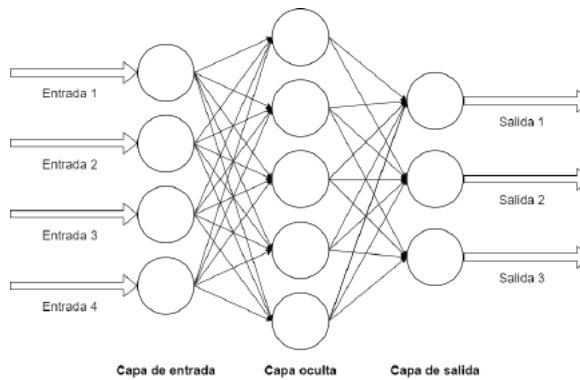


Figura 2.4: Estructura de una Red Neuronal.

Redes Bayesianas

En [44] una red bayesiana se define como una representación gráfica de dependencias para razonamiento probabilístico. En la Fig. 2.5 se muestra un ejemplo de una Red bayesiana.

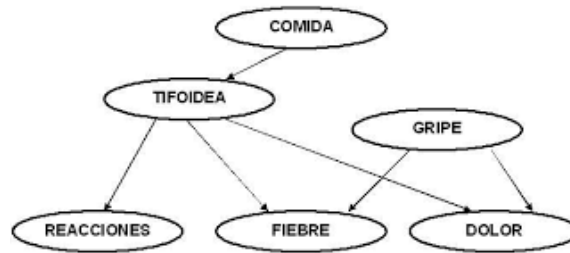


Figura 2.5: Ejemplo de una red Bayesiana.

Máquinas de Soporte Vectorial

Las máquinas de soporte vectorial (SVM, por sus siglas en inglés) etiquetan las clases y se entrenan de manera que sean capaces de predecir o clasificar los nuevos datos [41]. Las SVM representan en un eje de coordenadas los vectores de entrenamiento y en otro eje las clases etiquetadas. En la Fig. 2.6 se muestra un ejemplo de la división del sistema de coordenadas de una SVM.

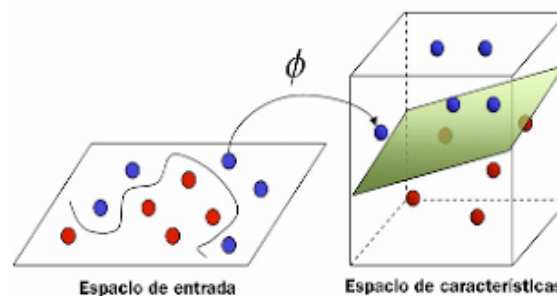


Figura 2.6: Ejemplo de una Máquina de Soporte Vectorial.

2.1.3. Aprendizaje No Supervisado

El aprendizaje no supervisado genera modelos basados en datos sin etiquetar, cuya estructura es desconocida. Su objetivo es extraer información significativa, es decir, identificar la relación existente entre las variables del conjunto de datos, tal como se muestra en la Fig. 2.7. El aprendizaje no supervisado ayuda a simplificar el análisis del conjunto de datos, por lo que son de tipo exploratorio. Este tipo de aprendizaje utiliza algoritmos de Agrupamiento o Clustering.

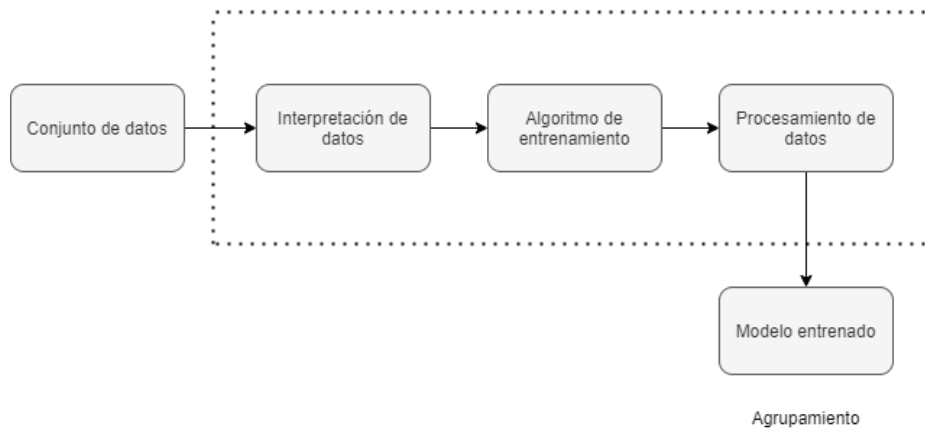


Figura 2.7: Proceso de Aprendizaje No supervisado

2.1.4. Algoritmos de agrupamiento

El agrupamiento es una técnica exploratoria de análisis de datos, que se usa para organizar información en grupos. Estos algoritmos realizan búsquedas dentro del conjunto de datos, estableciendo una variable de representación dentro del grupo. Su objetivo es obtener un conjunto de grupos de características similares. Uno de los algoritmos más importantes es el conocido como K-means (k-medias).

K-means

Este algoritmo segmenta los objetos en K particiones en donde un objeto se asigna al clúster con la media más cercana [4]. El algoritmo asigna K centros aleatoriamente, para que después cada uno de los objetos sea asignado al centro más cercano. En la Fig. 2.8 se muestra un ejemplo esquemático del proceso de agrupamiento de información.

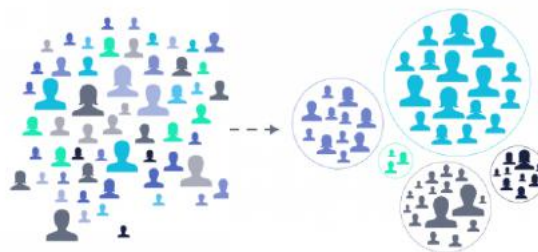


Figura 2.8: Ejemplo del algoritmo K-means.

2.2. Selección de características

La selección de características es un proceso que escoge un subconjunto de características relevantes dentro de las que se incluyen en un conjunto de datos. En [39] se mencionan

tres objetivos principales de la selección de características, los cuales se mencionan a continuación:

1. Evitar el sobre-entrenamiento y mejorar el rendimiento del modelo.
2. Generar modelos rápidos y con un bajo costo computacional.
3. Aumentar el nivel de abstracción de información que generan los datos.

2.2.1. Estrategias de selección de características

Existen diferentes formas de colaboración entre el proceso de selección de características y el proceso de clasificación. Los métodos de selección de características se dividen en tres categorías: (1) métodos de filtrado, (2) métodos de envoltorio y (3) métodos embebidos. En la Fig. 2.9 se muestra el esquema general de cada una de las tres categorías.

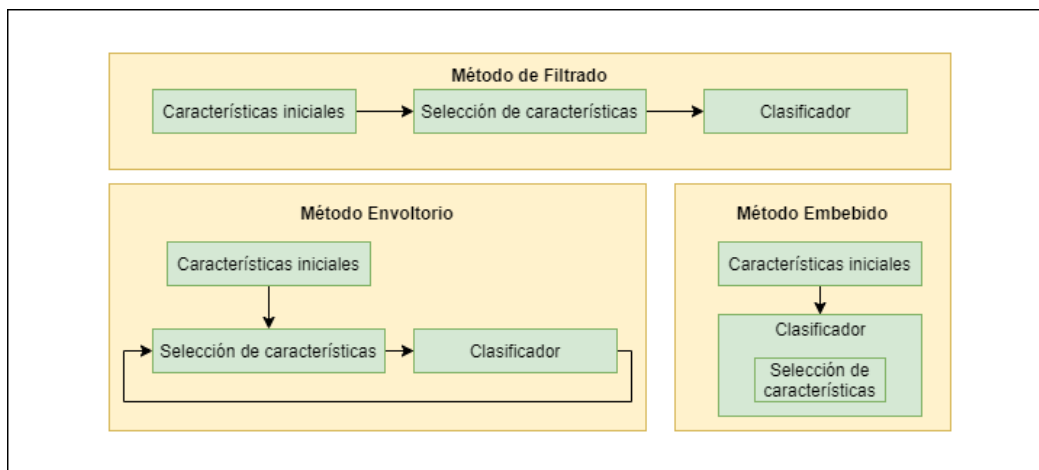


Figura 2.9: Enfoques de Selección de Características.

Método de Filtrado

Los métodos de filtrado utiliza propiedades estadísticas para "filtrar" las características que resulten menos relevantes antes de aplicar el algoritmo de aprendizaje [27]. Una forma de realizar la selección es utilizando un puntaje o score que se calcula a cada característica con base a su correlación con respecto a la variable objetivo, de esta forma se eliminan los que tengan un menor puntaje. Las ventajas y desventajas de esta estrategia de selección de características son las siguientes:

- Son computacionalmente simples y se ejecutan de manera rápida.
- Son escalables en conjunto de datos de gran tamaño.
- Ignoran las interacciones con el método de clasificación y las relaciones entre las variables.

Método Envoltorio

Los métodos de envoltorio exploran el conjunto de características para asignarles un puntaje de acuerdo a su poder predictivo, esto con base a la función de clasificación utilizada. Para explorar el espacio de los posibles subconjuntos se implementa un algoritmo de búsqueda que envuelve al método de clasificación, el cual es utilizado para evaluar la calidad predictiva de los subconjuntos. Esta estrategia utiliza frecuentemente la Selección secuencial hacia Adelante (SFS, por sus siglas en inglés) y la Eliminación Secuencial hacia atrás (SBE, por sus siglas en inglés). Las ventajas y desventajas de esta estrategia de selección de características son las siguientes:

- Consideran las interacción entre las características.
- Tienen la capacidad de reducir la redundancia que existe con las características.
- El costo computacional puede ser alto.

Método Embebido

Los métodos embebidos buscan un subconjunto utilizando un algoritmo de aprendizaje específico, por ejemplo los árboles de decisión. Existen dos enfoques dentro de los métodos embebidos, en el primero, el clasificador proporciona información que guía la exploración de los subconjuntos y uno de los métodos utilizados para esto es el de Eliminación Recursiva de Atributos (RFE, por sus siglas en inglés), el cual inicia considerando todas las características y recursivamente va eliminando algunas de ellas. Para realizar esta eliminación se entrena un clasificador y la característica eliminada será aquella que tenga menor influencia en su función objetivo. Los métodos penalizados constituyen el segundo enfoque de los métodos embebidos, en ellos la selección de características es integrada en el proceso de construcción del clasificador agregando a la función objetivo una función de penalización que toma en cuenta el número de características.

En la Tabla 2.1 se muestra una comparación de las tres estrategias de selección de características.

Tabla 2.1: Comparación de las 3 estrategias de selección de características.

Método	Precisión de Clasificación	Costo Computacional	Generalidad
Filtrado	Bajo	Bajo	Alto
Envoltorio	Alto	Alto	Bajo
Embebido	Medio	Medio	Medio

2.2.2. Algoritmos tradicionales

Existen diversos algoritmos para la selección de características, los cuales se clasifican con base a la estrategia de selección. Dos de los algoritmos más utilizados son el de Selección Secuencial hacia Adelante y el de Eliminación Secuencial hacia Atrás. Ambos algoritmos basados en la estrategia de selección por método de envoltorio [5].

Selección secuencial hacia Adelante (SFS)

Fue propuesto por Whitney en 1971 [51] y opera de manera ascendente, es decir, inicia con un conjunto vacío de características y en cada iteración la característica que maximiza la función es seleccionada y agregada al conjunto actual. Una vez seleccionada la característica no puede ser eliminada por los pasos subsecuentes. Este proceso se repite hasta que se selecciona el número de características predefinidas.

Eliminación Secuencial hacia Atrás (SBS)

Fue propuesto por Maril y Green en 1963 [29] y opera de manera descendente, a diferencia del algoritmo SFS, este algoritmo inicia considerando todas las características del conjunto y en cada iteración una característica es eliminada con base en un criterio de evaluación. Este proceso se repite hasta que el número de características restantes sea igual al número de características predefinido.

2.2.3. Ventajas y desventajas de las estrategias de selección de características

De los distintos métodos que existen basados en las estrategias para la selección de características, se puede observar que tienen ciertas características que favorecen y desfavorecen a los algoritmos de clasificación. Por ello se muestran algunas ventajas y desventajas de utilizar los algoritmos tradicionales para la selección de características.

Ventajas

Los métodos de filtrado tienen un costo computacional muy bajo, son simples y rápidos, lo cual generalmente evita el sobre ajuste.

Los métodos de envoltorio capacitan un modelo para identificar las características relevantes. Este tipo de modelos suelen encontrar un subconjunto competitivo de características.

Los métodos embebidos realizan la selección de características en el proceso de entrenamiento del algoritmo de aprendizaje, de este modo, este tipo de método localiza el mejor subconjunto con un costo computacional menor al de los métodos de envoltorio.

Desventajas

Los métodos de filtrado, al no contar con el algoritmo de aprendizaje, pueden fallar en encontrar el mejor subconjunto de características para el conjunto de datos de entrenamiento.

Los métodos de envoltorio son costosos computacionalmente y al utilizar un algoritmo de aprendizaje automático pueden causar un sobre ajuste, lo cual afecta el desempeño del algoritmo de clasificación en la fase de prueba.

Capítulo 3

Cómputo Evolutivo

El cómputo evolutivo es un área de investigación dentro de la Inteligencia Artificial y se enfoca en el estudio de una serie de metaheurísticas estocásticas denominadas “Algoritmos Evolutivos”, inspiradas en la teoría de evolución de las especies de Charles Darwin, la cual sostiene que los individuos más aptos a su ambiente son los que tienen una posibilidad mayor de sobrevivir. La metáfora fundamental del cómputo evolutivo relaciona la evolución natural con un estilo particular de resolución de problemas, el de prueba y error [13]. Esta relación se puede observar en la Fig. 3.1.

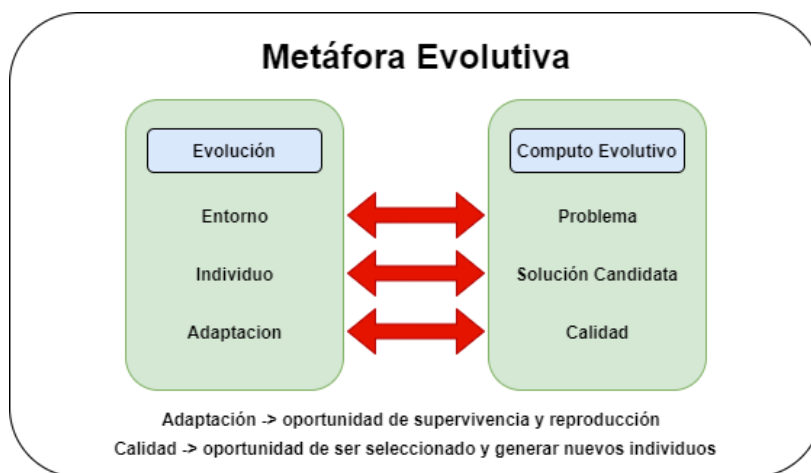


Figura 3.1: Metáfora Evolutiva.

La evolución natural, de manera muy general, establece la existencia de un ambiente determinado donde existe una población de individuos que luchan por reproducirse y sobrevivir. La probabilidad de reproducción y supervivencia de un individuo está determinada por la afinidad de sus características con respecto al ambiente. Aplicando esta metáfora al proceso de resolución de problemas de búsqueda, se tiene una población formada por una colección de posibles soluciones. Su calidad determina la posibilidad de que se mantengan para ser base en la construcción de nuevas soluciones candidatas o en otro caso ser eliminadas.

Metaheurísticas

Una metaheurística es un procedimiento que aplica una regla o un conjunto de reglas para explorar el espacio de búsqueda de manera eficiente. Suelen considerarse como técnicas de búsqueda y optimización de uso general ya que suelen requerir de poca información específica del problema y son útiles cuando el espacio de búsqueda es extenso [53].

3.1. Algoritmos evolutivos

Los Algoritmos Evolutivos (AEs) son metaheurísticas estocásticas, esto debido a su comportamiento, el cual se basa en el uso de números aleatorios, por lo que no generan el mismo resultado cada vez que son ejecutados. Los AEs agrupan a aquellos basados en fenómenos relacionados con la evolución de las especies y la supervivencia del más apto. El modelo general para la solución de AEs se da mediante los siguientes 4 pasos.

1. Generar una población inicial aleatoriamente, la cual representara un conjunto de posibles soluciones.
2. Seleccionar los individuos mas aptos, esto tomando como criterio de selección el valor de su función de calidad.
3. Aplicar operadores de variación (mutación y cruza) para generar nuevos individuos.
4. Reemplazar la población con los individuos mas aptos, esto hasta cumplir con una condición de paro establecida.

Este modelo se puede observar en la Figura 3.2.

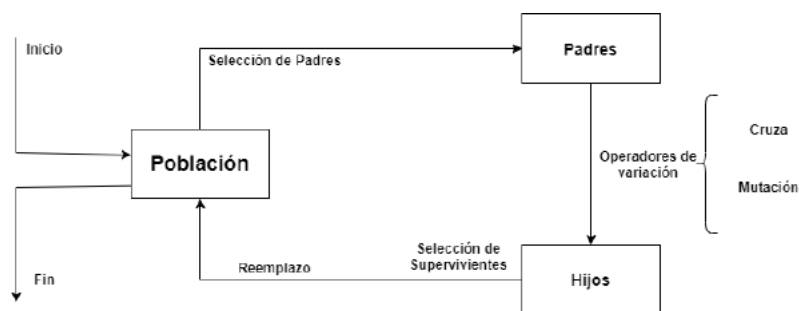


Figura 3.2: Modelo General de un Algoritmo Evolutivo.

Los algoritmos evolutivos tienen componentes que están basados en el proceso de selección natural, los cuales se definen a continuación:

1. **Representación de soluciones.** Puede ser de nivel fenotipo (soluciones reales) o genotipo (soluciones representadas mediante un código).
2. **Función de aptitud o calidad.** Constituye la base de la selección, es una función que asigna una medida de calidad a los genotipos.

3. **Población de soluciones.** Grupo de soluciones potenciales del problema.
4. **Mecanismos de selección de padres.** Este mecanismo distingue de entre los individuos cual es el mejor, considerando su calidad. Generalmente tiene elementos probabilísticos.
5. **Operadores de variación.** El papel de los operadores de variación es crear nuevos individuos a partir de los ya existentes.
6. **Mecanismo de reemplazo.** Distingue de entre los individuos de la población los mejores basado en su calidad, esto para mantener la población con un tamaño fijo.

Los paradigmas principales de los algoritmos evolutivos son:

- Algoritmos Genéticos (AG)
- Programación Genética (PG)
- Estrategias Evolutivas (EE)
- Programación Evolutiva (PE)
- Evolución Diferencial (ED)

3.1.1. Algoritmos Genéticos (AG)

Los AG son algoritmos de búsqueda, basados en los mecanismos de selección natural, los cuales pueden representar sus soluciones a un nivel genotipo o fenotipo, dependiendo del tipo de problema que se quiera resolver. Los AG's contienen los elementos de los AE's que son: selección de padres, recombinación o cruza, mutación y reemplazo. La forma de trabajar se muestra en el Algoritmo 1.

Algoritmo 1: Algoritmo Genético Simple

```

Iniciar Población;
mientras El criterio de paro no sea satisfecho hacer
    | Evaluar población;
    | Seleccionar padres;
    | Aplicar recombinación de hijos;
    | Aplicar mutación de hijos;
    | Realizar reemplazo;
fin

```

3.1.2. Programación Genética (PG)

La programación genética es muy similar a los algoritmos genéticos, su diferencia radica en que los individuos de la PG son estructuras en forma de árbol en lugar de cadena de bits como los AG, lo cual hace que su representación sea vista como estructuras no lineales.

3.1.3. Estrategias Evolutivas (EE)

Las EE simulan la evolución a nivel de los individuos, y aunque existe un operador de cruza, este operador no es de tanta importancia para el algoritmo. El operador principal para este algoritmo es la mutación. La representación en las EE es a nivel fenotipo y el proceso de selección de sobrevivientes es determinístico y extintivo, es decir, que el peor de los individuos no sobrevive. El proceso de selección de padres se lleva a cabo seleccionando de entre la población de manera aleatoria cada uno de ellos, debido a esto, cada padre puede ser seleccionado más de una vez. La manera en que trabajan las EE se muestra en el Algoritmo 2.

Algoritmo 2: Algoritmo de Estrategias Evolutivas

```

Generar aleatoriamente una población inicial de soluciones;
Calcular la aptitud de la población inicial;
mientras El criterio de paro no sea satisfecho hacer
    Seleccionar dos o mas padres de manera aleatoria;
    Aplicar cruza para crear hijos;
    Aplicar el operador de mutación a los hijos;
    Evaluar cada hijo;
    Seleccionar los mejores individuos para la siguiente generación;
fin

```

3.1.4. Programación Evolutiva (PE)

Fue propuesta por Fogel en 1960 [15], donde remarca las relaciones de herencia y el comportamiento entre padres e hijos. La PE simula la evolución a nivel de las especies, por lo tanto, no hay un operador de cruza, el principal y único operador es la mutación. La selección de padres es determinista, ya que cada padre genera un hijo mediante al operador de mutación. La técnica de selección de sobrevivientes se basa en torneos estocásticos donde compiten los padres e hijos y sobreviven aquellos que tengan en los torneos un mayor puntaje, éste se determina comparando la aptitud de cada individuo.

Algoritmo 3: Algoritmo de Programación Evolutiva

```

Generar aleatoriamente una población inicial de soluciones;
Calcular la aptitud de la población inicial;
mientras El criterio de paro no sea satisfecho hacer
    Aplicar la mutación a toda la población para crear hijos.;
    Evaluar cada hijo;
    Seleccionar mediante torneos estocásticos los mejores individuos para la
    siguiente generación;
fin

```

La manera en que trabajan la PE se muestra en el Algoritmo 3.

3.1.5. Evolución Diferencial (ED)

La ED es una estrategia de búsqueda poblacional, muy similar a un algoritmo evolutivo estándar y fue propuesta por Storn y Price en 1995 [43]. La principal diferencia son sus operadores de cruce y mutación. El operador de mutación no está basado en una función de distribución de probabilidad predefinida, sino que la distribución se basa en la distribución actual de las soluciones (llamadas vectores) en la población. La Evolución Diferencial tiene una representación a nivel fenotipo y el operador de mutación consiste en una diferencia aritmética entre pares de vectores seleccionados aleatoriamente. Cada vector padre denominado “target” genera un vector hijo denominado “trial”. Para realizar este proceso se seleccionan 3 vectores de forma aleatoria dentro de la población, por lo que hace que la selección de padres sea determinista.

Algoritmo 4: Algoritmo de Evolución Diferencial

```

Generar aleatoriamente una población inicial de vectores;
Calcular la aptitud de la población inicial;
repetir
  Por cada target
    Crear el vector ruido mediante mutación;
    Crear el vector trial mediante recombinación;
  /* Comparar el vector trial con el vector target */
  Sí el vector trial es mejor que el vector target, el trial tomará el lugar del
  target.
hasta que El criterio de paro sea satisfecho;

```

La selección de supervivientes se lleva a cabo de forma determinística, esto debido a que si el vector trial es mejor que el target, este último será reemplazado en la siguiente generación, es decir, el vector más apto permanecerá en la población. La manera en que trabaja el algoritmo de ED se muestra en el Algoritmo 4.

3.2. Optimización Multiobjetivo

Un problema de optimización consiste en maximizar o minimizar una función. De forma general, la optimización implica descubrir el valor óptimo de una función en un dominio definido.

Problema Multiobjetivo

Un problema puede ser llamado problema multiobjetivo cuando el valor óptimo en dos o más de sus funciones suelen estar en conflicto.

Un problema multiobjetivo puede definirse como:

Un vector de variables de decisión que satisfagan un cierto conjunto de restricciones y optimice un conjunto de funciones objetivo. Estas funciones forman una descripción matemática de los criterios de desempeño que suelen estar en conflicto unos con otros y que se suelen medir en unidades diferentes.

(Osyczka, 1984 [33])

Existen tres tipos de situaciones que pueden presentarse en un problema multiobjetivo:

- Minimizar todas las funciones.
- Maximizar todas las funciones.
- Minimizar algunas funciones y maximizar otras.

Sin embargo, por cuestiones de simplicidad, generalmente todas las funciones se convierten ya sea en un problema de maximización o a uno de minimización. Una de las formas para realizar esta conversión es utilizando una identidad para todas las funciones a maximizarse y de esta manera poder generar un problema de minimización. Esta identidad se describe a continuación:

$$\max f_i(\vec{X}) = \min(-f_i(\vec{X}))$$

Función Objetivo

Para determinar que tan “buena” es una solución, es necesario contar con algún criterio para evaluarla, estos criterios suelen denotar aspectos cuantificables con alguna métrica o escala contra las que pueden medirse las alternativas disponibles. Estos criterios se expresan como funciones computables de las variables de decisión a las que se denomina **funciones objetivo**, estas funciones pueden ser conmensurables o no conmensurables. Es decir, medidas en las mismas unidades o en unidades distintas [6].

Óptimo de Pareto

Al tener varias funciones objetivo, la noción de “óptimo” cambia, ya que se busca obtener un compromiso (trade offs, en inglés) entre las soluciones, en lugar de una solución única como en un problema de optimización mono-objetivo.

Un vector de variables de decisión es considerado un óptimo de Pareto si no existe ningún vector factible de variables de decisión que decremente alguna función objetivo sin causar un incremento simultáneo en al menos otra función objetivo. Sin embargo, casi siempre se produce no una solución única sino un conjunto de ellas a las que se les denomina como conjunto de óptimos de Pareto. Los vectores correspondientes a las soluciones incluidas en el conjunto de óptimos de Pareto son llamados **no dominados** [6].

Dominancia de Pareto

La dominancia de Pareto implica que, para que una solución domine a otra, no debe ser peor en ninguno de los objetivos, y debe ser estrictamente mejor en al menos uno de ellos.

Tabla 3.1: Relaciones de preferencia para comparación de vectores objetivo.

Relación	Símbolo	Interpretación
domina estrictamente	$Z^1 \prec\prec Z^2$	Z^1 es mejor que Z^2 en todos los objetivos
domina	$Z^1 \prec Z^2$	Z^1 no es peor que Z^2 en todos los objetivos y es mejor en al menos un objetivo
domina débilmente	$Z^1 \preceq Z^2$	Z^1 no es peor que Z^2 en todos los objetivos
incomparable	$Z^1 \parallel Z^2$	no se cumple que $Z^1 \preceq Z^2$ ni que $Z^2 \preceq Z^1$
indiferente	$Z^1 \sim Z^2$	Z^1 tiene el mismo valor que Z^2

Estas relaciones de preferencia entre vectores objetivo se muestran en la tabla 3.1. Debido a esto, al comparar dos soluciones A y B usando la relación de preferencia descrita con anterioridad, puede haber 3 posibles resultados:

- A domina a B.
- A es dominado por B.
- A y B no se dominan entre sí.

La relación de dominancia tiene ciertas propiedades las cuales se describen a continuación:

- La relación de dominancia no es reflexiva, puesto que cualquier relación p no se domina a sí misma.
- La relación de dominancia no es simétrica. Es decir, si p domina a q, entonces q no domina a p.
- Puesto que la relación de dominancia no es simétrica, tampoco puede ser asimétrica.
- La relación de dominancia es transitiva. Es decir, que si $p \preceq q$ y $q \preceq r$, entonces $p \preceq r$.

Otra propiedad interesante que posee la relación de dominancia es que si la solución p no domina a la solución q, no implica que q domine a p.

Debido a estas propiedades la relación de dominancia califica como una relación de orden. Sin embargo, dado que la relación de dominancia no es reflexiva, es un orden parcial estricto.

Frente de Pareto

Se denomina frente de Pareto al conjunto de óptimos de Pareto los cuales forman un compromiso en el espacio de búsqueda y el cual puede observarse en la figura 3.3.

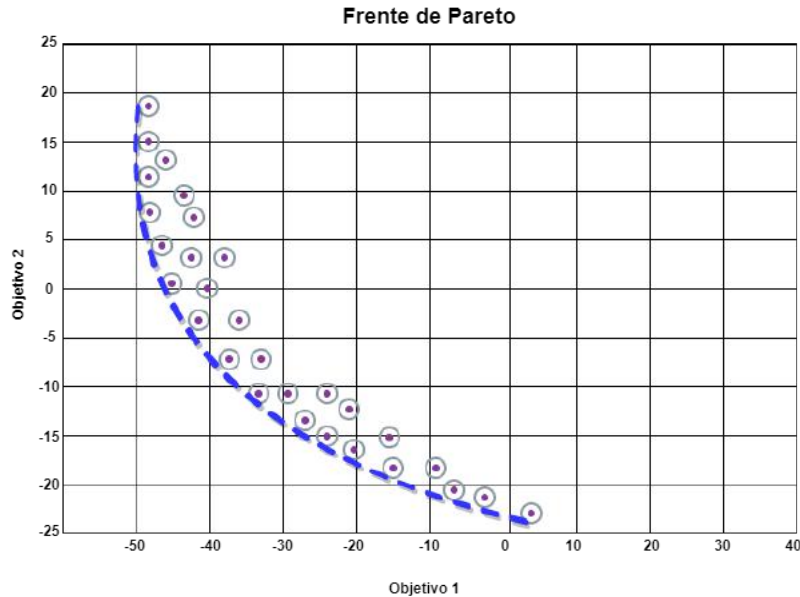


Figura 3.3: Frente de Pareto

El procedimiento normal para generar el llamado Frente de Pareto es calculando todas las soluciones factibles dentro del espacio de búsqueda y obteniendo sus valores correspondientes a las funciones objetivo, de esta manera cuando se tiene un número suficiente de valores, es posible determinar la no dominancia entre ellos, esta no dominancia puede ser débil o fuerte.

- La No Dominancia débil sucede cuando una solución es mejor o igual que otra del conjunto en todos sus valores.
- La No Dominancia Fuerte sucede cuando una solución es mejor que otra del conjunto en todos sus valores.

3.2.1. Algoritmos Evolutivos en la Optimización Multiobjetivo

El uso de AEs para resolver problemas de esta naturaleza ha sido motivado principalmente por la naturaleza poblacional de los algoritmos evolutivos, los cuales permiten la generación de varios elementos del conjunto de óptimos de Pareto en una sola ejecución [8]. El potencial de los AEs para resolver problemas de optimización multiobjetivo se remonta a Richard Rosenberg en 1967 [37]. Sin embargo, aunque Rosenberg plantea por primera vez la posibilidad de usar algoritmos genéticos para resolver un problema multiobjetivo, no presentó una implementación de dicho algoritmo. El primero en desarrollar una implementación de un algoritmo evolutivo multiobjetivo (MOEA, por sus siglas en inglés) fue John David Schaffer en 1984 [42]. El algoritmo de Schaffer, llamado Vector Evaluated Genetic Algorithm (VEGA) fue publicado en la First International Conference on Genetic Algorithm en 1985.

Desde entonces se han desarrollado diversos MOEA's (Multi-Objective Evolutionary Algorithms), los cuales se dividen en dos tipos, No elitistas y Elitistas y se en listan a continuación:

Algoritmos No Elitistas.

- VEGA (Vector Evaluated Genetic Algorithm).
- MOGA (Multi Objective Genetic Algorithm).
- NSGA (Nondominated Sorting Genetic Algorithm).
- NPGA (Niched Pareto Genetic Algorithm) y NPGA II.

Algoritmos Elitistas.

En optimización mono-objetivo, el elitismo es un operador mediante el cual la mejor solución en la población pasa intacta a la siguiente generación, sin ser afectada por la cruce o la mutación.

En la optimización multiobjetivo, el elitismo funciona de manera similar, pero en este caso, requiere retener a las soluciones no dominadas que hayan sido generadas por el MOEA, sin embargo, como es impráctico retener todas las soluciones, suele definirse un número máximo de soluciones que serán retenidas. Esto es particularmente importante en los MOEA's en los que las soluciones elitistas se incorporan en el mecanismo de selección.

Se requiere el uso del elitismo para garantizar convergencia de un MOEA al verdadero frente de Pareto de un problema multiobjetivo [38]. Algunos de los algoritmos que se basan en el elitismo son los siguientes:

- SPEA (Strength Pareto Evolutionary Algorithm) y SPEA 2
- NSGA II (Nondominated Sorting Genetic Algorithm II).
- PAES (Pareto Archived Evolution Strategy).
- PESA (Pareto Envelope Selection Algorithm) y PESA II.
- MOMGA (Multi Objective Messy Genetic Algorithm).

Algoritmos Recientes.

- MOEA/D (Multi Objective Evolutionary Algorithm on Decomposition).
- IBEA (Indicator Based Evolutionary Algorithm).
- HyPE (Hypervolume estimation algorithm for multi objective optimization).
- NSGA III (Nondominated Sorting Genetic Algorithm III).

Diversidad

Todos los algoritmos evolutivos tienden a converger a una solución única debido al ruido estocástico, por lo que mantener la diversidad en la población es un tema fundamental. Este problema ha sido estudiado y se han propuesto diversos mecanismos para preservar la diversidad en la población. Algunos de ellos son:

- **Crowding.** Propuesto por Holland en 1975, era capaz de identificar situaciones en las cuales más individuos dominan un cierto nicho ecológico.
- **Fitness sharing.** Propuesto por Goldberg y Richardson [16], este método subdivide la población en varias subpoblaciones con base en la similitud de los individuos.
- **Crowding Determinístico.** Propuesto por Mahfound en 1992 [25]. donde se busca minimizar los errores de reemplazo.
- **Crowding probabilístico.** Propuesto por Mengshoel y Goldberg en 2008 [30], esta técnica adopta una función de aceptación probabilística con la cual mostraron que produce subpoblaciones más estables que las obtenidas con el Crowding determinístico.
- **RTS (Restricted Tournament Selection).** Propuesta por Harik en 1995 [20]. Este algoritmo selecciona dos elementos de la población, los combina y muta, después recombina y toma una muestra de los individuos de la población seleccionados mediante el factor de Crowding. De esta manera cada hijo compete con el individuo más cercano en la muestra y los ganadores son insertados en la población.
- **Clearing.** Propuesta por Pétrowski en 1996 [34], es una técnica similar a la fitness sharing, pero se basa en la noción de la existencia de una cantidad limitada de recursos en el ambiente.

También el uso de restricciones a la cruce es una forma con la cual se mantiene la diversidad en los algoritmos evolutivos. Existen muchas otras técnicas para mantener la diversidad pero solo un pequeño porcentaje de ellas se han utilizado en optimización evolutiva multiobjetivo. En optimización multiobjetivo es muy común aplicarla en el espacio fenotípico, ya que lo que normalmente se busca es una distribución uniforme del Frente de Pareto.

3.2.2. Indicadores de Desempeño

En el año 2000 se indicó que al medir el desempeño de un MOEA, existen tres metas deseables:

1. Minimizar la distancia de nuestra aproximación al frente óptimo de Pareto.
2. Obtener una buena distribución de soluciones. Normalmente, lo que se busca es una distribución uniforme de soluciones.
3. Maximizar la extensión de nuestra aproximación del frente de Pareto.

Sin embargo, no siempre estas metas son deseables o relevantes. Normalmente la primera meta es la de mayor relevancia cuando se mide el desempeño de un MOEA. Algo muy interesante con respecto a las tres metas es que no es posible medirlas con un solo indicador de desempeño, si no que requerirá una combinación de indicadores.

Relaciones de Preferencia

Para expresar la relación entre dos conjuntos de vectores objetivo no dominados A y B, Hansen y Jaszkiwicz definieron un número de outperformance relations [19], las cuales se definen a continuación:

Sobre-desempeño débil (O_W): Se dice que el conjunto A supera débilmente al conjunto B si para cada punto en B existe un punto en A que es igual o domina a B y hay al menos un punto en A que no está contenido en B.

$$AO_W B \iff ND(A \cup B) = A$$

$$A \neq B$$

Se puede obtener una aproximación que supere débilmente a B agregando a B un nuevo punto que no esté dominado con respecto a todos los puntos en B.

Sobre-desempeño fuerte (O_S): Se dice que el conjunto A supera fuertemente al conjunto B si para cada punto en B existe un punto en A que es igual o domina a B y al menos un punto en B es dominado por un punto en A.

$$AO_S B \iff ND(A \cup B) = A$$

$$B \setminus ND(A \cup B) \neq \emptyset$$

Se puede obtener una aproximación que supera fuertemente a B agregando a B un nuevo punto que domine al menos un punto en B.

Sobre-desempeño completo (O_C): Se dice que en conjunto A supera completamente al conjunto B si para cada punto en B existe un punto en A que lo domina.

$$AO_C B \iff ND(A \cup B) = A$$

$$B \cap ND(A \cup B) \neq \emptyset$$

Se puede obtener una aproximación que supere completamente a B agregando a B uno o varios puntos que dominen a todos los puntos del conjunto B.

Dado que O_C es mejor que O_S y esta a su vez es mejor que O_W , se dice que el **Sobre-desempeño completo** es la relación más fuerte y el **Sobre-desempeño débil** es la relación más débil. Todas estas relaciones muestran las formas en que se pueden relacionar las aproximaciones al frente de Pareto (PF_{know}) puesto que son compatibles con la dominancia de Pareto.

Indicadores de Desempeño

Existen diversos indicadores de desempeño de los cuales algunos se explicarán brevemente así como su Pareto compatibilidad, ventajas y desventajas.

La **Pareto Compatibilidad** se refiere a la compatibilidad con las relaciones de outperformance O_C , O_S y O_W . Entre menos compatible sea una medida, menos confiable resultará, dado que proporcionaría valores para conjuntos no dominados que no reflejan de manera precisa su valor en relación con el frente de Pareto.

Hipervolumen o metrica S

El hipervolumen fue propuesto por Zitzler en 1999 y se define como el área de cobertura de las aproximaciones al frente de Pareto con respecto al espacio de las funciones objetivo. Matemáticamente, para cada solución $i \in Q$, un hipercubo v_i se construye con un punto de referencia W y la solución i como las esquinas diagonales del hipercubo. El punto de referencia se define a partir de los peores valores de la función objetivo. A partir de esto se encuentra la unión de todos los hipercubos y su hipervolumen se calcula matemáticamente de la siguiente forma:

$$HV = \text{volume}\left(\bigcup_{i=1}^{|Q|} (v_i)\right)$$

Donde Q es un conjunto de soluciones no dominadas, i es una solución de Q y v_i es un hipercubo de la solución i . Este es un indicador independiente, sin embargo, necesita elegir un punto de referencia de forma que induzca un ordenamiento completo y no es cardinal.

Pareto Compatibilidad: Compatible con O_W si el punto de referencia se establece de tal forma que todos los conjuntos no dominados sean evaluados como positivos. Normalmente, los peores valores de las funciones objetivo se usan para definir este punto de referencia.

Ventajas: Compatible con las relaciones de outperformance, independiente, puede diferenciar entre distintos grados de outperformance.

Desventajas: Requiere definir una cota superior de las regiones dentro de las cuales se encontrarán todos los puntos factibles. Tiene un costo computacional elevado, lo que hace imposible utilizarlo para problemas con muchas funciones objetivo.

Error Ratio

Este indicador de desempeño fue propuesto por Van Veldhuizen en 1999 y matemáticamente se define como:

$$ER = \frac{\sum_{i=1}^n e_i}{n}$$

donde n es el número de vectores en la aproximación al frente de Pareto. $e_i = 0$ si el vector i está en el verdadero frente de Pareto y $e_i = 1$ si el vector i no se encuentra en el verdadero frente de Pareto.

Valores pequeños de ER representan mejores aproximaciones. ER es la proporción de puntos no Pareto óptimos en nuestra aproximación. Es un indicador de referencia que usa el verdadero frente de Pareto como su conjunto de referencia.

Pareto Compatibilidad: Solo es débilmente compatible con O_C . No es débilmente compatible con O_S o O_W . Es decir, si un algoritmo encuentra dos vectores no dominados, de los cuales uno está en el verdadero frente de Pareto y el otro está lejos de él, su error ratio será de 0.5.

Ventajas: Es fácil de entender y fácil de calcular. Es independiente de la escala y para problemas de prueba puede usarse como un mecanismo rápido y crudo de medir el progreso hacia el verdadero frente de Pareto.

Desventajas: Se requiere conocer el verdadero frente de Pareto. Es incompatible con las relaciones de outperformance.

Distancia Generacional (GD)

La Distancia Generacional (GD) fue propuesta por Veldhuizen en 1999 y reporta qué tan lejos se encuentra, en promedio nuestra aproximación al frente de Pareto con el verdadero frente de Pareto. Matemáticamente se define de la forma siguiente:

$$GD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n}$$

donde n es el número de vectores en la aproximación al frente de Pareto y d_i es la distancia Euclidiana en el espacio de las funciones objetivo y el miembro más cercano en el verdadero frente de Pareto. Cuando $GD=0$, nuestra aproximación es igual al verdadero frente de Pareto. Los valores menores de GD representan las mejores aproximaciones al verdadero frente de Pareto. La Distancia Generacional es una medida de referencia que usa al verdadero frente de Pareto como referencia. Induce el ordenamiento total y no es cardinal.

Pareto Compatibilidad: No es débilmente compatible con O_W , pero es compatible con O_S . Es decir, el valor de GD favorece a un vector cercano al verdadero frente de Pareto sobre un conjunto que contenga a ese vector más otros, siempre y cuando los otros no estén más cerca. No exhibe relatividad débil, debido a que cualquier subconjunto del verdadero frente de Pareto tiene una GD óptima.

Ventajas: Para un tamaño constante del conjunto no dominado, GD es compatible con O_S . No es costosa computacionalmente.

Desventajas: Ya que no es compatible con O_W no puede ser utilizada de manera confiable para conjuntos no dominados que cambian de cardinalidad. No puede diferenciar de manera confiable entre niveles de outperformance. Se requiere conocer el verdadero frente de Pareto del problema.

Distancia Generacional Invertida (IGD)

La Distancia Generacional Invertida reporta qué tan lejos está, en promedio el verdadero frente de Pareto de nuestra aproximación al frente de Pareto. Es una variante de la Distancia Generacional propuesta por Coello y Cruz en 2005 [7] y esta definida matemáticamente como:

$$IGD \triangleq \frac{(\sum_{i=1}^n d_i^p)^{\frac{1}{p}}}{n}$$

donde n es el número de vectores en el verdadero frente de Pareto, $p = 2$, y d_i es la distancia Euclidiana en el espacio de las funciones objetivo entre cada vector del verdadero frente de Pareto y el miembro más cercano en la aproximación al frente de Pareto. Cuando $IGD=0$, nuestra aproximación es igual al verdadero frente de Pareto. Los valores menores de IGD representan las mejores aproximaciones al verdadero frente de Pareto.

Aunque al medir la distancia promedio desde el verdadero frente de Pareto a nuestra aproximación al frente de Pareto logra resolver varios problemas de la Distancia Generacional original, este indicador no es Pareto Compatible. Sin embargo, IGD puede medir no solo la convergencia, si no también la distribución de las soluciones a lo largo del frente de Pareto.

Debido a su simplicidad, es muy popular en problemas de optimización con muchas funciones objetivo y presenta un bajo costo computacional.

Espaciado de Schott

Propuesto por Schott en 1995, este indicador mide la varianza en la distancia de los vectores vecinos en la aproximación al frente de Pareto. Está definido matemáticamente como:

$$SS = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2}$$

y

$$d_i = \min(|f_1^i(\vec{x}) - f_1^j(\vec{x})| + |f_2^i(\vec{x}) - f_2^j(\vec{x})|)$$

donde $i, j = 1 \dots n$, \bar{d} es la media de todas las d_i , y n es el número de vectores en la aproximación al frente de Pareto.

Este indicador no requiere conocer el verdadero frente de Pareto, es una medida independiente, induce el ordenamiento completo y es cardinal.

Pareto Compatibilidad: No es ni siquiera compatible débilmente con O_W . No exhibe ni monotonía ni relatividad, puesto que el verdadero frente de Pareto puede ser no uniforme.

Ventajas: Es usado en combinación con otros indicadores ya que proporciona información sobre la distribución de vectores obtenidos. Tiene un costo computacionalmente bajo. Puede generalizarse a más de dos objetivos extendiendo la definición de d_i .

Desventajas: La definición de Schott de d_i no especifica el uso de distancias normalizadas, razón por la cual puede ser problemático. Debido a su incompatibilidad con las relaciones de outperformance es poco confiable.

Espaciado de Deb

El espaciado de Deb fue utilizado por Deb en el año 2000 y es similar al espaciado de Schott. Matemáticamente esta definido como:

$$\Delta = \sum_{i=1}^n \frac{|d_i - \bar{d}|}{n}$$

donde n es el número de vectores en la aproximación al frente de Pareto, d_i es la distancia Euclidiana entre dos vectores consecutivos en nuestra aproximación al frente de Pareto y \bar{d} es el promedio de estas distancias.

Este indicador tiene como propósito medir que tan distribuidos uniformemente están los puntos de nuestra aproximación.

Pareto Compatibilidad: No es débilmente compatible con ninguna relación de out-performance. Es muy común que el verdadero frente de Pareto tenga una distribución no uniforme de puntos.

Ventajas: Es usado en combinación con otros indicadores ya que proporciona información sobre la distribución de vectores obtenidos. Tiene un costo computacionalmente bajo.

Desventajas: Solo es funcional para problemas con dos funciones objetivo y debido a que no tiene compatibilidad con las relaciones outperformance no se considera confiable.

comparación de cobertura relativa de dos conjuntos

Propuesto por Zitzler en el año 2000 como un indicador binario, se define como el mapeo del par ordenado (A, B) al intervalo $[0, 1]$ definido matemáticamente como:

$$\mathcal{C}(A, B) = \frac{|\{b \in B | \exists a \in A : a \preceq b\}|}{|B|}$$

Cuando se obtiene un valor $\mathcal{C}(A, B) = 1$, significa que todos los vectores de decisión en B son débilmente dominados por A y cuando se obtiene el valor $\mathcal{C}(A, B) = 0$, significa que ninguno de los puntos en B es débilmente dominado por A.

Este indicador tiene como propósito comparar dos conjuntos no dominados en términos de su calidad. Se considera una métrica comparativa directa ya que proporciona un solo valor de mérito. Es difícil establecer si este indicador induce un ordenamiento completo debido a que no resulta claro cómo debe interpretarse el par de valores en \mathcal{C} que produce.

Pareto Compatibilidad: Debido a que este indicador produce dos valores cuando se comparan los conjuntos A y B, $\mathcal{C}(A, B)$ y $\mathcal{C}(B, A)$, es difícil analizar si es compatible o no con las relaciones de outperformance, sin embargo, utilizando conjuntos de referencia este indicador es débilmente compatible con la relación O_W .

Ventajas: Tiene un costo computacional bajo en comparación con hipervolumen. Es compatible con O_S y no requiere conocimiento del conjunto eficiente ni de los rasgos del conjunto factible.

Desventajas: Su incompatibilidad con O_W . Se considera no confiable si dos conjuntos son de cardinalidad diferente o si la distribución de los conjuntos no es uniforme. No puede determinar el grado de outperformance si un conjunto supera completamente a otro.

Zitzler en 2003 [58] realizó un análisis de los indicadores de desempeño y uno de los resultados obtenidos fue que el hipervolumen es el único indicador unario que es Pareto Compatible .

Capítulo 4

Evolución Diferencial Multiobjetivo

Existen diversos algoritmos utilizados para resolver problemas de optimización multiobjetivo, uno de ellos es la evolución diferencial(ED), esto debido a su simplicidad y su buen desempeño. Originalmente la DE fue pensada para la optimización de un solo objetivo sin restricciones, sin embargo existen muchas extensiones para optimización multiobjetivo han sido propuestas. Las primeras propuestas convirtieron los problemas de optimización multiobjetivo en problemas mono-objetivo como se muestra en [2]. Las siguientes propuestas se basaron principalmente en la dominancia de Pareto, algunos de estos algoritmos son:

- Pareto-based DE approach.
- Pareto Differential Evolution.
- PDE algorithm.
- Generalized Differential Evolution (GDE).

El algoritmo GDE fue introducido por primera vez por S. Kukkonen y J. Lampinen en [24] como parte de su tesis doctoral en el año 2004.

4.1. Generalized Differential Evolution (GDE)

GDE es una extensión de la Evolución diferencial para optimización multiobjetivo. La diferencia que presenta con otros enfoques es que no requiere parámetros de control adicionales, esto con el objetivo de mantener la simplicidad del algoritmo de Evolución Diferencial y evitar una complejidad innecesaria. Por lo tanto, GDE es un optimizador mono-objetivo y multiobjetivo y es más simple en comparación con otros enfoques.

GDE utiliza la estrategia DE/rand/1/bin. Esta estrategia fue seleccionada debido a su simplicidad y buen desempeño, también debido a que es la estrategia comúnmente más usada en la literatura especializada.

Existen varias versiones del algoritmo GDE, las cuales difieren en la forma en que se realiza la optimización multiobjetivo con mayor precisión y en como se mantiene la diversidad de las soluciones durante la búsqueda.

Primera Versión, GDE1

La primera versión de la evolución diferencial generalizada fue obtenida modificando el operador de selección con el objetivo de crear un algoritmo para optimización multiobjetivo con restricciones. Esta modificación consistió en utilizar el criterio de no dominancia entre el vector hijo con el vector padre, y de esta forma reemplazarlo en la siguiente generación. Esta versión proporcionaba soluciones muy buenas pero el algoritmo era muy sensible a sus parámetros de control.

Segunda Versión, GDE2

En esta versión se agregó el uso de la distancia de crowding cuando el vector hijo y el vector padre son no-dominados entre sí en el espacio de las funciones objetivo. Esta modificación mejoró la extensión y distribuciones de las soluciones en el frente de Pareto. Sin embargo, hizo más lenta la convergencia del algoritmo debido a que esta nueva modificación favorecía a los vectores más aislados del Frente de Pareto. Aunque esta versión tuvo mejores resultados que la primera versión, todavía era demasiado sensible a los valores de sus parámetros de control.

Tercera Versión, GDE3

La tercera versión extiende la estrategia DE/rand/1/bin a problemas multiobjetivo con restricciones. Este algoritmo maneja cualquier número de objetivos y cualquier número de restricciones.

Uno de los cambios en esta versión es la etapa de selección la cual se basa en las siguientes tres reglas:

- En el caso de los vectores no factibles, si el vector hijo domina débilmente al vector padre en el espacio de las restricciones, este es seleccionado para la nueva generación, en caso contrario el vector padre es seleccionado.
- En el caso de que uno vector es factible y el otro no, el vector factible es seleccionado.
- Si ambos vectores son factibles, el vector hijo es seleccionado si domina débilmente al vector padre en el espacio de los objetivos, en caso contrario el vector padre es seleccionado. Existe la posibilidad de que ninguno de los dos vectores domine al otro, para tal caso ambos vectores son seleccionados para la siguiente generación.

Debido a la tercera regla de selección, existe la posibilidad de que el tamaño de la población incremente. Para este caso, la población se vuelve a reducir al tamaño original clasificando a los vectores usando el criterio de no-dominancia y la distancia de Crowding.

La distancia de Crowding se calcula clasificando primero el conjunto de soluciones en función del valor de su función objetivo de manera ascendente. La distancia de Crowding de una solución particular es la distancia promedio de sus soluciones vecinas. Las soluciones en los límites, es decir, la mejor solución y la peor solución de acuerdo a los valores de las funciones objetivo reciben valores de distancia de Crowding infinitos, esto con la finalidad de que siempre sean seleccionados. El valor final de la distancia de Crowding se calcula

sumando los valores de distancia de Crowding individuales en cada función objetivo. Los miembros de la población con los peores valores obtenidos basados en estas medidas, son eliminados para reducir la población a su tamaño original.

Para el caso de un solo objetivo sin restricciones, GDE 3 es idéntico al algoritmo de Evolución Diferencial original. Los detalles del algoritmo GDE 3 se pueden ver en el Algoritmo 5.

Algoritmo 5: Evolución Diferencial Generalizada 3

Entrada: Porcentaje de cruce (CR), factor de escalamiento (F), número máximo de generaciones (G_{max}), tamaño de población ($NP \geq 4$)

mientras $G < G_{max}$ **hacer**

para $i \in \{1, \dots, NP\}$ **hacer**

 Mutación

$r_1, r_2, r_3 \in NP$ /* Seleccionados de manera aleatoria y diferentes del vector i */

$V_{mutante} \leftarrow F(r_1 - r_2) + r_3$ /* Creación del vector mutante */

 Recombinación

para V_{target_j} **hacer**

si $rand_j(0, 1) < CR$ **entonces**

$V_{trial_j} = V_{mutante_j}$

en otro caso

$V_{trial_j} = V_{target_j}$

 Selección

si $V_{trial} \preceq V_{target}$ **entonces**

$V_{target} \leftarrow V_{trial}$

en otro caso

si $V_{target} \preceq V_{trial}$ **entonces**

$V_{target} \leftarrow V_{target}$

en otro caso

$V_{target} \neq V_{trial}$

 /* Ambos vectores son agregados a la siguiente generación */

$m = m + 1$

mientras $m > 0$ **hacer**

si $\vec{x} \not\prec_c (\vec{x} + 1)$ **entonces**

 Ordenamiento por No Dominancia

 Truncar tamaño de población

en otro caso

para $i \in \{1, \dots, NP\}$ **hacer**

 Distancia de Crowding

 Remover Solución

$m = m - 1$

$G = G - 1$

4.2. Feature Selection Generalized Differential Evolution 3 (FSGDE 3)

FSGDE 3, la propuesta principal de este trabajo de tesis, es un algoritmo multiobjetivo basado en GDE 3 diseñado para el problema de selección de características utilizando el método envoltorio (Fig. 4.1), es decir, se selecciona un subconjunto de características y se utiliza un clasificador para evaluar el subconjunto seleccionado y entonces generar un modelo de clasificación. De esta manera teniendo el modelo de clasificación, se obtienen medidas de calidad, las cuales para el problema de selección de características y en particular para este trabajo de tesis son el porcentaje de error y el número de características seleccionadas.

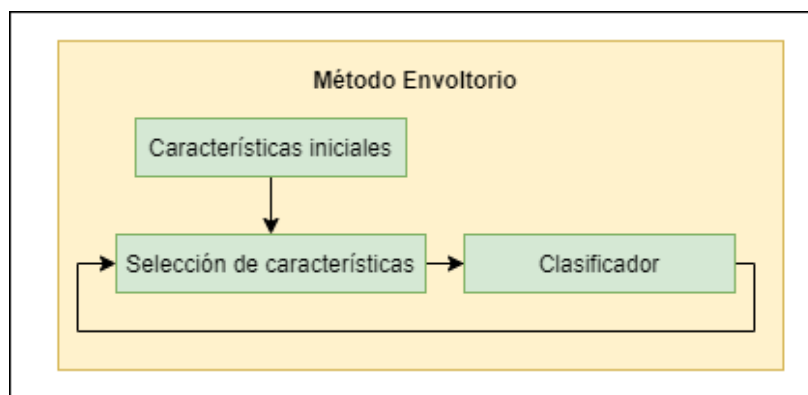


Figura 4.1: Método Envoltorio.

FSGDE 3 fue diseñado siguiendo los seis pasos para resolver un problema dado usando algoritmo evolutivo.

4.2.1. Representación

La representación utilizada para el desarrollo de este algoritmo, se hace mediante el uso de permutaciones tomando en cuenta cuatro aspectos:

1. Se codifican todas las características existentes dentro del conjunto de datos utilizando su valor de índice.
2. El tamaño del subconjunto de características se determina dentro del proceso evolutivo utilizando un elemento adicional dentro del vector codificado.
3. La relevancia de cada característica utilizada para la construcción de un mejor clasificador se representa a través de su posición en el vector y se puede utilizar un operador de mutación basado en una matriz de permutación para generar nuevas soluciones.
4. Se implementa un operador de recombinación basado en reparaciones para asegurar la construcción de nuevas soluciones factibles.

Esquema de codificación

El vector utilizado para codificar las características del conjunto de datos es creado de la siguiente manera:

- Para un conjunto de datos con m características, El tamaño del vector codificado sera $n \leftarrow m + 1$.
- Los valores en el vector codifican la posición de las características dentro del conjunto de datos, siendo el número 1 el valor que representa a la primera característica.
- El elemento adicional utilizado, se representa con el número 0 el cual se utiliza como un indicador para seleccionar las características a utilizar en el proceso de construcción del clasificador. Tomando los índices ubicados a la izquierda del valor 0 como parte del subconjunto candidato de datos y los índices ubicados a la derecha como descartados.

Este esquema puede observarse en la Fig. 4.2.

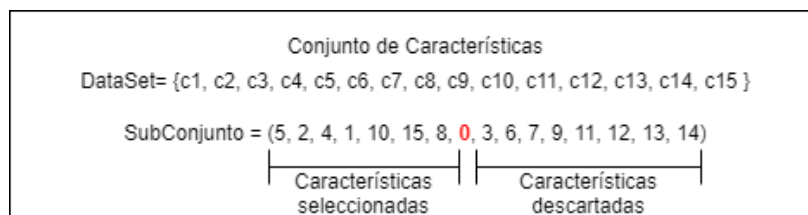


Figura 4.2: Esquema de codificación.

4.2.2. Funciones Objetivo

El algoritmo GDE 3 es un algoritmo que resuelve problemas de optimización multiobjetivo y para efectos de esta tesis se busca un buen compromiso entre dos funciones objetivo que se encuentran en conflicto.

- Minimizar el numero de características seleccionadas (favorecer modelos más compactos y que sean más fácilmente explicables).
- Minimizar el error de clasificación, es decir, aumentar la precisión del clasificador.

4.2.3. Población de Soluciones

Se generan NP soluciones de manera aleatoria. Cada solución candidata (permutación) representa un subconjunto de características, como se muestra en la Fig. 4.3.

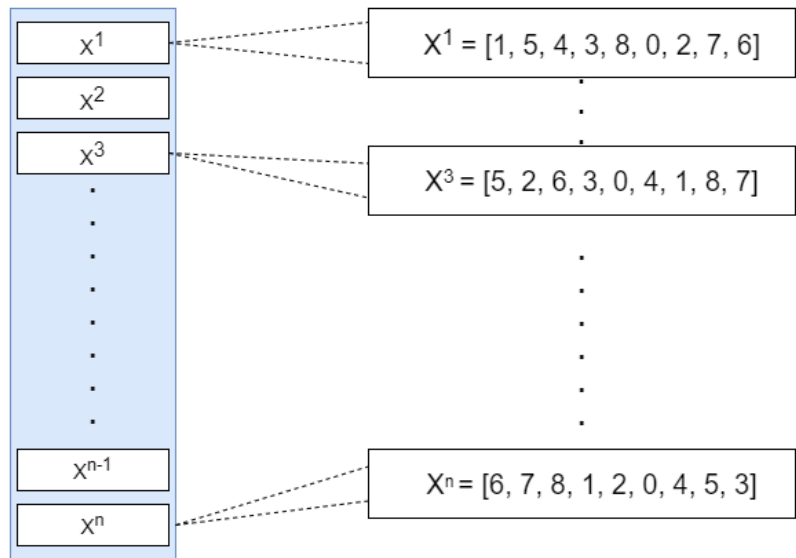


Figura 4.3: Población de Soluciones.

4.2.4. Mecanismo de Selección Padres

Evolución diferencial no tiene un proceso de selección de padres como tal, pues cada vector de la población genera un descendiente $vector_{hijo}$. Sin embargo, para poder generar el vector mutante que se cruza con el vector padre $vector_i$, se seleccionan aleatoriamente tres vectores distintos entre sí y distintos al $vector_i$. Este proceso se puede observar en la Fig. 4.4

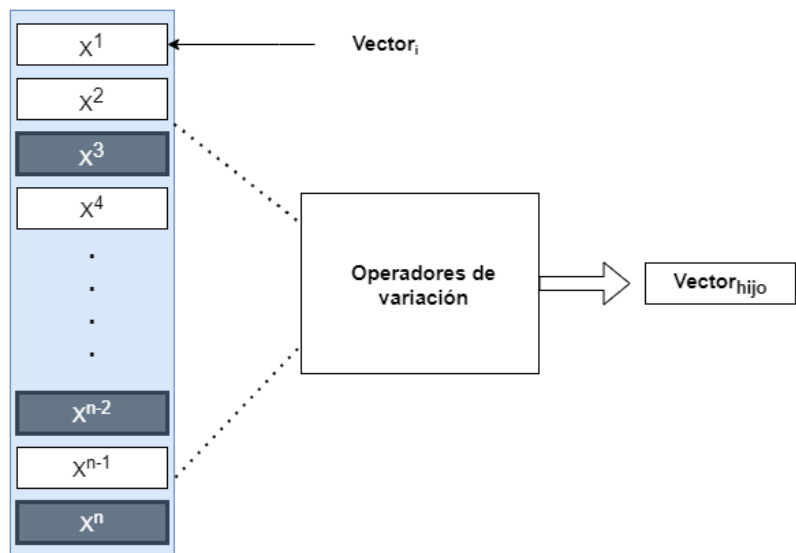


Figura 4.4: Selección de Vectores.

4.2.5. Operadores de variación

Los operadores de variación utilizados por el algoritmo FSGDE 3, son los mismos de la Evolución Diferencial Simple, Mutación y Recombinación, los cuales pueden observarse en la Fig. 4.5. Sin embargo al utilizar una representación basada en permutaciones, estos operadores son adaptados para el manejo de ellas.

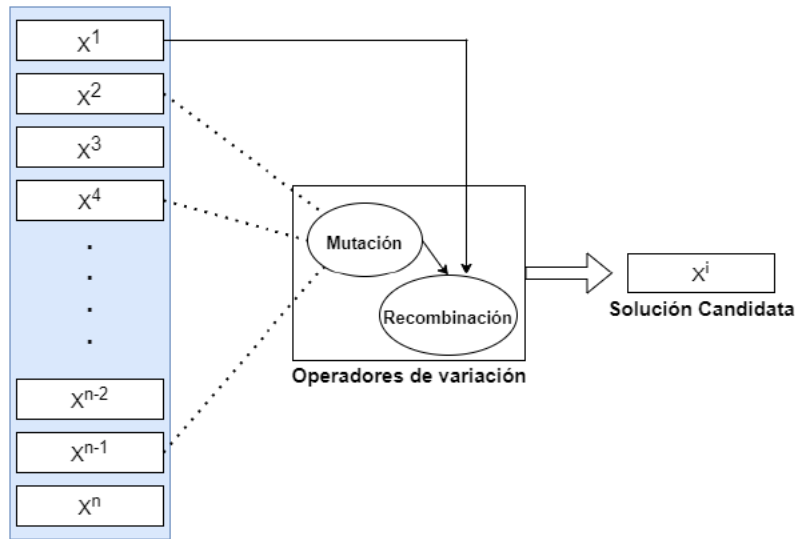


Figura 4.5: Operadores de Variación del algoritmo FSGDE 3.

Mutación

Este operador implementa el enfoque de la matriz de permutaciones [35], el cual representa mediante una matriz P la diferencia entre dos vectores representados mediante permutaciones.

Teniendo los vectores $X_{r_1} = [1, 5, 4, 0, 3, 2]$, $X_{r_2} = [3, 1, 0, 5, 2, 4]$ y $X_{r_3} = [4, 5, 3, 1, 0, 2]$ se genera la matriz de permutación P de la forma $X_{r_2} \leftarrow PX_{r_3}$ como se muestra a continuación:

$$\begin{bmatrix} 3 \\ 1 \\ 0 \\ 5 \\ 2 \\ 4 \end{bmatrix} \leftarrow \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 4 \\ 5 \\ 3 \\ 1 \\ 0 \\ 2 \end{bmatrix}$$

La matriz de permutaciones se genera colocando un 1 en la posición donde se encuentra el elemento del vector X_{r_2} en el vector X_{r_3} , es decir, se genera la primera fila de la siguiente forma, $[0, 0, 1, 0, 0, 0]$ debido a que el primer elemento del vector X_{r_2} se encuentra en la tercera posición del vector X_{r_3} . Siguiendo este procedimiento se genera la matriz de permutaciones.

Teniendo completa la matriz de permutaciones y un factor de escalamiento F , el cual se usa como un valor de umbral, se genera el vector mutante usando el vector X_{r_1} de la

siguiente manera: $V^i \leftarrow P_F X^{r^1}$.

Para cada valor i , se genera de manera aleatoria un valor r_i , si r_i es mayor que F entonces se busca la j -ésima fila donde el valor sea distinto de cero y se intercambian las filas. Por ejemplo, considerando $P_F = (0.3, 0.7, 0.3, 0.6, 0.3, 0.8)$ el procedimiento que genera el vector mutante se muestra en la Fig. 4.6:

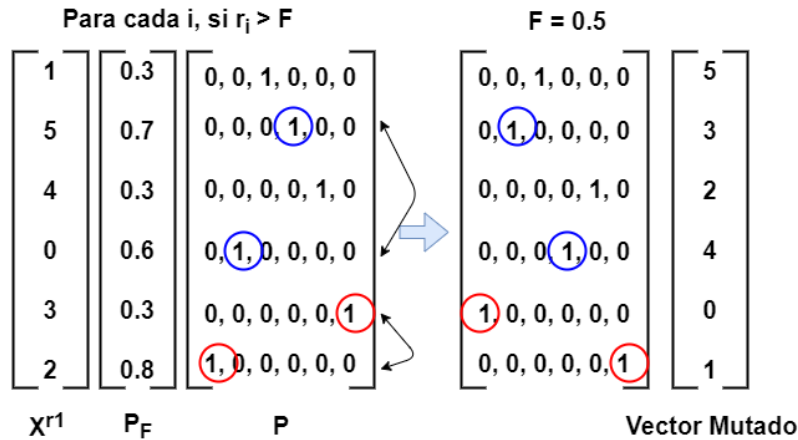


Figura 4.6: Procedimiento del operador de Mutación.

El vector mutante resultante se toma a partir del vector X_{r^1} , tomando la posición del valor 1 en cada una de las columnas de la matriz de permutaciones. De esta forma el primer elemento del vector mutante es el segundo elemento del vector X_{r^1} , el segundo elemento es el penúltimo elemento del vector X_{r^1} y así sucesivamente hasta obtener el vector mutado.

Recombinación

Este operador de variación combina los valores del vector mutado con el vector X_i para generar el vector hijo (trial). Sin embargo, el vector trial generado puede contener valores duplicados, los cuales no son permitidos debido a que se está trabajando con permutaciones. Para solucionar este problema se realiza un procedimiento de reparación, el cual consiste en recorrer el vector obtenido e ir eliminando los elementos duplicados; cada vez que un elemento es eliminado, el siguiente elemento en el vector se recorre a la izquierda. De este modo se obtiene un vector con un cierto número de posiciones vacías. Para completar el vector resultante, se toma el vector X_i y se hace una comparación con los elementos contenidos en el vector trial. Cuando un elemento del vector X_i no se encuentra dentro del vector trial, este elemento se agrega al final del vector para completar los elementos faltantes. En la Fig. 4.7 se muestra el procedimiento de reparación.

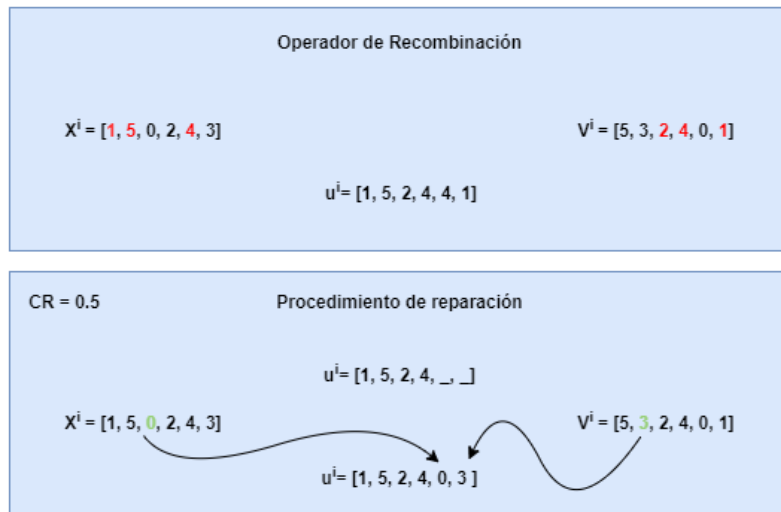


Figura 4.7: Procedimiento de reparación.

En el ejemplo de la Fig. 4.7 se tiene un valor de $Cr = 0.5$. De manera aleatoria se genera un valor entre 0 y 1, si el valor generado es mayor al valor de CR se toma el elemento del Vector X^i , en caso contrario se toma el valor del vector V^i , de esta manera se genera el vector u^i . Sin embargo, como se muestra en la Fig. 4.7, los últimos dos elementos del vector contienen valores duplicados, teniendo esto en cuenta se realiza un recorrido del vector u^i y se eliminan los elementos duplicados, los espacios faltantes se completan utilizando los valores de los vectores X^i y V^i que no se encuentran dentro del vector u^i .

4.2.6. Mecanismo de Reemplazo

El mecanismo de reemplazo utilizado en este algoritmo al tratarse de un algoritmo multi-objetivo, es mediante la no dominancia entre soluciones, este proceso puede ser implementado siguiendo las 3 reglas de selección, las cuales se ven a continuación:

1. Si el vector X^i domina débilmente al vector u^i , el vector X^i es seleccionado como parte de la siguiente generación.
2. Si el vector u^i domina débilmente al vector X^i , el vector u^i es seleccionado como parte de la siguiente generación.
3. Si ninguna de las dos soluciones se dominan entre si, ambas soluciones son agregadas a la siguiente generación.

En el algoritmo 6 puede observarse el funcionamiento del FSGDE 3. Como puede observarse en el algoritmo, el FSGDE 3 es una extensión de GDE 3, donde los cambios realizados fueron los siguientes:

1. Representación basada en Permutaciones.
2. Cambio de los operadores de variación.

Algoritmo 6: Evolución Diferencial Generalizada 3 para selección de características

Entrada: Porcentaje de cruce (CR), factor de escalamiento (F), número máximo de generaciones (G_{max}), tamaño de población ($NP \geq 4$)

mientras $G < G_{max}$ **hacer**

- para** $i \in \{1, \dots, NP\}$ **hacer**
 - Mutación
 - $r_1, r_2, r_3 \in NP$ /* Seleccionados de manera aleatoria y diferentes del vector i */
 - $V_{mutante} \leftarrow P_F X_{r_1}$ /* Creación del vector mutante */
 - Recombinación
 - para** V_{target_j} **hacer**
 - si** $rand_j(0, 1) < CR$ **entonces**
 - $V_{trial_j} = V_{mutante_j}$
 - en otro caso**
 - $V_{trial_j} = V_{target_j}$
 - Reparación
 - Selección
 - si** $V_{trial} \preceq V_{target}$ **entonces**
 - $V_{target} \leftarrow V_{trial}$
 - en otro caso**
 - si** $V_{target} \preceq V_{trial}$ **entonces**
 - $V_{target} \leftarrow V_{target}$
 - en otro caso**
 - $V_{target} \neq V_{trial}$
 - /* Ambos vectores son agregados a la siguiente generación */
 - $m = m + 1$

- mientras** $m > 0$ **hacer**
- si** $\vec{x} \not\prec_c (\vec{x} + 1)$ **entonces**
 - Ordenamiento por No Dominancia
 - Truncar tamaño de población
- en otro caso**
 - para** $i \in \{1, \dots, NP\}$ **hacer**
 - Distancia de Crowding
 - Remove Solución
- $m = m - 1$
- $G = G - 1$

Capítulo 5

Experimentación y Resultados

En este capítulo se describen los estudios experimentales realizados para analizar el desempeño del algoritmo FSGDE 3, el cual se desarrolló en este trabajo de tesis. Se definirán los conjuntos de datos usados para la experimentación y se definirán los parámetros utilizados por el algoritmo FSGDE 3. También se realizará una descripción del algoritmo NSGA II, el cual será utilizado para su comparación, se definirán las medidas de desempeño utilizadas y las pruebas estadísticas para validar los resultados obtenidos. Así se realizará una comparación con los resultados obtenidos por algunos algoritmos de la literatura especializada, los cuales se describen a continuación:

- **SFS (Sequential Forward Selection).** Esta estrategia de selección agrega secuencialmente una característica la cual se combina con las seleccionadas previamente. Esta estrategia comienza con un conjunto de características vacío.
- **SBE Sequential Backward Elimination).** Esta estrategia elimina secuencialmente la característica que contribuye menos a la mejora del valor de la medida de calidad. Esta estrategia comienza con el conjunto completo de características.
- **DE-FS (Differential Evolution Feature Selection).** Este algoritmo está basado en permutaciones y utiliza un enfoque de envoltorio para su desarrollo.

5.1. Marco Experimental

FSGDE 3 se encuentra implementado en lenguaje Java utilizando la librería JMetal [11]. Para validarlo se compara con FSNSGA II, cuya implementación también se realizó en lenguaje Java utilizando la librería JMetal y los resultados obtenidos por los algoritmos de la literatura descritos con anterioridad.

En la tabla 5.1 se listan los parámetros usados en los experimentos, el número máximo de evaluaciones, el cual se definió en 50000 para ambos algoritmos evitando descartar el buen desempeño del algoritmo con un número mayor de evaluaciones.

Las funciones de aptitud que se utilizaron en ambos algoritmos calculan el error de clasificación y el tamaño del subconjunto de características seleccionadas. Ambos algoritmos

se implementaron utilizando el enfoque de envoltorio, es decir, realizando la búsqueda y prueba de los subconjuntos de datos dentro del algoritmo mediante un clasificador, el cual para este problema, fue iBk [9].

iBk es un método de clasificación no paramétrico supervisado, el cual permite clasificar un elemento dentro de la clase con mayor frecuencia a la cual pertenecen sus K vecinos más cercanos, el valor de K utilizado usualmente comprende los valores entre 3 y 7. Por lo que para este trabajo se definió como $k = 5$ debido a que el algoritmo mono objetivo utilizó el mismo valor de $k = 5$ para realizar la clasificación.

Tanto FSGDE 3 y FSNSGA II utilizan la técnica de validación cruzada con 10 particiones para estimar el error y probar la precisión dentro del algoritmo evolutivo.

Tabla 5.1: Parámetros utilizados en los experimentos. Para ambos algoritmos, los parámetros fueron calibrados mediante IRACE

Parámetro	FSGDE 3	FSNSGA II
Tamaño de población	250	100
Valor de aptitud	Porcentaje de error, tamaño	Porcentaje de error, tamaño
Porcentaje de cruce (GDE 3)	0.7049	————
Factor de escalamiento (GDE 3)	0.8305	————
Máximo número de evaluaciones	50000	50000
Probabilidad de mutación	————	0.9684
Probabilidad de cruce	————	0.0914

5.2. Selección de datos de prueba

Para realizar el estudio experimental se utilizó un grupo de conjuntos de datos (datasets) los cuales se muestran en la Tabla 5.2. Los primeros 20 conjunto de datos han sido seleccionados del repositorio de aprendizaje automático de la UCI [10] y los 5 restantes, son microarreglos seleccionados del repositorio de conjunto de datos biomédicos de Kent Ridge [57]. Estos conjuntos de datos han sido seleccionados debido a que poseen características numéricas y categóricas, también debido a que el número de características es variable. En la tabla 5.3 se muestran los conjuntos de datos divididos en 3 criterios:

- Conjunto de datos pequeños. Entre 1 y 100 características.
- Conjunto de datos medianos. Entre 101 y 500 características.
- Conjunto de datos grandes. Mas de 500 características.

5.3. Sintonización de parámetros

La configuración de parámetros en un algoritmo evolutivo es muy importante para su diseño, ya que de esto dependerá su buen desempeño. Debido a que la tarea de configurar dichos parámetros no es trivial, existen diversas herramientas para automatizar este proceso.

Tabla 5.2: Descripción de los datasets utilizados en los experimentos

Dataset	Instancias	Características	Clases	Dataset	Instancias	Características	Clases
arrhythmia	452	279	16	audiology	226	69	24
australian	690	14	2	crx	690	15	2
cylinder-b	540	39	2	german-c	1000	20	2
dermatology	366	34	6	ionosphere	351	34	2
h-valley	1212	100	2	m-libras	360	90	15
musk-1	476	168	2	parkinsons	195	22	2
someion	1593	256	10	sonar	208	60	2
soybean	683	35	19	spectf	267	44	2
splice	3190	60	3	vehicle	846	18	4
vote	435	16	2	wdbc	569	30	2
microarreglos:							
all-aml	72	7129	3	colon	62	2000	2
leukemia	72	7129	2	lymphoma	62	4026	3
srbct	83	2308	4				

En [12] se divide la configuración de parámetros en dos grupos: **Control de Parámetros** y **Calibración de parámetros**.

El control de parámetros consiste en variar los valores durante la ejecución del algoritmo, por lo que se definen valores iniciales y un mecanismo de control, el cual permite al algoritmo adaptarse automáticamente.

La calibración de parámetros es tratada como un problema de optimización y existen una gran variedad de métodos. Para este trabajo de tesis el método utilizado para la calibración de parámetros fue la herramienta llamada IRACE. El objetivo de la configuración de parámetros es encontrar la mejor configuración de ellos.

IRACE es una implementación del método racing iterado [3]. El racing iterado es un método de configuración automática de tres pasos:

1. Muestrear nuevas configuraciones.
2. Seleccionar las mejores configuraciones.
3. Actualizar la distribución de las configuraciones en función de las mejores configuraciones.

Estos pasos se realizan de manera iterativa hasta alcanzar un criterio de minimización.

El racing iterado es implementado en IRACE. Para cada parámetro se asocia una distribución de muestreo independiente de los otros parámetros.

El proceso de racing se describe como una competencia de carreras, la cual comienza con un conjunto de configuraciones candidatas. En cada paso de la carrera, cada configuración es evaluada, después de un número determinado de pasos, las soluciones con la peor calidad son eliminadas y la carrera continua con las configuraciones sobrevivientes.

5.3.1. Resultados de IRACE

Para calibrar los parámetros de manera automática con IRACE, se tomaron 5 conjuntos de datos con diferentes características, los cuales pueden observarse en la tabla 5.4, el ob-

Tabla 5.3: Clasificación de los conjuntos de datos basado en su número de características

Conjunto de datos	Características	Criterio	Conjunto de datos	Características	Criterio
australian	14	pequeño	arrhythmia	279	mediano
cylinder-b	39	pequeño	musk-1	168	mediano
dermatology	34	pequeño	semeion	256	mediano
h-valley	100	pequeño	leukemia	7129	grande
audiology	69	pequeño	colon	2000	grande
crx	15	pequeño	all-aml	7129	grande
german-c	20	pequeño	lymphoma	4026	grande
ionosphere	34	pequeño	srbc	2308	grande
m-libras	90	pequeño			
soybean	35	pequeño			
splice	60	pequeño			
vote	16	pequeño			
parkinsons	22	pequeño			
sonar	60	pequeño			
spectf	44	pequeño			
vehicle	18	pequeño			
wdbc	30	pequeño			

jetivo utilizado para la minimización fue con base a la medida del indicador de desempeño Spacing.

Tabla 5.4: Conjunto de datos Utilizados para la calibración de parámetros utilizando IRACE.

Dataset	Instancias	Características	Clases
australian	690	14	2
dermatology	366	34	6
glass	214	9	7
ionosphere	351	34	2
soybean	683	35	19

Para la calibración de parámetros se utilizó un equipo de cómputo con las siguientes especificaciones: AMD Ryzen 5 2500U with Radeon Vega Mobile Gfx 2.0 GHz, Memoria RAM 8 GB y sistema operativo Windows 10 Home Single.

Se realizaron cinco iteraciones de IRACE con los rangos de valores para cada parámetro. Como se puede observar en la tabla 5.5 se realizó una calibración de parámetros para FSGDE 3 y para FSNSGA II.

Después de realizar las 5 iteraciones de IRACE para cada uno de los algoritmos, se obtuvieron las siguientes configuraciones de parámetros:

FSGDE 3

- Configuración 1 : $CR = 0.6440$, $F = 0.8636$, $NP = 250$
- Configuración 2 : $CR = 0.7554$, $F = 0.8425$, $NP = 250$

Tabla 5.5: Parámetros y rangos de valores para calibración con IRACE.

FSGDE 3	
Parámetro	Rango de valores
CR	(0.0, 1.0)
F	(0.0, 1.0)
NP	(100, 150, 200, 250, 300)
FSNSGA II	
CP	(0.0, 1.0)
MP	(0.0, 1.0)
NP	(100, 150, 200, 250, 300)

- Configuración 3 : $CR = 0.7174$, $F = 0.7855$, $NP = 250$

FSNSGA II

- Configuración 1 : $CP = 0.0914$, $MP = 0.9684$, $NP = 100$
- Configuración 2 : $CP = 0.1178$, $MP = 0.9831$, $NP = 200$
- Configuración 3 : $CP = 0.1581$, $MP = 0.9985$, $NP = 250$

Para ambos algoritmos la configuración de parámetros utilizada fue la primera, esto debido a que fue la configuración que obtuvo mayor peso, basado en los cálculos estadísticos que realizó IRACE, los cuales pueden observarse en la tabla 5.1.

5.4. Diseño Experimental

En este estudio experimental se busca determinar que la selección de características se puede realizar utilizando un algoritmo con enfoque multiobjetivo que presente un compromiso entre el porcentaje de clasificación y el número de características seleccionadas.

5.4.1. Experimento 1

El primer experimento realiza una comparación entre los resultados de FSGDE 3 y FSNSGA II con el objetivo de medir el rendimiento del algoritmo. Para este experimento se utilizan los 25 conjuntos de datos descritos anteriormente. Se realizaron 30 ejecuciones de cada algoritmo con cada uno de los conjuntos de datos utilizados, los cuales generaron valores de Spacing e Hipervolumen respectivamente, tomando estos valores se realizó un estudio estadístico para analizar el desempeño de FSGDE 3 en relación con FSNSGA II.

5.4.2. Experimento 2

El segundo experimento realiza una comparación de los resultados de FSGDE 3 con DE-FS[36], el cual tiene un enfoque mono objetivo, tomando como función de calidad solo el porcentaje de error, también se comparan los resultados con los algoritmos tradicionales SFS y SBE así como con el porcentaje de error obtenido al utilizar el conjunto de datos completo.

Para su análisis se toma el mejor, el peor y un valor intermedio denominado punto rodilla del frente acumulado generado por FSGDE 3, el cual comparará la precisión en cada una de las soluciones seleccionadas.

Para este experimento se tomaron los resultados de la tabla 2 del artículo publicado en [36] así como los conjuntos de datos utilizados, para su comparación de los resultados obtenidos con FSGDE 3.

5.5. Resultados

En esta sección se presentan los resultados obtenidos de los experimentos descritos en el apartado anterior, los cuales se describen a continuación:

- El primer experimento consiste en una comparación entre (FSGDE 3) y (NSGA II) adaptado para el problema de selección de características.
- El segundo experimento consiste en una comparación entre FSGDE 3 y los resultados obtenidos del algoritmo DE-FS así como con otros métodos tradicionales de selección de características.

A continuación se muestra de manera detallada los resultados obtenidos en cada uno de los experimentos realizados.

5.5.1. Experimento 1

Para el primer experimento se realizaron 30 ejecuciones de cada uno de los 25 conjuntos de datos mostrados en la tabla 5.2, buscando analizar un compromiso entre sus dos funciones objetivo (obtener el menor porcentaje de error utilizando un menor número de características posible). Estas funciones objetivo generan un conjunto de soluciones, las cuales forman un frente acumulado de Pareto: de estas soluciones se tomó la que obtuvo un mayor porcentaje de clasificación para realizar la comparación de los resultados de ambos algoritmos.

En la tabla 5.6 puede observarse el porcentaje de clasificación de la mejor solución del frente acumulado de los 25 conjuntos de datos utilizados, este porcentaje se obtuvo utilizando el clasificador iBk con un valor $k = 5$ y un esquema de validación mediante Hold-out utilizando un 70 % para entrenamiento y un 30 % para prueba.

Como puede observarse, los porcentajes de clasificación de ambos algoritmos son similares, sin embargo, FSNSGA II obtiene un mejor porcentaje de clasificación en la mayoría de los casos, lo cual se debe al tamaño de la solución seleccionada para su comparación.

Tabla 5.6: Tabla de resultados considerando la precisión de la mejor solución del frente acumulado.

Conjunto de Datos	FSNSGA II		FSGDE 3	
	Precisión	Rango	Precisión	Rango
arrhythmia	76.42 %	1	71.38 %	2
audiology	74.52 %	1	72.61 %	2
australian	90.89 %	1.5	90.89 %	1.5
crx	90.68 %	1	90.27 %	2
cylinder-b	82.01 %	1	78.04 %	2
dermatology	100.00 %	1	99.61 %	2
german-c	78.71 %	1	77.43 %	2
h-valley	64.74 %	1	62.97 %	2
ionosphere	95.93 %	1	94.72 %	2
m-libras	82.35 %	1	80.00 %	2
musk-1	99.10 %	1	97.30 %	2
parkinsons	99.27 %	1	98.54 %	2
semeion	92.38 %	1	92.29 %	2
sonar	97.26 %	1	91.78 %	2
soybean	94.17 %	1	92.29 %	2
spectf	92.51 %	1	88.77 %	2
splice	89.03 %	1	87.78 %	2
vehicle	78.04 %	1	77.03 %	2
vote	98.36 %	1	98.03 %	2
wdbc	99.50 %	1	98.99 %	2
Micro arreglos:				
all-aml	96.08 %	2	98.04 %	1
leukemia	100.00 %	1.5	100.00 %	1.5
colon	97.67 %	1.5	97.67 %	1.5
lymphoma	100.00 %	1.5	100.00 %	1.5
srbc	100.00 %	1.5	100.00 %	1.5
Promedio de rangos		1.14		1.86

En la tabla 5.7 puede observarse que el tamaño del subconjunto de datos de la solución obtenida con FSGDE 3 en la mayoría de los casos es menor que el número de características seleccionadas por FSNSGA II, por lo tanto, debido a que el algoritmo propuesto reduce considerablemente el subconjunto de datos, su porcentaje de error aumenta en comparación con FSNSGA II.

Tabla 5.7: Tabla de resultados considerando el tamaño del subconjunto de la mejor solución del conjunto de soluciones no dominadas.

Conjunto de Datos	Características	FSNSGA II		FSGDE 3	
		Tamaño	Reducción	Tamaño	Reducción
arrhythmia	452	19	95.8 %	8	98.2 %
audiology	69	10	85.5 %	8	88.4 %
australian	14	6	57.1 %	4	71.4 %
crx	15	14	6.7 %	4	73.3 %
cylinder-b	39	11	71.8 %	12	69.2 %
dermatology	34	30	11.8 %	26	23.5 %
german-c	20	13	35.0 %	4	80.0 %
h-valley	100	10	90.0 %	5	95.0 %
ionosphere	34	3	91.2 %	4	88.2 %
m-libras	90	22	75.6 %	17	81.1 %
musk-1	168	12	92.9 %	4	97.6 %
parkinsons	22	8	63.6 %	6	72.7 %
semeion	256	202	21.1 %	172	32.8 %
sonar	60	19	68.3 %	21	65.0 %
soybean	35	15	57.1 %	26	25.7 %
spectf	44	12	72.7 %	2	95.5 %
splice	60	5	91.7 %	4	93.3 %
vehicle	18	9	50.0 %	8	55.6 %
vote	16	9	43.8 %	8	50.0 %
wdbc	30	8	73.3 %	7	76.7 %
Micro arreglos:					
all-aml	7129	12	99.8 %	2	100.0 %
leukemia	7129	17	99.8 %	2	100.0 %
colon	2000	6	99.7 %	2	99.9 %
lymphoma	4026	5	99.9 %	1	100.0 %
srbc	2308	48	97.9 %	37	98.4 %

Pruebas estadísticas

Para determinar si el algoritmo propuesto es mejor que FSNSGA II se obtuvieron los valores de los indicadores de desempeño Spacing e Hipervolumen de las 30 ejecuciones realizadas con cada uno de los conjuntos de datos, con estos resultados se realizó la prueba estadística no paramétrica de Wilcoxon [52], la cual consiste en la comparación del rango medio de dos muestras relacionadas y así determinar si existen diferencias entre ellas. Para nuestro caso se hizo uso de esta prueba con un 95 % de confiabilidad para determinar si ambos algoritmos se comportan de manera similar o no, es decir, cuando el resultado de la prueba estadística es menor a 0.5, se dice que existen diferencias significativas, lo cual puede interpretarse como que alguno de los dos algoritmos es mejor, no obstante, si el resultado es mayor a 0.5, se dice que ambos algoritmos se comportan de manera similar, por lo cual, cualquiera de los dos ofrecerá un conjunto de soluciones adecuadas.

Tabla 5.8: Prueba de Wilcoxon utilizando el Indicador de Desempeño Spacing.

Prueba de Wilcoxon – Spacing				
Conjunto de Datos	FSGDE 3	FSNSGA II	p-value	Resultado
arrhythmia	0.0135	0.0237	0.0385	+
audiology	0.0169	0.0064	0.4761	=
australian	0.0135	0.0237	0.6048	=
crx	0.0318	0.0199	0.6573	=
cylinder-b	0.0387	0.0191	0.1091	=
dermatology	0.0570	0.0611	0.2132	=
german-c	0.0212	0.0262	0.8087	=
h-valley	0.0034	0.0159	0.1188	=
ionosphere	0.0122	0.0258	0.0176	+
m-libras	0.0598	0.0595	0.6865	=
musk-1	0.0850	0.0763	0.0722	=
parkinsons	0.0198	0.0238	0.8130	=
semeion	0.0367	0.0435	0.0033	+
sonar	0.0315	0.0356	0.8430	=
soybean	0.0579	0.0628	0.0003	+
spectf	0.0000	0.0054	0.0184	+
splICE	0.0345	0.0244	0.2359	=
vehicle	0.0475	0.0492	0.3208	=
vote	0.0018	0.0018	0.1047	=
wdbc	0.0695	0.0286	0.0371	-
Micro arreglos:				
all-aml	0.0000	0.0000	0.8011	=
leukemia	0.0000	0.0000	0.5440	=
colon	0.0000	0.0000	0.0069	=
lymphoma	0.0000	0.0000	0.3337	=
srbc	0.0562	0.0217	0.0180	-

En la tabla 5.8 y 5.9 se pueden observar los valores obtenidos después de la prueba de Wilcoxon. Como se puede notar, los resultados en su mayoría, ambos algoritmos son comparables, es decir, que ambos algoritmos genera un conjunto de soluciones factibles, teniendo un porcentaje de clasificación alto y manteniendo una reducción considerable del número de características totales.

Tabla 5.9: Prueba de Wilcoxon utilizando el Indicador de Desempeño Hipervolumen.

Prueba de Wilcoxon – Hipervolumen				
Conjunto de Datos	FSGDE 3	FSNSGA II	p-value	Resultado
arrhythmia	0.6868	0.7134	1.1×10^{-12}	-
audiology	0.6815	0.7015	6.2×10^{-07}	-
australian	0.8175	0.8243	1.2×10^{-02}	-
crx	0.8205	0.8236	3.2×10^{-03}	-
cylinder-b	0.7317	0.7482	7.6×10^{-07}	-
dermatology	0.9035	0.9196	1.2×10^{-10}	-
german-c	0.7190	0.7246	3.5×10^{-04}	-
h-valley	0.6101	0.6239	6.8×10^{-07}	-
ionosphere	0.9007	0.9074	3.0×10^{-03}	-
m-libras	0.7501	0.7748	2.3×10^{-08}	-
musk-1	0.9579	0.9709	1.6×10^{-09}	-
parkinsons	0.9129	0.9197	1.9×10^{-02}	-
semeion	0.8563	0.8612	4.4×10^{-05}	-
sonar	0.8648	0.9014	2.3×10^{-09}	-
soybean	0.8098	0.8362	7.4×10^{-11}	-
spectf	0.8354	0.8539	5.8×10^{-06}	-
splice	0.8315	0.8462	2.3×10^{-03}	-
vehicle	0.6883	0.6960	3.8×10^{-04}	-
vote	0.9052	0.9068	3.7×10^{-02}	-
wdbc	0.9465	0.9499	1.1×10^{-02}	-
Micro arreglos:				
all-aml	0.9606	0.9181	6.5×10^{-07}	+
leukemia	0.9998	0.9526	4.7×10^{-07}	+
colon	0.9412	0.9260	2.7×10^{-02}	+
lymphoma	0.9997	0.9954	2.0×10^{-11}	+
srbc	0.9623	0.9784	4.1×10^{-05}	+

Ejemplos

Se muestran 3 conjuntos de datos de la tabla 5.3, la cual describe cuantas características tiene cada uno de los conjuntos de datos utilizados para este experimento, en este apartado se seleccionó 1 conjunto de datos de cada tamaño para observar su comportamiento utilizando ambos algoritmos.

- **Conjunto de datos pequeño:** m-libras, contiene 360 instancias, 90 características y 15 clases.
- **Conjunto de datos mediano:** semeion, contiene 1593 instancias, 256 características y 10 clases.

- **Conjunto de datos grande:** srbct, contiene 83 instancias, 2308 características y 4 clases.

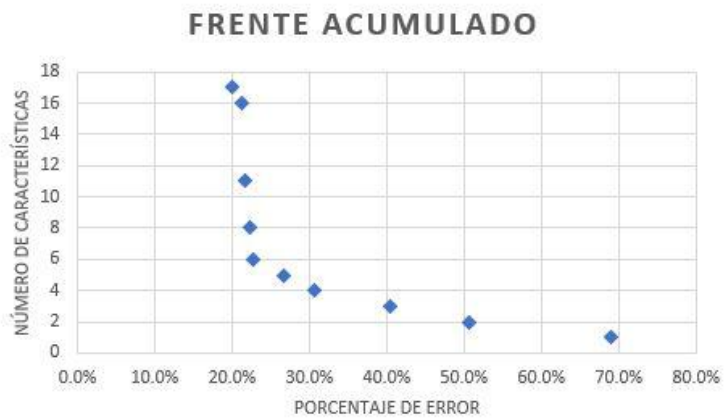
Conjunto de datos m-libras

Utilizando el conjunto de datos m-libras, el cual contiene 360 instancias, 90 características y 15 clases, se realizaron 30 ejecuciones de FSGDE 3 y FSNSGA II para selección de características.

FSGDE 3

FSGDE 3 generó un frente acumulado con un total de 10 soluciones, las cuales se muestran en la Fig. 5.1, así como también la representación gráfica del frente acumulado.

frente Acumulado	
% error	Tamaño
30.59%	4
22.35%	8
26.67%	5
21.57%	11
22.75%	6
50.59%	2
69.02%	1
21.18%	16
40.39%	3
20.00%	17



(a) Soluciones no Dominadas

(b) Frente Acumulado

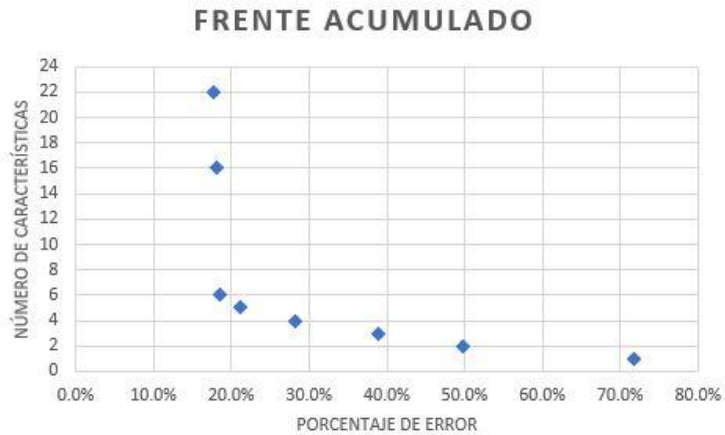
Figura 5.1: Resultados de FSGDE 3 utilizando el conjunto de datos m-libras.

En la tabla de la Fig. 5.1a puede observarse que la solución con un mayor porcentaje de clasificación fue la que obtuvo un 80 % de precisión utilizando solo 17 características, lo cual implica una reducción del conjunto de datos original del 81.1 %, mientras que la peor solución encontrada obtuvo aproximadamente un 31 % de porcentaje de clasificación, sin embargo, esta solución se obtuvo utilizando solo 1 característica, lo cual implica una reducción del conjunto de datos de aproximadamente el 99 %. Las demás soluciones del frente acumulado se ubican entre los dos extremos mencionados, las cuales pueden observarse gráficamente en la Fig. 5.1b.

FSNSGA II

FSNSGA II generó un frente acumulado con un total de 8 soluciones, las cuales se muestran en la Fig. 5.2, así como su representación gráfica.

Frente Acumulado	
% error	Tamaño
71.76%	1
49.80%	2
28.24%	4
18.04%	16
21.18%	5
38.82%	3
17.65%	22
18.43%	6



(a) Soluciones no Dominadas

(b) Frente Acumulado

Figura 5.2: Resultados de FSNSGA II utilizando el conjunto de datos m-libras.

En la tabla de la Fig. 5.2a puede observarse que la solución con un mayor porcentaje de clasificación fue la que obtuvo un 82.35 % de precisión utilizando solo 22 características, lo cual implica una reducción del conjunto de datos original del 75.6 %, mientras que la peor solución encontrada obtuvo aproximadamente un 28.24 % de porcentaje de clasificación, sin embargo, esta solución se obtuvo utilizando solo 1 característica, lo cual implica una reducción del conjunto de datos de aproximadamente el 99 %. Las demás soluciones del frente acumulado se ubican entre los dos extremos mencionados, las cuales pueden observarse gráficamente en la Fig. 5.2b.

Ambos algoritmos generaron soluciones con un alto porcentaje de clasificación utilizando un número de características diferente. Basándonos en la precisión del algoritmo de clasificación, FSNSGA II da como resultado 1 solución con una precisión mayor a la obtenida con FSGDE 3, sin embargo, esta solución requiere de 22 características para lograr este resultado, mientras que FSGDE 3 solo requiere de 17 características para obtener un resultado similar. Revisando la información de la tabla 5.8, podemos observar que las muestras resultantes de las 30 ejecuciones del conjunto de datos utilizando FSGDE 3 y FSNSGA II nos dice que ambos algoritmos se comportan de manera similar, por lo que cualquier de los dos puede generar soluciones óptimas para nuestro problema.

Conjunto de datos semeion

Utilizando el conjunto de datos semeion, el cual contiene 1593 instancias, 256 características y 10 clases, se realizaron 30 ejecuciones de FSGDE 3 y FSNSGA II para selección de características.

FSGDE 3

FSGDE 3 generó un frente acumulado con un total de 56 soluciones, las cuales se muestran en la Fig. 5.3, así la representación gráfica del frente acumulado, el cual puede observarse en la Fig. 5.4.

Frete Acumulado		48.12%	5	8.87%	118
% error	Tamaño	23.92%	21	18.19%	35
31.00%	13	55.47%	3	15.59%	42
13.44%	54	45.52%	6	18.46%	30
32.80%	12	38.35%	9	11.47%	69
79.93%	1	9.50%	114	8.69%	137
50.09%	4	64.96%	2	7.71%	172
11.74%	60	10.84%	73	20.61%	26
16.31%	36	13.98%	51	16.22%	40
19.89%	28	10.04%	99	13.80%	52
21.68%	25	10.66%	83	9.68%	107
15.50%	44	14.61%	49	21.95%	24
37.19%	10	12.81%	56	23.48%	23
29.03%	16	10.22%	88	26.08%	17
7.97%	155	8.51%	146	20.07%	27

(a) Soluciones no Dominadas (b) Soluciones no Dominadas (c) Soluciones no Dominadas

10.13%	92
30.65%	15
9.77%	102
9.59%	108
11.56%	68
39.78%	8
35.04%	11
18.73%	29
15.86%	41
43.19%	7
14.70%	47
25.36%	20
18.28%	34

(d) Soluciones no Dominadas

Figura 5.3: Resultados de FSGDE 3 utilizando el conjunto de datos semeion.

En la tabla de la Fig. 5.3c puede observarse que la solución con un mayor porcentaje de clasificación fue la que obtuvo un 92.29 % de precisión utilizando 172 características de las 256 que contiene el conjunto de datos original, lo cual implicó una reducción del 32.8 %, mientras que la peor solución encontrada obtuvo aproximadamente un 20 % de precisión, sin embargo, esta solución fue obtenida utilizando solo 1 característica, lo cual implicó una

reducción del conjunto de datos original de aproximadamente 99.6 %. Las demás soluciones del frente acumulado se ubican entre los dos extremos mencionados, las cuales pueden observarse gráficamente en la Fig. 5.4.

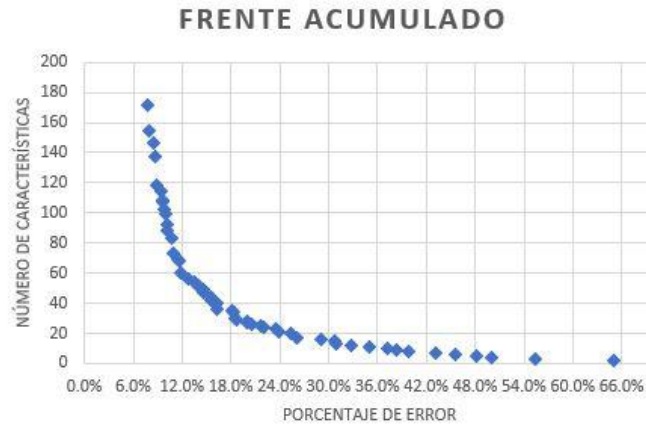


Figura 5.4: Frente acumulado del conjunto de datos Semeion

FSNSGA II

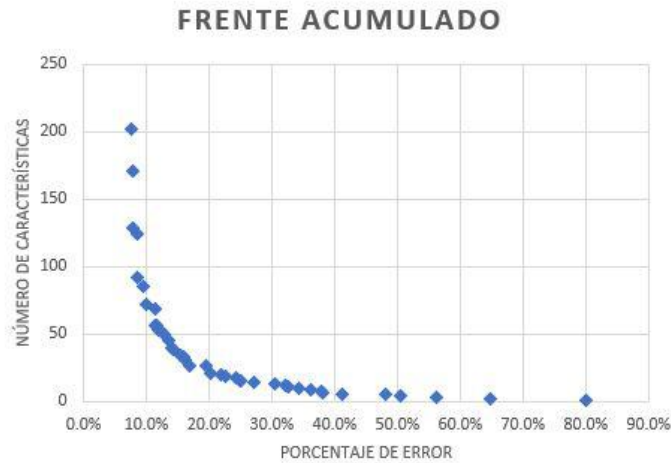
FSNSGA II generó un frente acumulado con un total de 41 soluciones, las cuales se muestran en la Fig. 5.5.

En la tabla de la Fig. 5.5c puede observarse que la solución con un mayor porcentaje de clasificación fue la que obtuvo un 92.38 % de precisión utilizando 202 características de las 256 que contiene el conjunto de datos, lo cual implicó una reducción del 21.6 %, mientras que la peor solución encontrada obtuvo aproximadamente un 20 % de precisión, esta solución al igual que la obtenida por el algoritmo FSGDE 3 fue evaluada utilizando solo 1 característica, lo cual implicó una reducción del conjunto de datos de aproximadamente 99.6 %. Las demás soluciones del frente acumulado obtenidas por el algoritmo se muestran gráficamente la Fig. 5.5d.

Los resultados de ambos algoritmos muestran una precisión similar utilizando un número de características distinto, aun cuando los resultados de la mejor solución encontrada por ambos algoritmos es prácticamente la misma, FSGDE 3 utiliza solo 172 características, 30 características menos que FSNSGA II, lo que ayuda a seleccionar FSGDE 3 como el mejor entre los dos algoritmos para el conjunto de datos seleccionado, esto debido a que cumple el objetivo de mantener un compromiso entre la precisión y la disminución del número de características seleccionadas. Basándonos los resultados obtenidos y a la prueba estadística de Wicolxon, la cual nos indica que existen diferencias significativas entre ambos, determinando que el algoritmo con el mejor desempeño para el conjunto de datos seleccionado es el algoritmo propuesto.

Frente Acumulado					
% error	Tamaño	36.20%	9		
41.13%	6	15.50%	34		
56.18%	3	30.56%	13		
22.67%	19	25.09%	15	34.32%	10
32.08%	12	13.98%	40	38.08%	7
20.25%	21	9.59%	85	24.37%	18
32.53%	11	24.91%	16	11.29%	69
50.45%	4	64.87%	2	14.16%	39
12.37%	52	37.81%	8	7.62%	202
80.02%	1	21.95%	20	11.83%	53
19.53%	26	8.60%	92	7.80%	171
27.15%	14	9.86%	72	7.89%	129
11.74%	56	12.81%	50	15.86%	33
48.12%	5	11.38%	57	8.42%	124
13.53%	45	16.22%	31	16.94%	27

(a) Soluciones no Dominadas (b) Soluciones no Dominadas (c) Soluciones no Dominadas



(d) Frente acumulado

Figura 5.5: Resultados de FSNSGA II utilizando el conjunto de datos semeion.

Conjunto de datos SRBCT

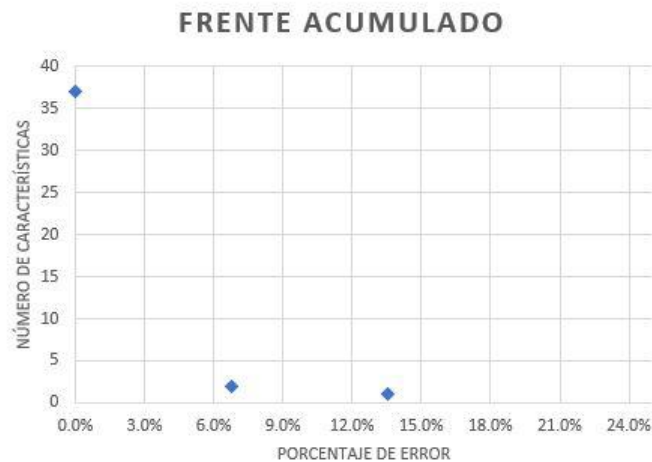
Se realizó una prueba utilizando el micro-arreglo SRBCT ya que contiene 83 instancias, 2308 características y 4 clases, se realizaron 30 ejecuciones de FSGDE 3 y FSNSGA II para selección de características.

FSGDE 3

Al finalizar las 30 ejecuciones de FSGDE 3, se obtuvieron solo 3 soluciones, las cuales se muestran en la Fig. 5.6

Frente Acumulado	
% error	Tamaño
6.78%	2
13.56%	1
0.00%	37

(a) Soluciones no Dominadas



(b) Frente acumulado

Figura 5.6: Resultados de FSGDE 3 utilizando el micro arreglo SRBCT.

El conjunto de soluciones del frente acumulado generado por FSGDE 3 consiste en 3 soluciones no dominadas, las cuales presentan una precisión del 100 % utilizando 37 características de las 2308 del micro arreglo original, lo que equivale a una reducción de 98.4 %, y una precisión del 86.44 % al utilizar solo 1 característica, lo cual implica una reducción del 99.9 %. La distribución de este conjunto de soluciones puede observarse gráficamente en la Fig. 5.6b.

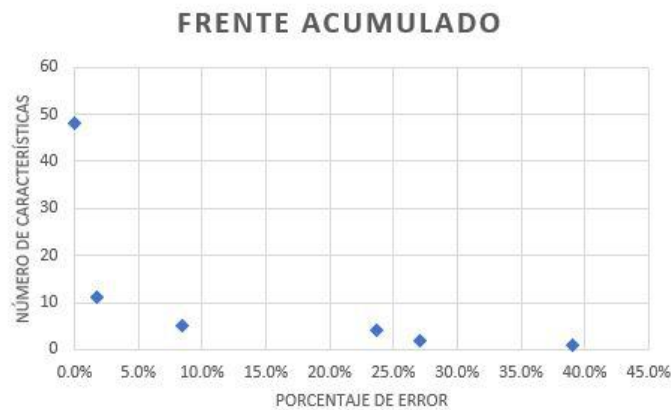
FSNSGA II

FSNSGA II utilizando el micro arreglo SRBCT generó 6 soluciones no dominadas, dichas soluciones se pueden observar en la Fig. 5.7 así como el frente acumulado generado.

En la tabla de la Fig. 5.7a se puede observar que al utilizar FSNSGA II se obtiene una precisión del 100 % utilizando 48 características del conjunto de datos original, el cual contiene 2308, implica una reducción del 97.7 %, mientras que al utilizar solo 1 característica y

Frente Acumulado	
% error	Tamaño
0.00%	48
8.47%	5
27.12%	2
23.73%	4
1.69%	11
38.98%	1

(a) Soluciones no Dominadas



(b) Frente acumulado

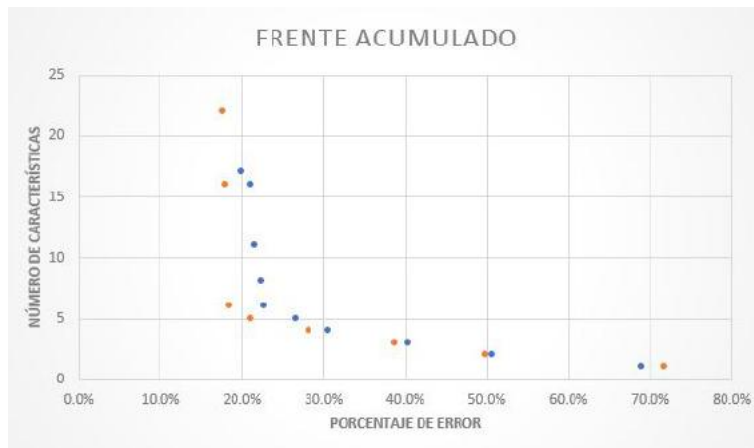
Figura 5.7: Resultados de FSNSGA II utilizando el micro arreglo SRBCT.

tener una reducción del 99.9 %, la precisión se reduce al 61 %. La distribución del conjunto de soluciones puede observarse en la gráfica de la Fig. 5.7b.

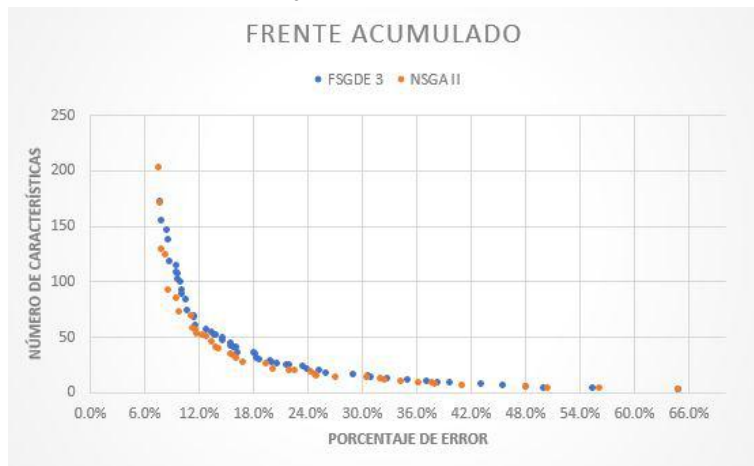
Para este conjunto de datos ambos algoritmos presentan una precisión del 100 % en una de sus soluciones, sin embargo FSNSGA II requiere de 48 características para obtener este porcentaje de clasificación, mientras que el algoritmo propuesto obtiene el mismo porcentaje utilizando solo 37 características.

En la Fig. 5.8 se pueden ver los frentes acumulados obtenidos por ambos algoritmos, como puede observarse, el patrón del Frente de Pareto obtenido es similar en ambos, por lo que puede determinarse que los conjuntos de soluciones encontradas con ambos algoritmos se encuentran dentro del Frente de Pareto del conjunto de datos seleccionado.

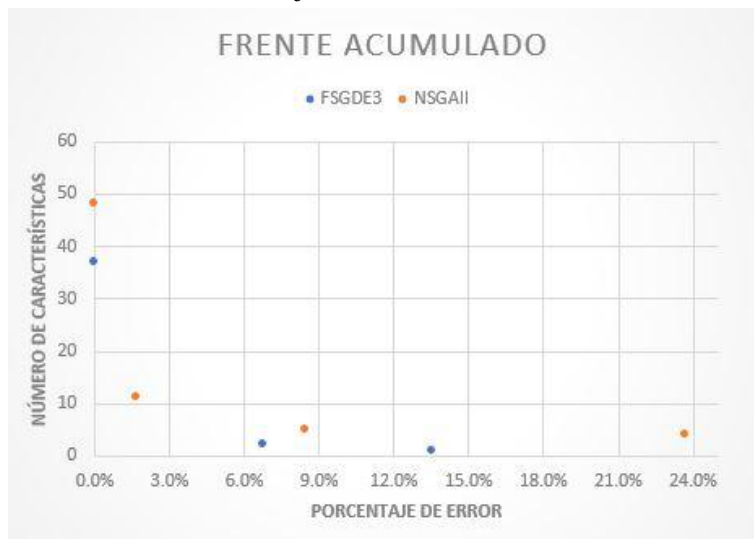
En el anexo A se pueden observar los 30 frentes de Pareto obtenidos en cada una de las ejecuciones realizadas a cada uno de los conjuntos de datos utilizados para este experimento con ambos algoritmos, FSGDE 3 y FSNSGA II.



(a) Conjunto de datos m-libras



(b) Conjunto de datos semeion



(c) Conjunto de datos SRBCT

Figura 5.8: Comparación del frente acumulado de FSGDE 3 y FSNSGA II.

5.5.2. Experimento 2

El experimento 2 consistió en una comparación entre FSGDE 3 utilizando uno de sus objetivos como métrica, el cual fue el porcentaje de error, contra el algoritmo DE-FS propuesto en [36] y sus resultados que muestran la comparación en precisión de algunos algoritmos tradicionales para selección de características. Para este experimento se utilizaron 18 conjuntos de datos, los cuales se muestran en la tabla 5.10.

Tabla 5.10: Conjunto de datos utilizados en el segundo experimento

Dataset	Instancias	Características	Clases	Dataset	Instancias	Características	Clases
arrhythmia	452	279	16	audiology	226	69	24
australian	690	14	2	crx	690	15	2
cylinder-b	540	39	2	german-c	1000	20	2
dermatology	366	34	6	ionosphere	351	34	2
h-valley	1212	100	2	m-libras	360	90	15
musk-1	476	168	2	parkinsons	195	22	2
sonar	208	60	2	soybean	683	35	19
spectf	267	44	2	vehicle	846	18	4
vote	435	16	2	wdbc	569	30	2

Se realizaron 30 ejecuciones para cada uno de los conjuntos de datos utilizando el conjunto de datos completo, el cual nos genero un conjunto de soluciones no dominadas, de las cuales se tomo la solución con el menor porcentaje de error, el mayor porcentaje de error y una solución con un valor intermedio entre la mejor y la peor solución, todo esto basándose en el porcentaje de error como función de calidad a comparar. Esta comparación se realizó entre las soluciones de FSGDE 3 y los resultados obtenido por DE-FS, los algoritmos tradicionales de selección de características, SFS y SBE así como el resultado del porcentaje de error del conjunto de datos sin selección de características.

Mejor Solución

Utilizando la mejor solución obtenida con cada uno de los conjuntos de datos utilizados y con la cual se realizo la comparación entre este valor y los resultados de los algoritmos descritos anteriormente, se realizó una tabla donde se muestra el porcentaje de clasificación y el rango que ocupa entre los 5 algoritmos analizados, así mismo se muestra el promedio del rango obtenido y las veces que cada una de las soluciones gano, empato o perdió (GEP) con respecto a las demás.

Tabla 5.11: Tabla de resultados obtenidos utilizando la mejor solución

Conjunto de Datos	Sin selección		SFS		SBE		DE-FS		FSGDE 3	
	Precisión	Rango	Precisión	Rango	Precisión	Rango	Precisión	Rango	Precisión	Rango
arrhythmia	58.92 %	5	73.73 %	2	63.72 %	4	76.19 %	1	71.02 %	3
audiology	76.29 %	4	83.83 %	3	84.28 %	2	85.43 %	1	73.89 %	5
australian	85.55 %	4	86.64 %	3	84.28 %	5	87.02 %	2	88.99 %	1
cylinder-b	73.68 %	5	83.31 %	2	79.57 %	3	85.09 %	1	77.99 %	4
crx	86.41 %	4	86.03 %	5	87.38 %	3	87.83 %	2	89.13 %	1
dermatology	96.98 %	5	98.37 %	2.5	97.94 %	4	98.37 %	2.5	98.91 %	1
german-c	75.11 %	5	76.73 %	2	75.81 %	4	77.22 %	1	76.40 %	3
h-valley	65.05 %	4	70.06 %	2	68.35 %	3	72.04 %	1	63.53 %	5
ionosphere	86.82 %	5	92.67 %	3	90.89 %	4	94.86 %	1	93.73 %	2
m-libras	86.13 %	4	88.85 %	2	87.53 %	3	89.57 %	1	83.06 %	5
musk-1	85.63 %	5	90.51 %	4	92.19 %	3	94.64 %	2	99.37 %	1
parkinsons	95.92 %	4	95.08 %	5	98.11 %	2	98.61 %	1	97.95 %	3
sonar	86.53 %	5	89.47 %	4	91.08 %	3	93.77 %	1	91.83 %	2
soybean	91.63 %	5	95.75 %	2	94.57 %	3	95.85 %	1	92.68 %	4
spectf	77.69 %	5	80.78 %	3	80.17 %	4	84.88 %	2	85.39 %	1
vehicle	70.44 %	5	71.26 %	4	72.96 %	3	74.44 %	2	76.00 %	1
vote	93.58 %	5	95.63 %	4	95.98 %	3	96.45 %	2	96.78 %	1
wdbc	97.01 %	5	97.26 %	4	97.47 %	3	97.64 %	2	98.24 %	1
GEP		0 - 0 - 18		0 - 1 - 17		0 - 0 - 18		10 - 1 - 7		8 - 0 - 10
Promedio de Rangos		4.78		3.14		3.22		1.47		2.39

En la tabla 5.11 puede observarse que los resultados del algoritmo propuesto fue mejor que los demás en solo 8 ocasiones y perdió en 10 conjuntos de datos, para los cuales el algoritmo DE-FS obtuvo el mejor resultado en cuanto a precisión (menor porcentaje de error). No obstante estos resultados son tomados analizando únicamente la función de calidad del menor porcentaje de error, y dado que el algoritmo propuesto es multiobjetivo se realizó una comparación de la reducción al conjunto de datos original, esto con la finalidad de observar por que resultado perdedor en 10 de los casos descritos en la tabla 5.11. Esta comparación entre la reducción de cada uno de los algoritmos para selección de características se muestra en la tabla 5.12.

Tabla 5.12: Reducción de dimensionalidad de los conjuntos de datos.

Conjunto de Datos	Características	SFS		SBE		DE-FS		FSGDE 3	
		Tamaño	Reducción	Tamaño	Reducción	Tamaño	Reducción	Tamaño	Reducción
arrhythmia	279	14	95.0 %	126	54.8 %	10.5	96.2 %	8	97.1 %
audiology	69	13.7	80.1 %	27.8	59.7 %	23.7	65.7 %	43	37.7 %
australian	14	3.1	77.9 %	10.2	27.1 %	8.5	39.3 %	4	71.4 %
cylinder-b	39	5.4	86.2 %	26.2	32.8 %	5.8	85.1 %	12	69.2 %
crx	15	2	86.7 %	10.8	28.0 %	7	53.3 %	4	73.3 %
dermatology	34	12.5	63.2 %	21.3	37.4 %	21.2	37.6 %	26	23.5 %
german-c	20	6.5	67.5 %	17.4	13.0 %	9.9	50.5 %	8	60.0 %
h-valley	100	9.1	90.9 %	79.2	20.8 %	10.9	89.1 %	5	95.0 %
ionosphere	34	4.3	87.4 %	25.7	24.4 %	7.2	78.8 %	4	88.2 %
m-libras	90	11.4	87.3 %	69.5	22.8 %	19.6	78.2 %	17	81.1 %
musk-1	168	14.1	91.6 %	102.2	39.2 %	35.8	78.7 %	4	97.6 %
parkinsons	22	6.6	70 %	12.6	42.7 %	10.3	53.2 %	6	72.7 %
sonar	60	10	83.3 %	44.6	25.7 %	20.4	66.0 %	31	48.3 %
soybean	35	13.8	60.6 %	20.2	42.3 %	14.7	58.0 %	26	25.7 %
spectf	44	2	95.5 %	38.5	12.5 %	5.3	88.0 %	2	95.5 %
vehicle	18	6.5	63.9 %	14.4	20.0 %	9.8	45.6 %	8	55.6 %
vote	16	1	93.8 %	9.1	43.1 %	5.5	65.6 %	8	50.0 %
wdbc	30	6.3	79.0 %	22.9	23.7 %	16.8	44 %	7	76.7 %

Como se puede observar en la tabla 5.12, el algoritmo que realiza un mejor porcentaje de reducción de dimensionalidad es SFS, ya que en 12 de los conjuntos de datos utilizados

mostró una reducción superior, en 1 de los conjuntos de datos alcanzo un empate con FSG-DE 3 y en solo 5 conjuntos de datos fue superado por FSGDE 3. No obstante, aun teniendo el mejor porcentaje de reducción de características en la mayoría de los conjuntos de datos, su porcentaje de clasificación fue en menor al obtenido por el algoritmo DE-FS.

Contando los resultados de las tablas 5.11 y 5.12 se puede determinar que el algoritmo multiobjetivo propuesto genera soluciones con un porcentaje de clasificación similar a los generados por los algoritmos tradicionales y evolutivos mono-objetivo.

En 10 de los conjuntos de datos utilizados obtuvo una precisión menor, debido a que el algoritmo propuesto busca un compromiso entre ambos objetivos, la reducción de dimensionalidad y el porcentaje de clasificación.

Peor Solución

Utilizando la peor solución del conjunto de soluciones no dominadas generada por el algoritmo FSGDE 3 con cada uno de los conjuntos de datos utilizados se realizó la comparación entre este valor y los resultados de los algoritmos definidos anteriormente en la tabla 5.11, se realizó una actualización del porcentaje de clasificación y el rango que ocupa entre los 5 algoritmos analizados, así mismo se muestra el promedio del rango obtenido y las veces que cada una de las soluciones Gano, Empato o Perdió (GEP) con respecto a las demás.

Tabla 5.13: Tabla de resultados obtenidos utilizando la peor solución

Conjunto de Datos	Sin selección		SFS		SBE		DE-FS		FSGDE 3	
	Precisión	Rango	Precisión	Rango	Precisión	Rango	Precisión	Rango	Precisión	Rango
arrhythmia	58.92 %	5	73.73 %	2	63.72 %	3	76.19 %	1	60.40 %	4
audiology	76.29 %	4	83.83 %	3	84.28 %	2	85.43 %	1	46.46 %	5
australian	85.55 %	3	86.64 %	2	84.28 %	5	87.02 %	1	85.51 %	4
cylinder-b	73.68 %	4	83.31 %	2	79.57 %	3	85.09 %	1	71.30 %	5
crx	86.41 %	3	86.03 %	4	87.38 %	2	87.83 %	1	85.51 %	5
dermatology	96.98 %	4	98.37 %	1.5	97.94 %	3	98.37 %	1.5	50.27 %	5
german-c	75.11 %	4	76.73 %	2	75.81 %	3	77.22 %	1	71.70 %	5
h-valley	65.05 %	4	70.06 %	2	68.35 %	3	72.04 %	1	57.10 %	5
ionosphere	86.82 %	4	92.67 %	2	90.89 %	3	94.86 %	1	85.47 %	5
m-libras	86.13 %	4	88.85 %	2	87.53 %	3	89.57 %	1	26.67 %	5
musk-1	85.63 %	4	90.51 %	3	92.19 %	2	94.64 %	1	72.90 %	5
parkinsons	95.92 %	3	95.08 %	4	98.11 %	2	98.61 %	1	89.74 %	5
sonar	86.53 %	4	89.47 %	3	91.08 %	2	93.77 %	1	74.04 %	5
soybean	91.63 %	4	95.75 %	2	94.57 %	3	95.85 %	1	74.04 %	5
spectf	77.69 %	5	80.78 %	2	80.17 %	3	84.88 %	1	81.65 %	4
vehicle	70.44 %	4	71.26 %	3	72.96 %	2	74.44 %	1	54.26 %	5
vote	93.58 %	5	95.63 %	3.5	95.98 %	2	96.45 %	1	95.63 %	3.5
wdbc	97.01 %	4	97.26 %	3	97.47 %	2	97.64 %	1	92.27 %	5
GEP		0 - 0 - 18		0 - 2 - 16		0 - 0 - 18		17 - 1 - 0		0 - 1 - 17
Promedio de Rangos		4.11		2.53		2.61		1.03		4.72

En la tabla 5.13 puede observarse que los resultados del algoritmo propuesto utilizando la peor de las soluciones encontrada fue el peor porcentaje de clasificación y en una ocasión obtuvo un empate con el algoritmo SFS. Por otro lado el algoritmo DE-FS obtuvo el mejor resultado en cuanto a precisión (menor porcentaje de error) en 17 de los 18 conjuntos de datos utilizados, sin embargo se realiza una comparación entre la reducción de cada uno de los algoritmos para selección de características como se muestra en la tabla 5.14.

Tabla 5.14: Reducción de dimensionalidad de los conjuntos de datos.

Conjunto de Datos	Características	SFS		SBE		DE-FS		FSGDE 3	
		Tamaño	Reducción	Tamaño	Reducción	Tamaño	Reducción	Tamaño	Reducción
arrhythmia	279	14	95.0 %	126	54.8 %	10.5	96.2 %	1	99.6 %
audiology	69	13.7	80.1 %	27.8	59.7 %	23.7	65.7 %	1	98.6 %
australian	14	3.1	77.9 %	10.2	27.1 %	8.5	39.3 %	1	92.9 %
cylinder-b	39	5.4	86.2 %	26.2	32.8 %	5.8	85.1 %	1	97.4 %
crx	15	2	86.7 %	10.8	28.0 %	7	53.3 %	1	93.3 %
dermatology	34	12.5	63.2 %	21.3	37.4 %	21.2	37.6 %	1	97.1 %
german-c	20	6.5	67.5 %	17.4	13.0 %	9.9	50.5 %	1	95.0 %
h-valley	100	9.1	90.9 %	79.2	20.8 %	10.9	89.1 %	1	99.0 %
ionosphere	34	4.3	87.4 %	25.7	24.4 %	7.2	78.8 %	1	97.4 %
m-libras	90	11.4	87.3 %	69.5	22.8 %	19.6	78.2 %	1	98.9 %
musk-1	168	14.1	91.6 %	102.2	39.2 %	35.8	78.7 %	1	99.4 %
parkinsons	22	6.6	70 %	12.6	42.7 %	10.3	53.2 %	1	95.5 %
sonar	60	10	83.3 %	44.6	25.7 %	20.4	66.0 %	1	98.3 %
soybean	35	13.8	60.6 %	20.2	42.3 %	14.7	58.0 %	1	97.1 %
spectf	44	2	95.5 %	38.5	12.5 %	5.3	88.0 %	1	97.7 %
vehicle	18	6.5	63.9 %	14.4	20.0 %	9.8	45.6 %	1	94.4 %
vote	16	1	93.8 %	9.1	43.1 %	5.5	65.6 %	1	93.8 %
wdbc	30	6.3	79.0 %	22.9	23.7 %	16.8	44 %	1	96.7 %

Como se puede observar en la tabla 5.14, a pesar de que en cuanto a precisión, el algoritmo FSGDE 3 fue el peor en todos los casos, sin embargo también es quien realiza un mejor porcentaje de reducción de dimensionalidad ya que en todos los conjuntos de datos solo utilizo 1 característica para realizar la clasificación.

Contando los resultados de las tablas 5.13 y 5.14 se puede determinar que el algoritmo multiobjetivo propuesto genera soluciones con un porcentaje de clasificación bajo en comparación con los algoritmos tradicionales y evolutivos mono-objetivo. Esto debido a que el algoritmo propuesto considero una solución utilizando al menos 1 característica lo que redujo considerablemente su porcentaje de clasificación.

Punto Rodilla - Solución Intermedia

Utilizando la solución intermedia del conjunto de soluciones no dominadas generada por el algoritmo FSGDE 3, la cual llamamos punto rodilla con cada uno de los conjuntos de datos utilizados, se realizó la comparación entre este valor y los resultados de los algoritmos definidos anteriormente, se realizo una actualización del porcentaje de clasificación y el rango que ocupa entre los 5 algoritmos analizados, así mismo se muestra el promedio del rango obtenido y las veces que cada una de las soluciones Gano, Empato o Perdió (GEP) con respecto a las demás.

Tabla 5.15: Tabla de resultados obtenidos utilizando la solución intermedia o punto rodilla

Conjunto de Datos	Sin selección		SFS		SBE		DE-FS		FSGDE 3	
	Precisión	Rango	Precisión	Rango	Precisión	Rango	Precisión	Rango	Precisión	Rango
arrhythmia	58.92 %	5	73.73 %	2	63.72 %	4	76.19 %	1	69.25 %	3
audiology	76.29 %	4	83.83 %	3	84.28 %	2	85.43 %	1	70.80 %	5
australian	85.55 %	3	86.64 %	2	84.28 %	5	87.02 %	2	87.83 %	1
cylinder-b	73.68 %	5	83.31 %	2	79.57 %	3	85.09 %	1	75.74 %	4
crx	86.41 %	4	86.03 %	5	87.38 %	3	87.83 %	2	88.26 %	1
dermatology	96.98 %	5	98.37 %	1.5	97.94 %	3	98.37 %	1.5	96.99 %	4
german-c	75.11 %	5	76.73 %	2	75.81 %	3	77.22 %	1	75.20 %	4
h-valley	65.05 %	4	70.06 %	2	68.35 %	3	72.04 %	1	63.28 %	5
ionosphere	86.82 %	5	92.67 %	2	90.89 %	4	94.86 %	1	92.59 %	3
m-libras	86.13 %	4	88.85 %	2	87.53 %	3	89.57 %	1	75.83 %	5
musk-1	85.63 %	5	90.51 %	4	92.19 %	3	94.64 %	2	98.53 %	1
parkinsons	95.92 %	4	95.08 %	5	98.11 %	2	98.61 %	1	96.92 %	3
sonar	86.53 %	5	89.47 %	3	91.08 %	2	93.77 %	1	89.42 %	4
soybean	91.63 %	4	95.75 %	2	94.57 %	3	95.85 %	1	86.97 %	5
spectf	77.69 %	5	80.78 %	3	80.17 %	4	84.88 %	1	83.90 %	2
vehicle	70.44 %	5	71.26 %	4	72.96 %	3	74.44 %	2	74.59 %	1
vote	93.58 %	5	95.63 %	4	95.98 %	2	96.45 %	1	95.86 %	3
wdbc	97.01 %	5	97.26 %	4	97.47 %	3	97.64 %	2	97.89 %	1
GEP		0 - 0 - 18		0 - 1 - 17		0 - 0 - 18		12 - 1 - 5		5 - 0 - 13
Promedio de Rangos		4.67		3.03		3.06		1.31		2.94

En la tabla 5.15 puede observarse que los resultados del algoritmo propuesto utilizando las soluciones intermedias de cada conjunto de soluciones no dominadas de cada uno de los conjuntos de datos obtuvo un porcentaje de clasificación similar al de los otros algoritmos y en 5 ocasiones fue superior. Por otro lado al realizar una comparación entre la reducción de cada uno de los algoritmos para selección de características se obtuvieron los resultados mostrados en la tabla 5.16.

Tabla 5.16: Reducción de dimensionalidad de los conjuntos de datos.

Conjunto de Datos	Características	SFS		SBE		DE-FS		FSGDE 3	
		Tamaño	Reducción	Tamaño	Reducción	Tamaño	Reducción	Tamaño	Reducción
arrhythmia	279	14	95.0 %	126	54.8 %	10.5	96.2 %	3	98.9 %
audiology	69	13.7	80.1 %	27.8	59.7 %	23.7	65.7 %	4	94.2 %
australian	14	3.1	77.9 %	10.2	27.1 %	8.5	39.3 %	3	78.6 %
cylinder-b	39	5.4	86.2 %	26.2	32.8 %	5.8	85.1 %	2	94.9 %
crx	15	2	86.7 %	10.8	28.0 %	7	53.3 %	4	73.3 %
dermatology	34	12.5	63.2 %	21.3	37.4 %	21.2	37.6 %	11	67.6 %
german-c	20	6.5	67.5 %	17.4	13.0 %	9.9	50.5 %	3	85.0 %
h-valley	100	9.1	90.9 %	79.2	20.8 %	10.9	89.1 %	3	97.0 %
ionosphere	34	4.3	87.4 %	25.7	24.4 %	7.2	78.8 %	2	94.1 %
m-libras	90	11.4	87.3 %	69.5	22.8 %	19.6	78.2 %	5	94.4 %
musk-1	168	14.1	91.6 %	102.2	39.2 %	35.8	78.7 %	3	98.2 %
parkinsons	22	6.6	70 %	12.6	42.7 %	10.3	53.2 %	4	81.8 %
sonar	60	10	83.3 %	44.6	25.7 %	20.4	66.0 %	5	91.7 %
soybean	35	13.8	60.6 %	20.2	42.3 %	14.7	58.0 %	9	74.3 %
spectf	44	2	95.5 %	38.5	12.5 %	5.3	88.0 %	2	95.5 %
vehicle	18	6.5	63.9 %	14.4	20.0 %	9.8	45.6 %	6	66.7 %
vote	16	1	93.8 %	9.1	43.1 %	5.5	65.6 %	2	87.5 %
wdbc	30	6.3	79.0 %	22.9	23.7 %	16.8	44 %	4	86.7 %

Como se puede observar en las tablas 5.16 y 5.16, el algoritmo FSGDE 3 en la mayoría de los casos obtuvo un mejor porcentaje de reducción de dimensionalidad, sin embargo solo en 5 conjuntos de datos obtuvo la mejor precisión, esto debido a que el algoritmo propuesto busca aumentar la precisión intentando reducir el número de características seleccionadas, mientras que el algoritmo DE-FS solo se enfoca en incrementar la precisión.

Basado en los resultados obtenidos en ambos experimentos, se puede observar que el desempeño del algoritmo propuesto fue bueno. El primer experimento, el cual realizo una comparación entre FSGDE 3 y FSNSGA II, obtuvo resultados que mostraron una mayor reducción de dimensionalidad por el algoritmo FSGDE 3, lo cual afecto su precisión y debido a ésto, en la mayoría de los resultados la precisión obtenida por FSNSGA II fue mejor. sin embargo , en conjunto de datos grandes como lo son los micro arreglos, FSGDE 3, obtuvo un mejor resultado, ya que en 4 de 5 micro arreglos logro un empate y en 1, fue mejor. A pesar de esto, los conjuntos de soluciones obtenidas por ambos algoritmos fueron muy similares. Después de realizar la prueba estadística de wilcoxon utilizando los resultados proporcionados por los indicadores de desempeño Spacing e Hipervolumen, se observo que en 5 de los casos de estudio (conjuntos de datos), FSGDE 3 fue mejor, en 2 FSNSGA II fue mejor y en 20 de los 25 conjuntos de datos utilizados, los resultados arrojaron que cualquier de los dos algoritmos proporciona conjuntos de soluciones óptimas, sin embargo, basado en el tiempo de procesamiento, se considera FSGDE 3 mejor que FSNSGA II.

Este mismo comportamiento puede observarse en el experimento 2, ésto debido a que FSGDE 3 busca un compromiso entre la reducción de dimensionalidad manteniendo la mayor precisión posible mientras que el algoritmo DE-FS solo busca obtener la mejor precisión posible, por ello los resultados obtenidos en su mayoría fueron menores que los obtenidos por DE-FS, estos resultados pueden observarse claramente en la tabla 5.11, dado que al utilizar la mejor solución del conjunto de soluciones obtenidas por FSGDE 3, solo en 8 de 18 casos de estudio se obtuvo una precisión mayor que la obtenida por DE-FS, también en comparación con DE-FS, se obtuvo una mejor reducción de dimensionalidad en 13 de los 18 casos de estudio realizados. Sin embargo, al realizar la comparación con el algoritmo tradicional SFS, la reducción de dimensionalidad fue mejor en 12 de los 18 casos estudiados como puede observarse en la tabla 5.12.

Capítulo 6

Conclusiones y Trabajo futuro

En este trabajo de tesis se propuso el diseño de un algoritmo evolutivo multiobjetivo para selección de características utilizando un método de envoltorio, para lo cual se utilizó el algoritmo GDE 3 para realizar su implementación, esto manteniendo el compromiso entre la disminución del error de calificación y la reducción del número de características seleccionadas.

6.1. Conclusiones

Dado que el algoritmo FSGDE 3 ha mostrado ser competitivo para la solución de problemas de selección de características e incluso mejor que otros algoritmos utilizados para su comparación, como el algoritmo multiobjetivo NSGA II, el algoritmo mono-objetivo de Evolución Diferencial y algunos algoritmos tradicionales se puede concluir que el objetivo general del trabajo se cumplió.

Los experimentos realizados utilizando un conjunto de datasets demostraron que basado en los resultados, en el primer experimento, el algoritmo FSGDE 3 generó un conjunto de soluciones óptimas para la mayoría de los conjuntos de datos en comparación con el algoritmo multiobjetivo NSGA II mientras que en el segundo experimento, el cual consistió en una comparación con un algoritmo evolutivo mono-objetivo y algunos algoritmos tradicionales, generó soluciones cercanas a las obtenidas por el algoritmo mono-objetivo, esto debido a que el algoritmo propuesto también considera el número de características como uno de sus objetivos, lo cual complica su competitividad con los algoritmos mono-objetivo.

Después de los experimentos realizados, de los resultados obtenidos en cada experimento y de las pruebas estadísticas aplicadas, se puede validar que la hipótesis planteada en el capítulo 1 es correcta, debido a que el algoritmo FSGDE 3 logró generar soluciones similares y mejores que las obtenidas por algoritmos tradicionales para la selección de características manteniendo un menor número de características seleccionadas y manteniendo el porcentaje de clasificación.

6.2. Trabajo Futuro

A partir de la realización de este trabajo de tesis se obtuvieron algunos desafíos para trabajos futuros los cuales se mencionan a continuación:

- Debido a que al aumentar el tamaño del conjunto de datos, el tiempo de ejecución aumenta se propondría implementar el algoritmo FSGDE 3 utilizando otra forma de representación así como otro enfoque de clasificación para comparar los tiempos de ejecución utilizando los mismos conjuntos de datos analizados.
- Comparar FSGDE 3 con otros algoritmos multiobjetivo como Particle swarm optimization (PSO) y algunas de sus variantes para el problema de selección de características.

Anexos

Capítulo A

Anexos

En este apartado se muestran los 30 frentes de Pareto generados por FSGDE 3 y FSNS-GA II con los 25 conjuntos de datos utilizados para el experimento 1.

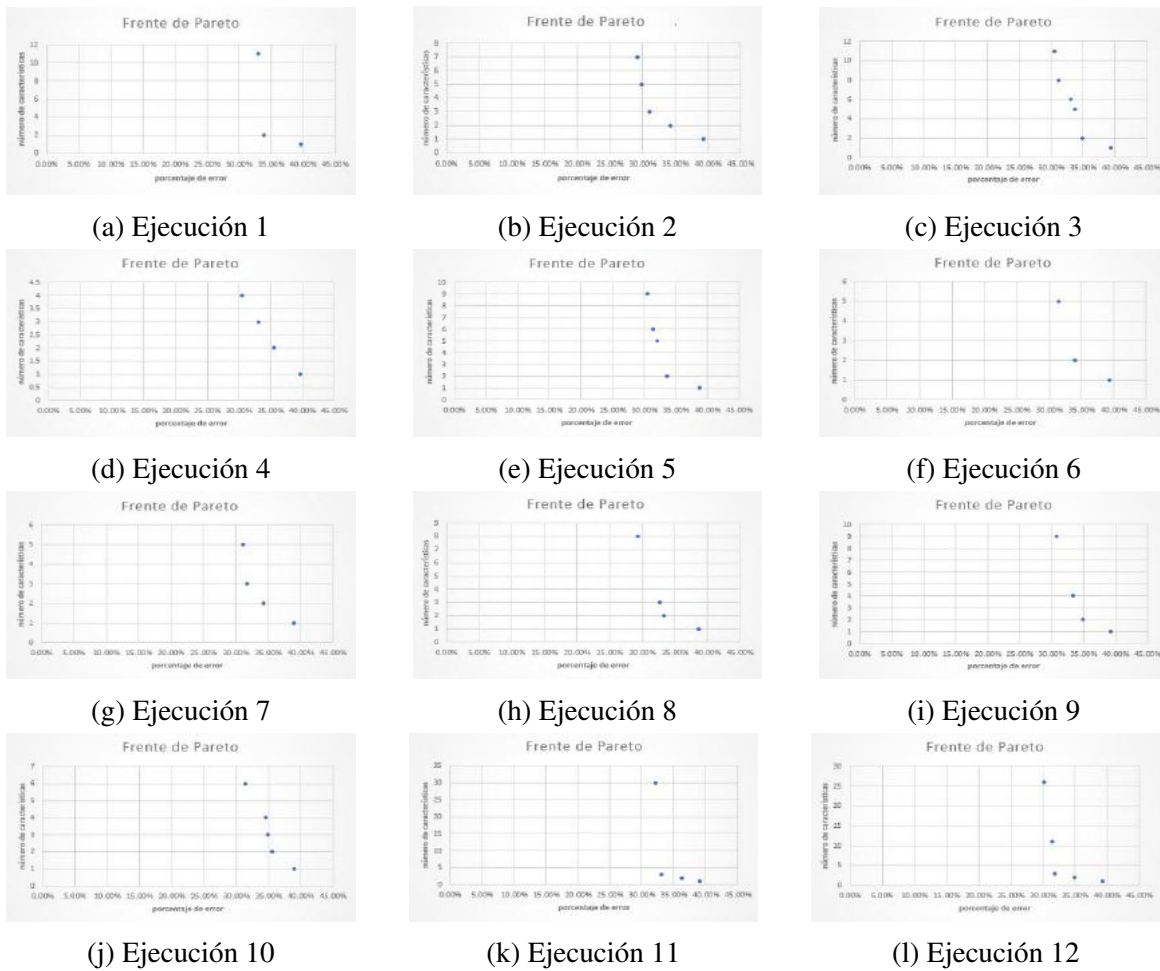
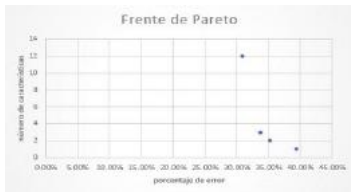
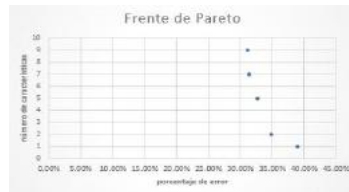


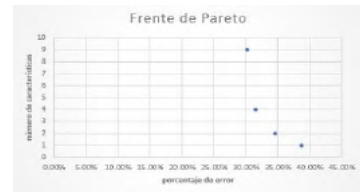
Figura A.1: Frentes de Pareto del Conjunto de datos arrhythmia con FSGDE 3. (1 - 2)



(a) Ejecución 13



(b) Ejecución 14



(c) Ejecución 15



(d) Ejecución 16



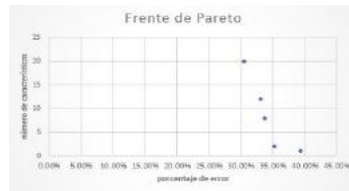
(e) Ejecución 17



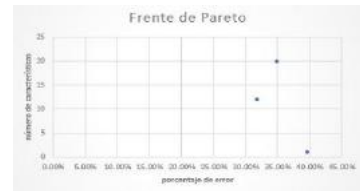
(f) Ejecución 18



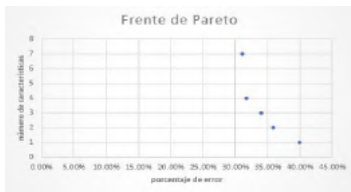
(g) Ejecución 19



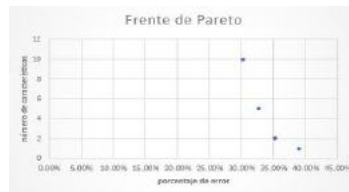
(h) Ejecución 20



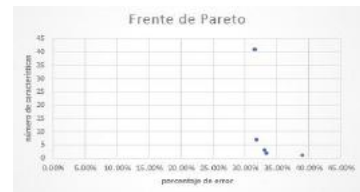
(i) Ejecución 21



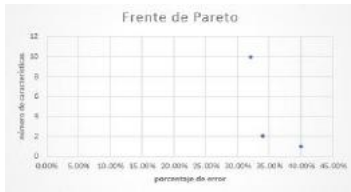
(j) Ejecución 22



(k) Ejecución 23



(l) Ejecución 24



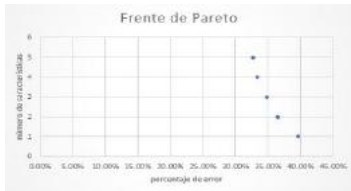
(m) Ejecución 25



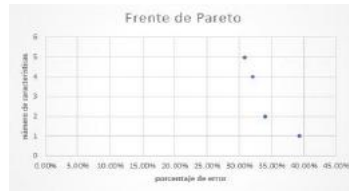
(n) Ejecución 26



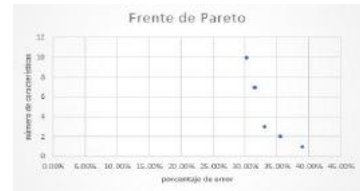
(ñ) Ejecución 27



(o) Ejecución 28

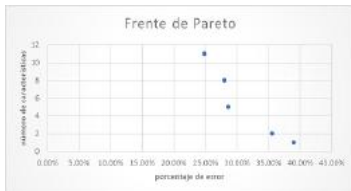


(p) Ejecución 29



(q) Ejecución 30

Figura A.2: Frentes de Pareto del Conjunto de datos arrhythmia con FSGDE 3. (2 - 2)



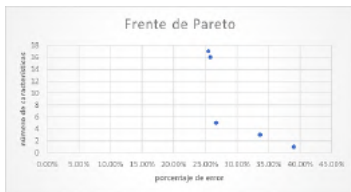
(a) Ejecución 1



(b) Ejecución 2



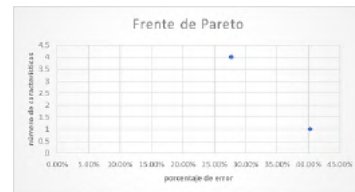
(c) Ejecución 3



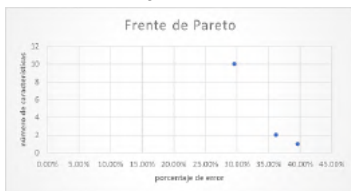
(d) Ejecución 4



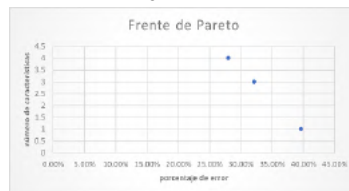
(e) Ejecución 5



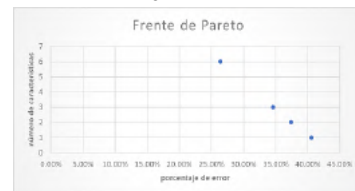
(f) Ejecución 6



(g) Ejecución 7



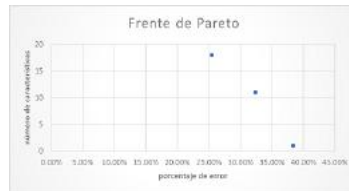
(h) Ejecución 8



(i) Ejecución 9



(j) Ejecución 10



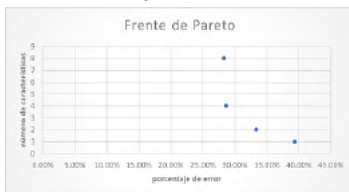
(k) Ejecución 11



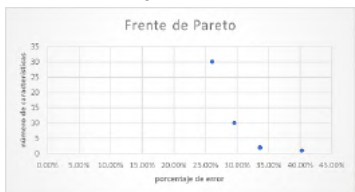
(l) Ejecución 12



(m) Ejecución 13



(n) Ejecución 14

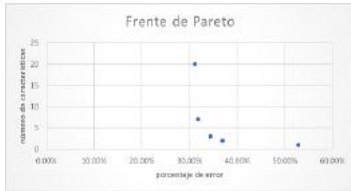


(ñ) Ejecución 15

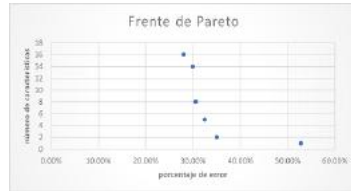
Figura A.3: Frentes de Pareto del Conjunto de datos arrhythmia con FSNSGA II. (1 - 2)



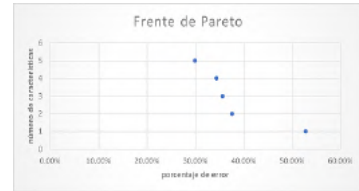
Figura A.4: Frentes de Pareto del Conjunto de datos arrhythmia con FSNSGA II. (2 - 2)



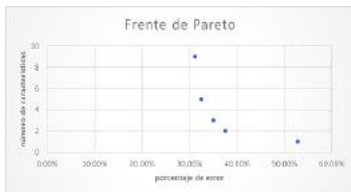
(a) Ejecución 1



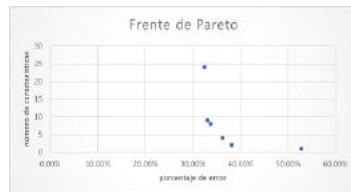
(b) Ejecución 2



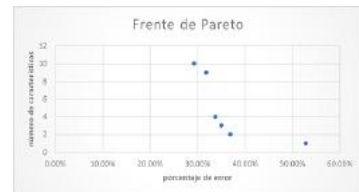
(c) Ejecución 3



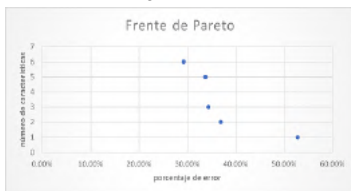
(d) Ejecución 4



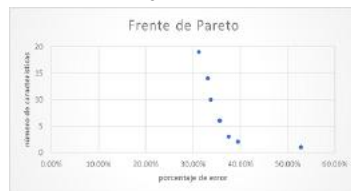
(e) Ejecución 5



(f) Ejecución 6



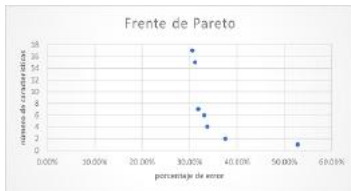
(g) Ejecución 7



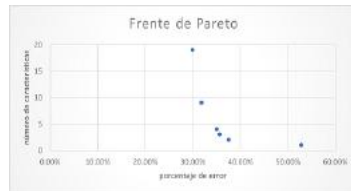
(h) Ejecución 8



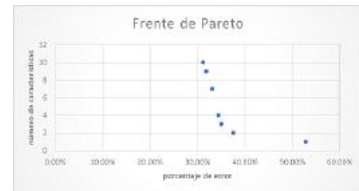
(i) Ejecución 9



(j) Ejecución 10



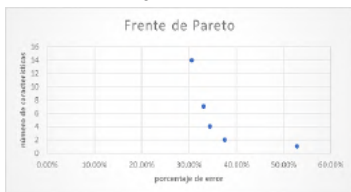
(k) Ejecución 11



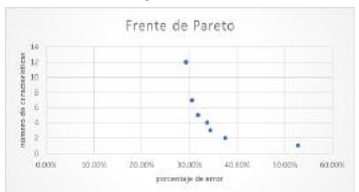
(l) Ejecución 12



(m) Ejecución 13



(n) Ejecución 14



(ñ) Ejecución 15

Figura A.5: Frentes de Pareto del Conjunto de datos audiology con FSGDE3. (1 - 2)

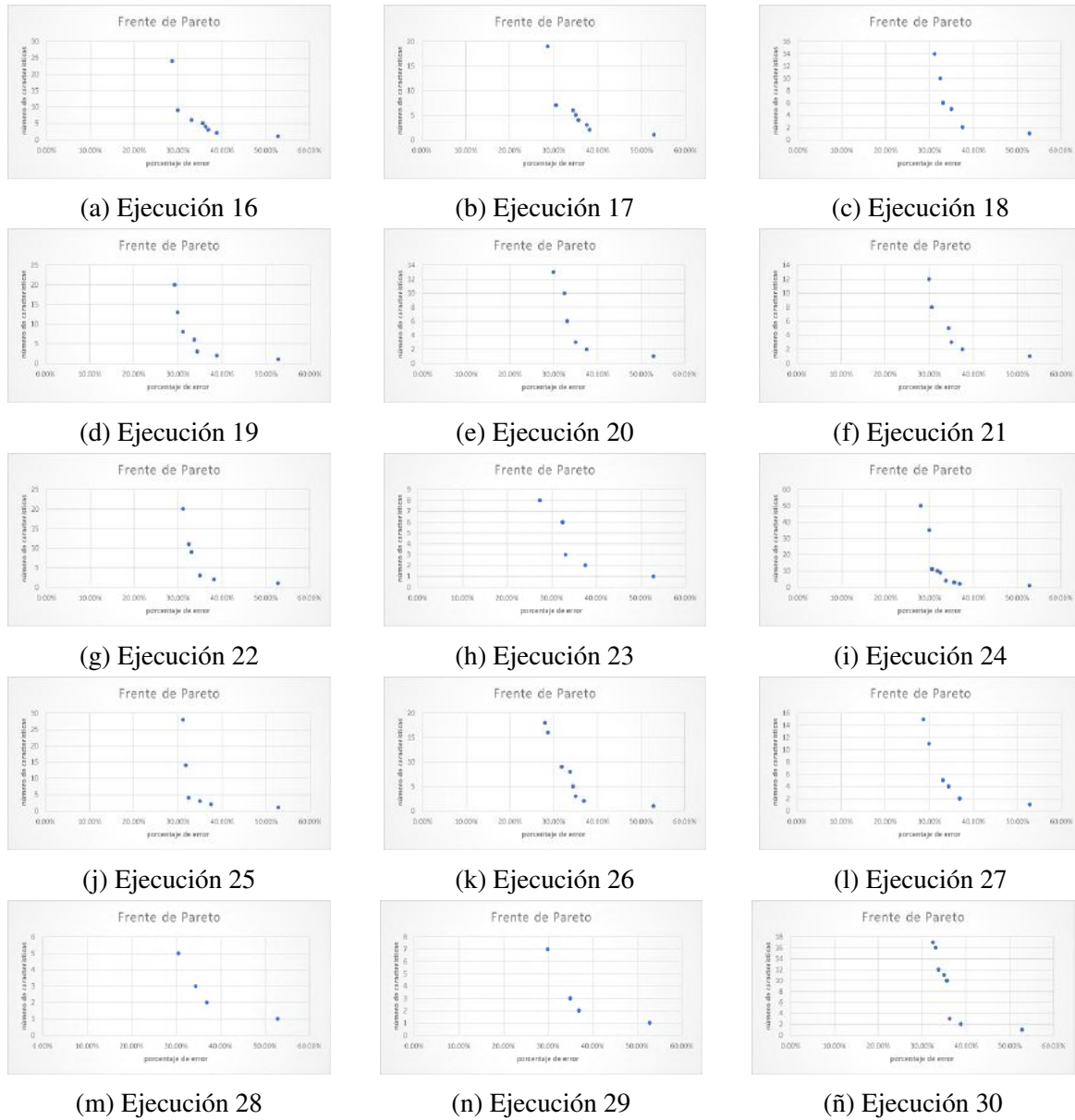
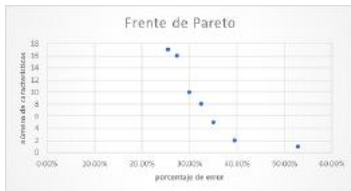
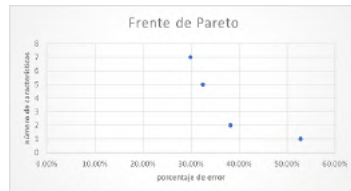


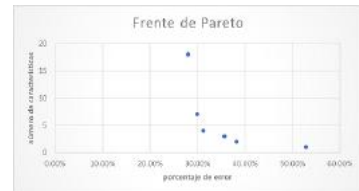
Figura A.6: Frentes de Pareto del Conjunto de datos audiology con FSGDE3. (2 - 2)



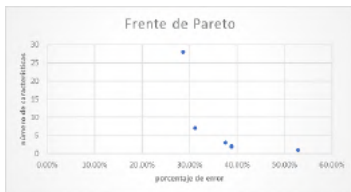
(a) Ejecución 1



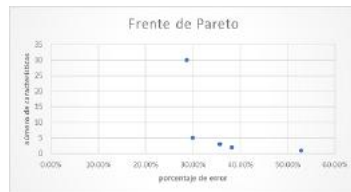
(b) Ejecución 2



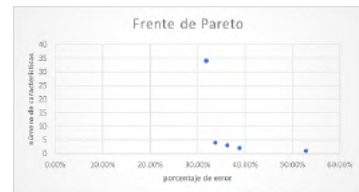
(c) Ejecución 3



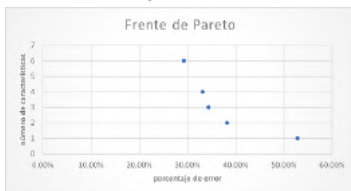
(d) Ejecución 4



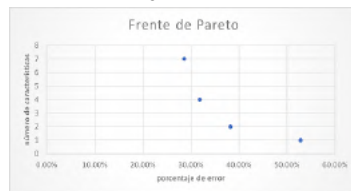
(e) Ejecución 5



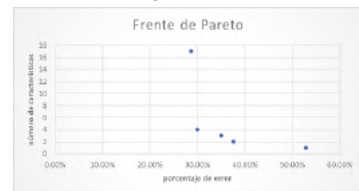
(f) Ejecución 6



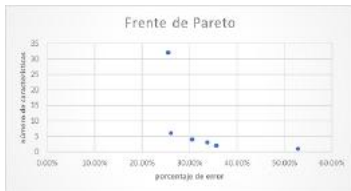
(g) Ejecución 7



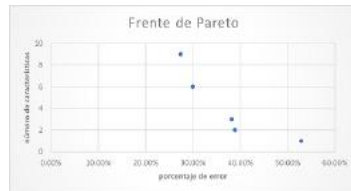
(h) Ejecución 8



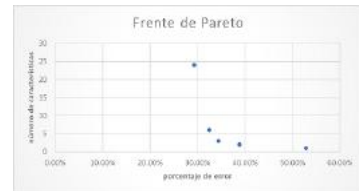
(i) Ejecución 9



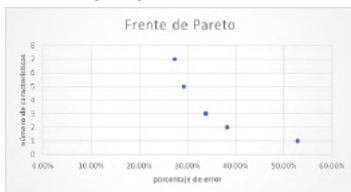
(j) Ejecución 10



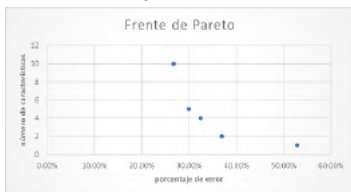
(k) Ejecución 11



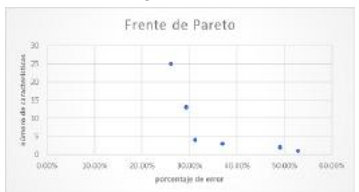
(l) Ejecución 12



(m) Ejecución 13

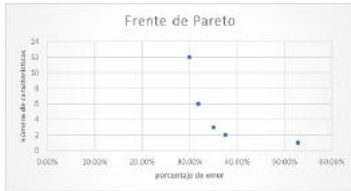


(n) Ejecución 14

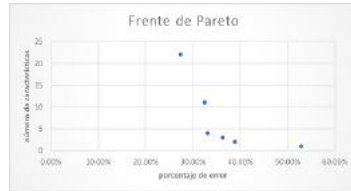


(ñ) Ejecución 15

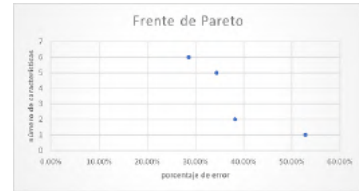
Figura A.7: Frentes de Pareto del Conjunto de datos audiology con FSNSGA II. (1 - 2)



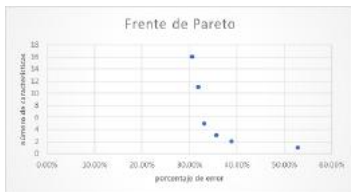
(a) Ejecución 16



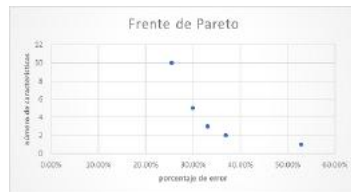
(b) Ejecución 17



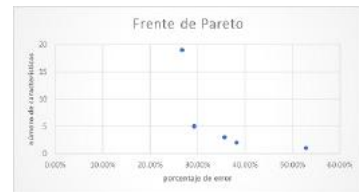
(c) Ejecución 18



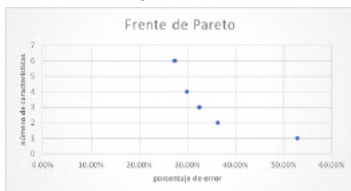
(d) Ejecución 19



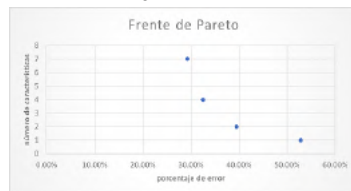
(e) Ejecución 20



(f) Ejecución 21



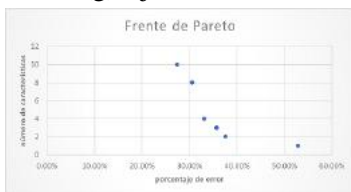
(g) Ejecución 22



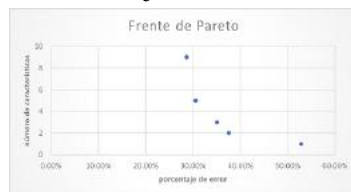
(h) Ejecución 23



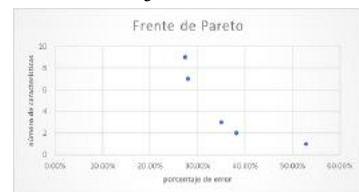
(i) Ejecución 24



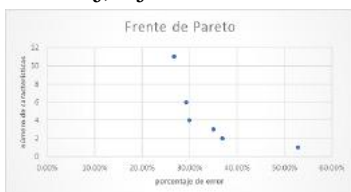
(j) Ejecución 25



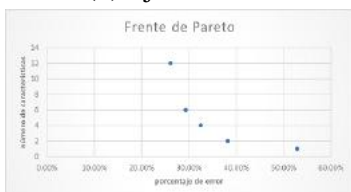
(k) Ejecución 26



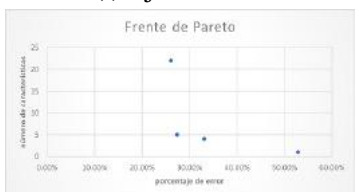
(l) Ejecución 27



(m) Ejecución 28



(n) Ejecución 29



(ñ) Ejecución 30

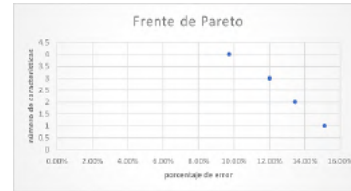
Figura A.8: Frontes de Pareto del Conjunto de datos audiology con FSNSGA II. (2 - 2)



(a) Ejecución 1



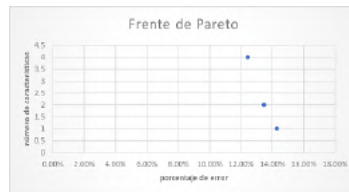
(b) Ejecución 2



(c) Ejecución 3



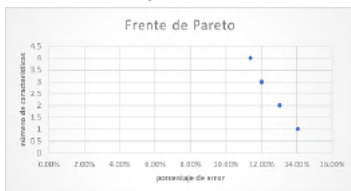
(d) Ejecución 4



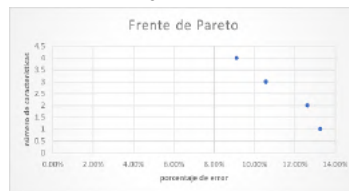
(e) Ejecución 5



(f) Ejecución 6



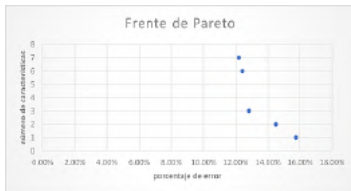
(g) Ejecución 7



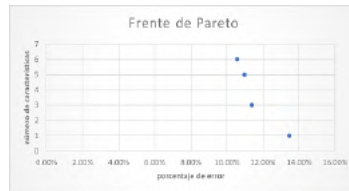
(h) Ejecución 8



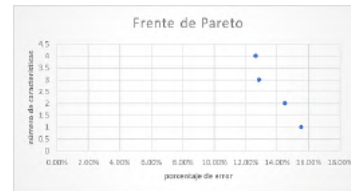
(i) Ejecución 9



(j) Ejecución 10



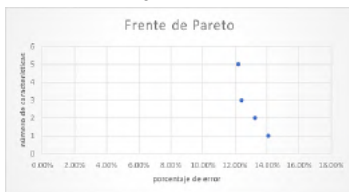
(k) Ejecución 11



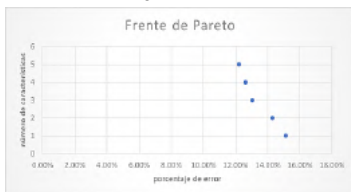
(l) Ejecución 12



(m) Ejecución 13

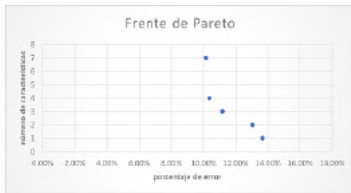


(n) Ejecución 14

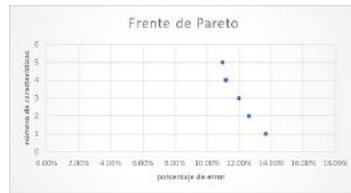


(ñ) Ejecución 15

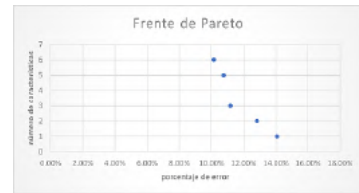
Figura A.9: Frentes de Pareto del Conjunto de datos australian con FSGDE3. (1 - 2)



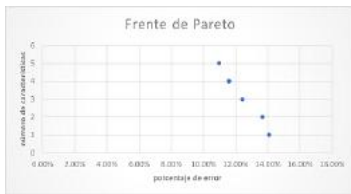
(a) Ejecución 16



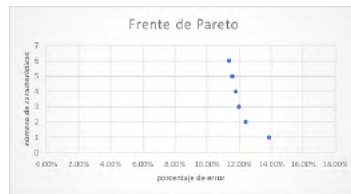
(b) Ejecución 17



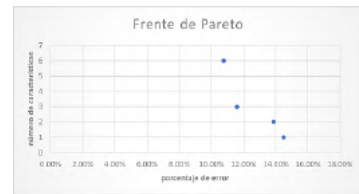
(c) Ejecución 18



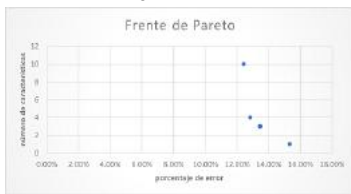
(d) Ejecución 19



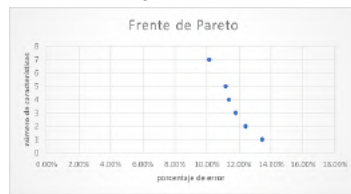
(e) Ejecución 20



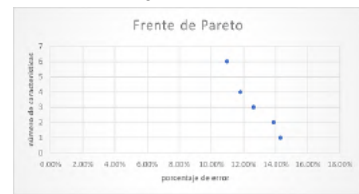
(f) Ejecución 21



(g) Ejecución 22



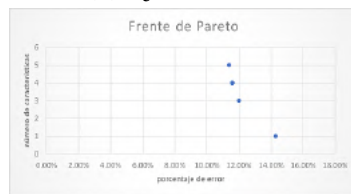
(h) Ejecución 23



(i) Ejecución 24



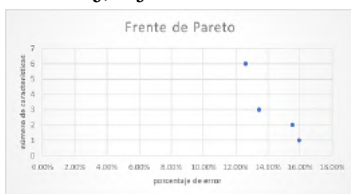
(j) Ejecución 25



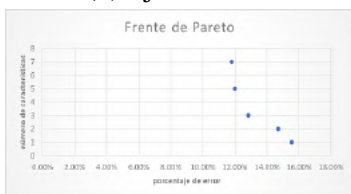
(k) Ejecución 26



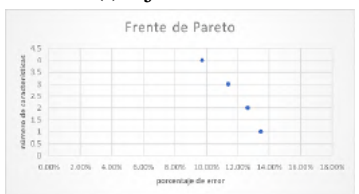
(l) Ejecución 27



(m) Ejecución 28

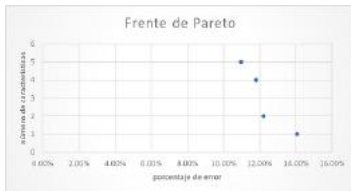


(n) Ejecución 29

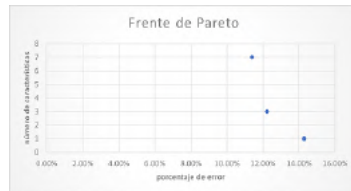


(ñ) Ejecución 30

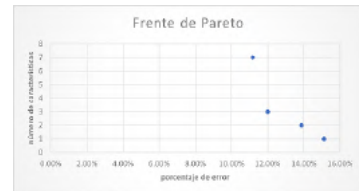
Figura A.10: Frentes de Pareto del Conjunto de datos australiano con FSGDE3. (2 - 2)



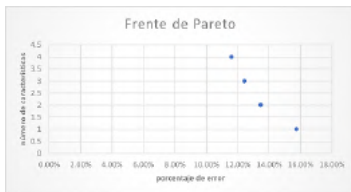
(a) Ejecución 1



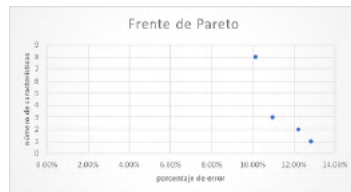
(b) Ejecución 2



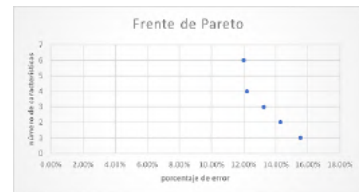
(c) Ejecución 3



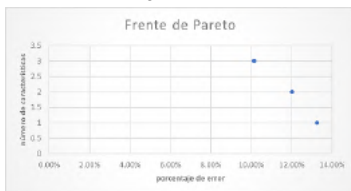
(d) Ejecución 4



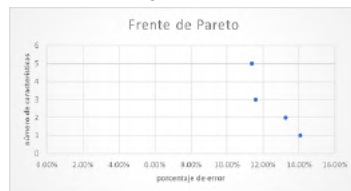
(e) Ejecución 5



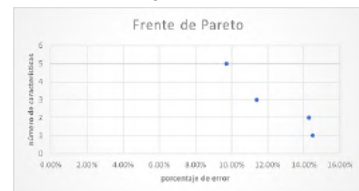
(f) Ejecución 6



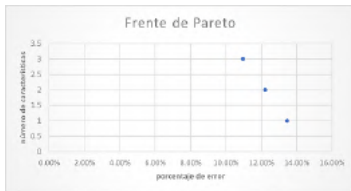
(g) Ejecución 7



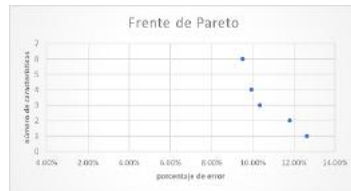
(h) Ejecución 8



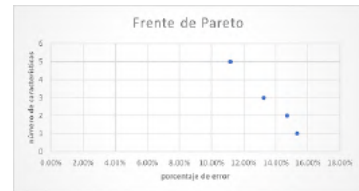
(i) Ejecución 9



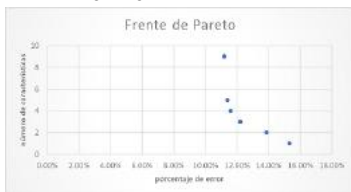
(j) Ejecución 10



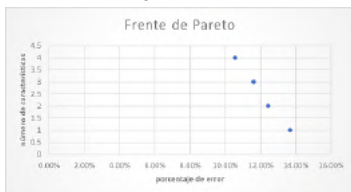
(k) Ejecución 11



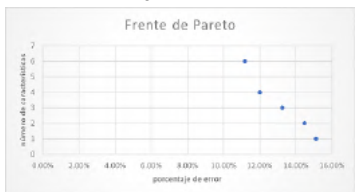
(l) Ejecución 12



(m) Ejecución 13

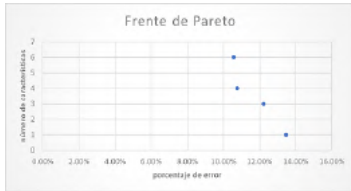


(n) Ejecución 14

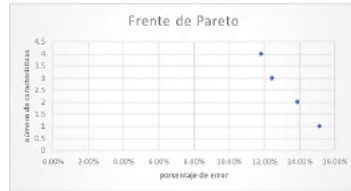


(ñ) Ejecución 15

Figura A.11: Frentes de Pareto del Conjunto de datos australian con FSNSGA II. (1 - 2)



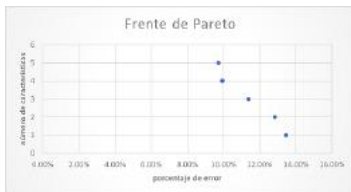
(a) Ejecución 16



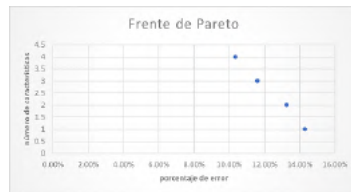
(b) Ejecución 17



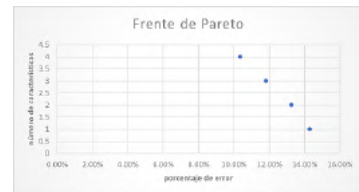
(c) Ejecución 18



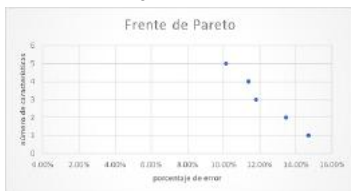
(d) Ejecución 19



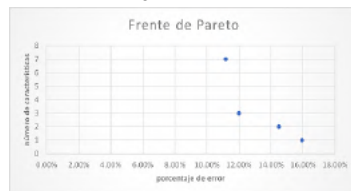
(e) Ejecución 20



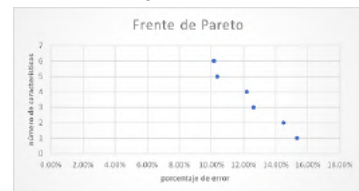
(f) Ejecución 21



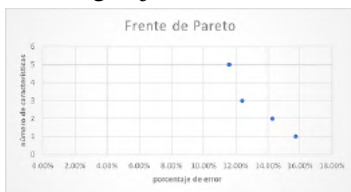
(g) Ejecución 22



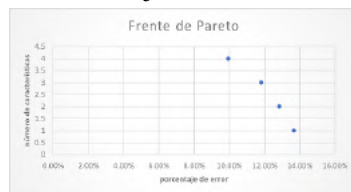
(h) Ejecución 23



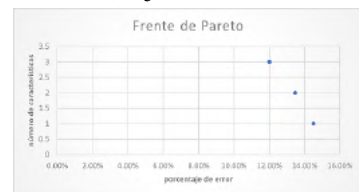
(i) Ejecución 24



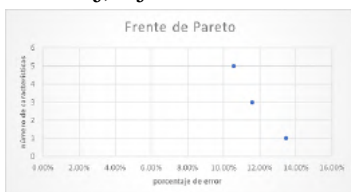
(j) Ejecución 25



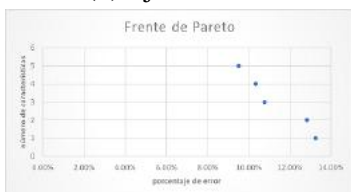
(k) Ejecución 26



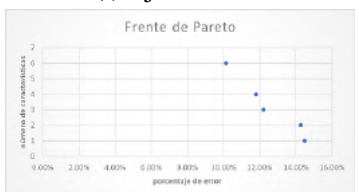
(l) Ejecución 27



(m) Ejecución 28

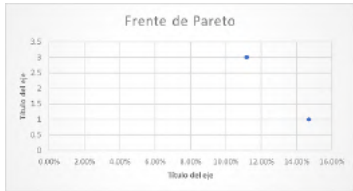


(n) Ejecución 29

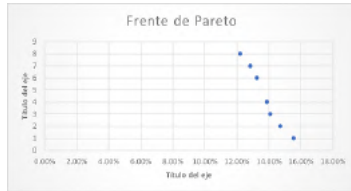


(ñ) Ejecución 30

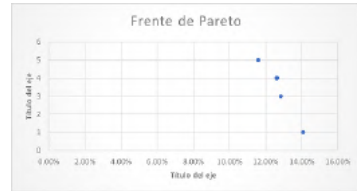
Figura A.12: Frentes de Pareto del Conjunto de datos australian con FSNSGA II. (2 - 2)



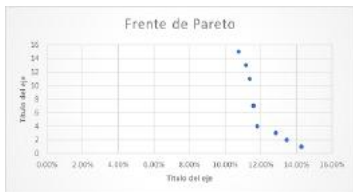
(a) Ejecución 1



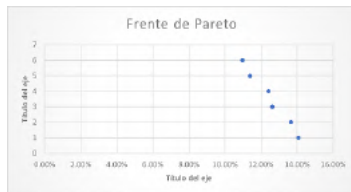
(b) Ejecución 2



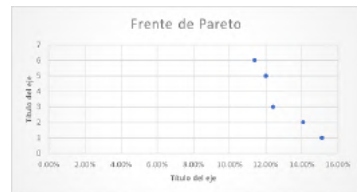
(c) Ejecución 3



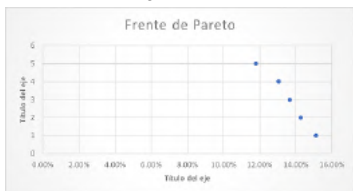
(d) Ejecución 4



(e) Ejecución 5



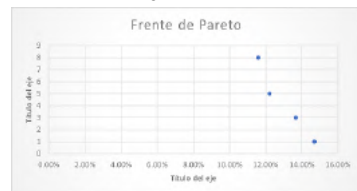
(f) Ejecución 6



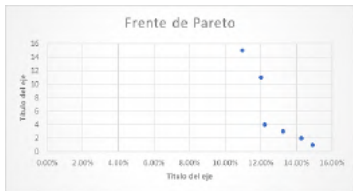
(g) Ejecución 7



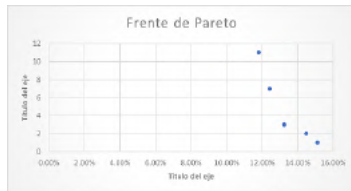
(h) Ejecución 8



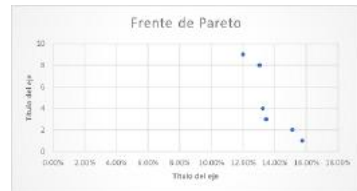
(i) Ejecución 9



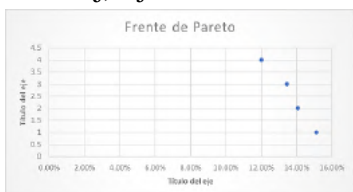
(j) Ejecución 10



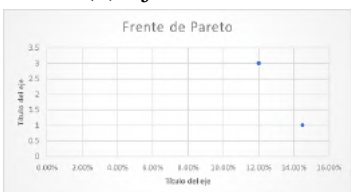
(k) Ejecución 11



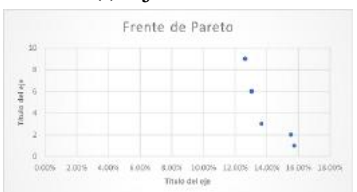
(l) Ejecución 12



(m) Ejecución 13

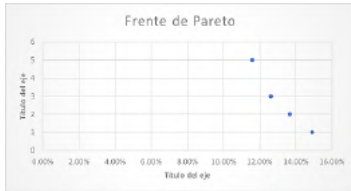


(n) Ejecución 14

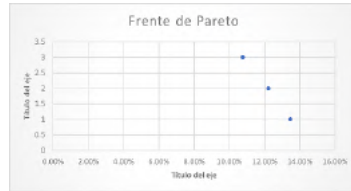


(ñ) Ejecución 15

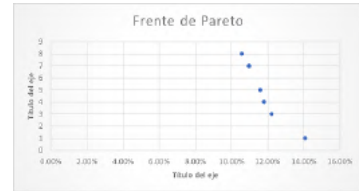
Figura A.13: Frentes de Pareto del Conjunto de datos crx con FSGDE3. (1 - 2)



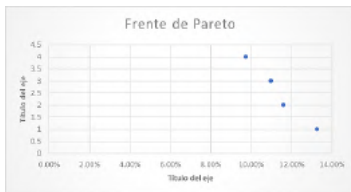
(a) Ejecución 16



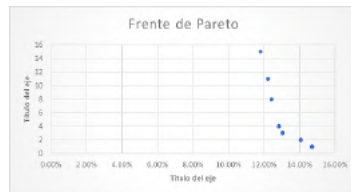
(b) Ejecución 17



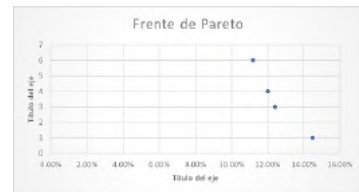
(c) Ejecución 18



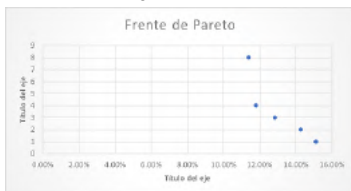
(d) Ejecución 19



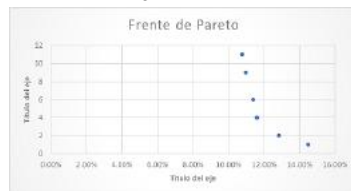
(e) Ejecución 20



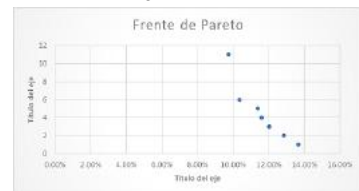
(f) Ejecución 21



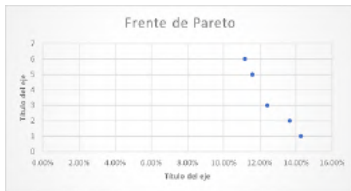
(g) Ejecución 22



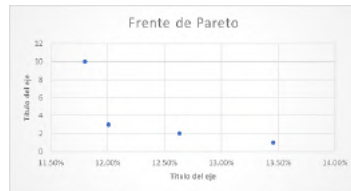
(h) Ejecución 23



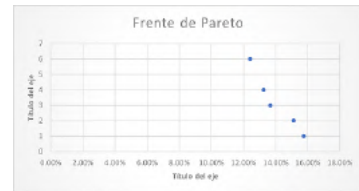
(i) Ejecución 24



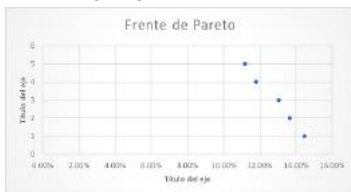
(j) Ejecución 25



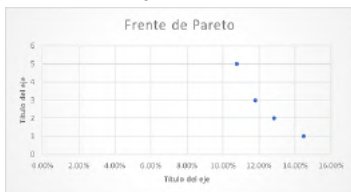
(k) Ejecución 26



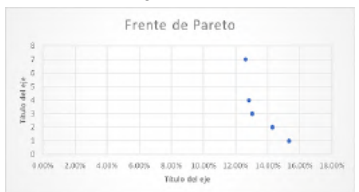
(l) Ejecución 27



(m) Ejecución 28

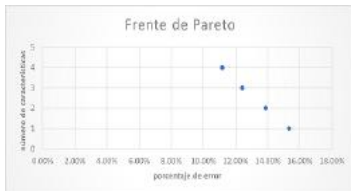


(n) Ejecución 29

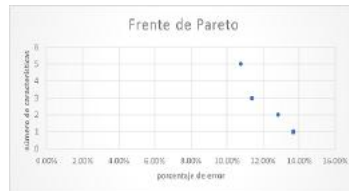


(ñ) Ejecución 30

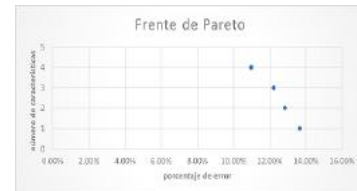
Figura A.14: Frentes de Pareto del Conjunto de datos crx con FSGDE3. (2 - 2)



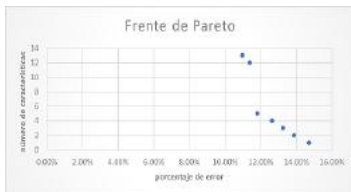
(a) Ejecución 1



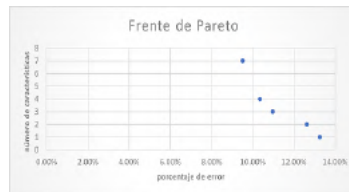
(b) Ejecución 2



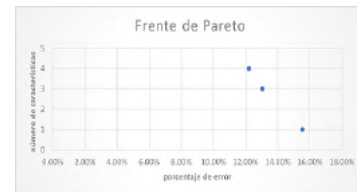
(c) Ejecución 3



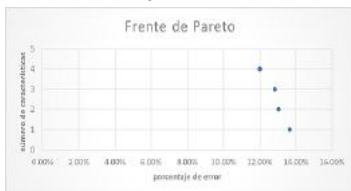
(d) Ejecución 4



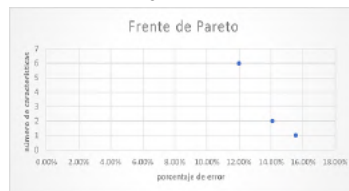
(e) Ejecución 5



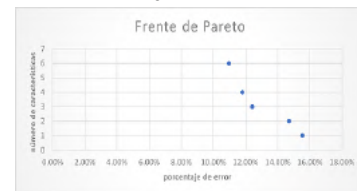
(f) Ejecución 6



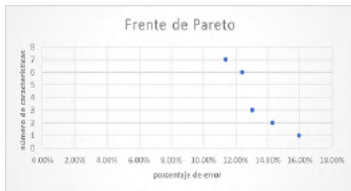
(g) Ejecución 7



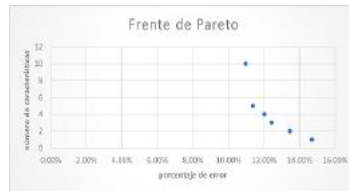
(h) Ejecución 8



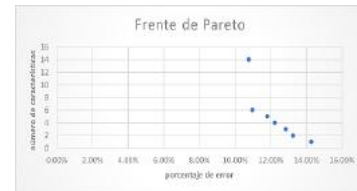
(i) Ejecución 9



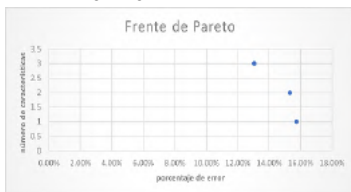
(j) Ejecución 10



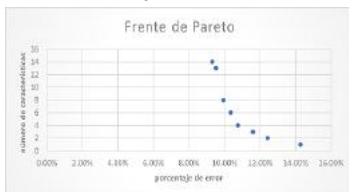
(k) Ejecución 11



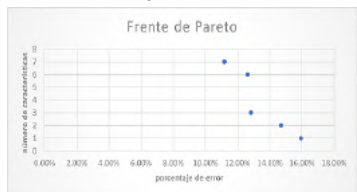
(l) Ejecución 12



(m) Ejecución 13



(n) Ejecución 14



(ñ) Ejecución 15

Figura A.15: Frentes de Pareto del Conjunto de datos crx con FSNSGA II. (1 - 2)



Figura A.16: Frentes de Pareto del Conjunto de datos crx con FSNSGA II. (2 - 2)

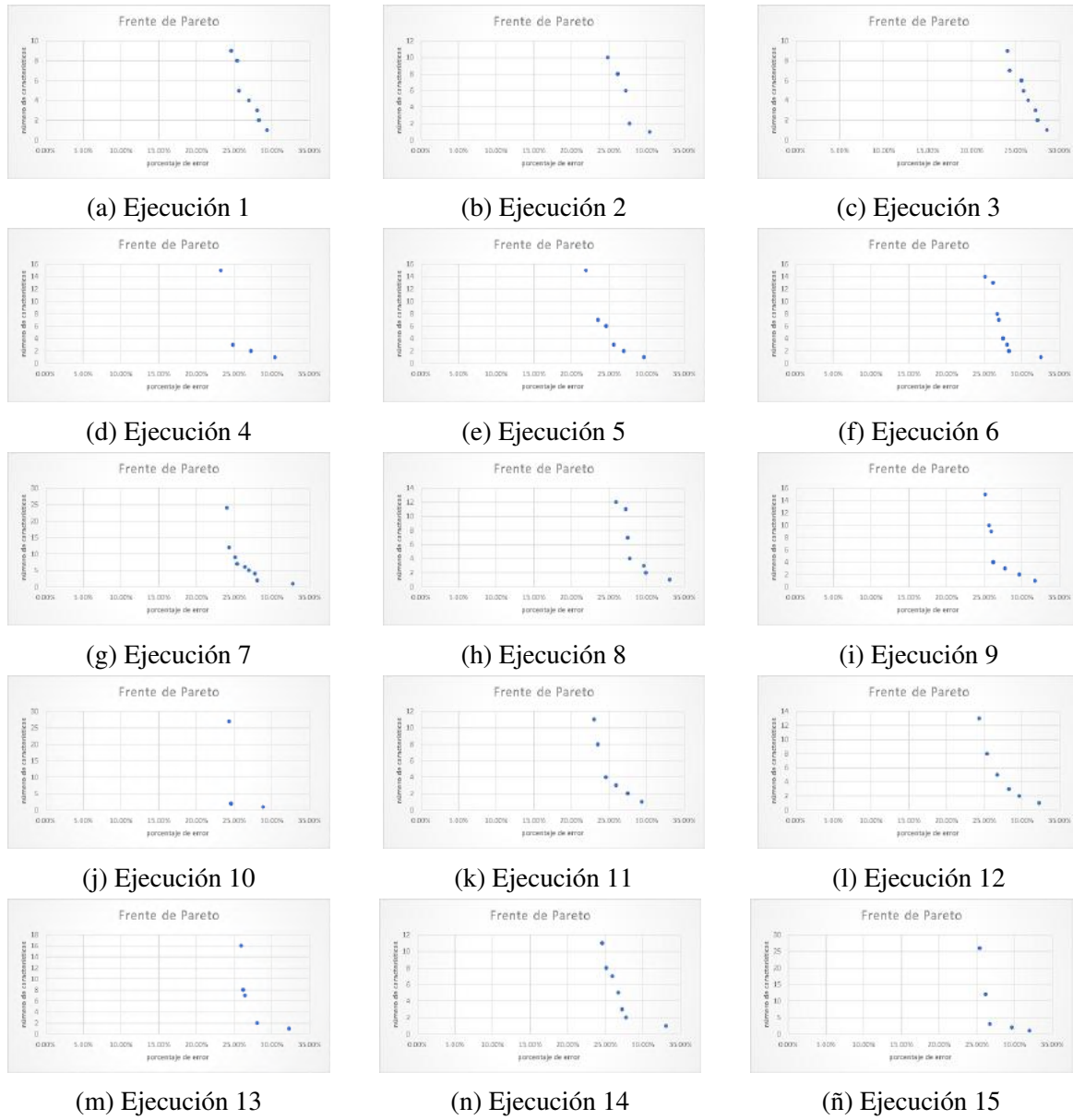
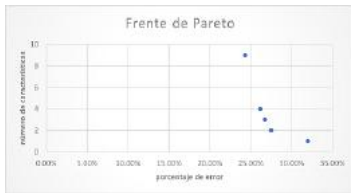


Figura A.17: Frentes de Pareto del Conjunto de datos cylinder-bands con FSGDE3. (1 - 2)



(a) Ejecución 16



(b) Ejecución 17



(c) Ejecución 18



(d) Ejecución 19



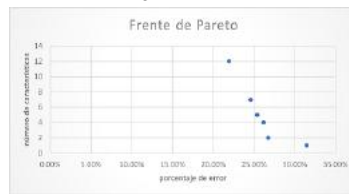
(e) Ejecución 20



(f) Ejecución 21



(g) Ejecución 22



(h) Ejecución 23



(i) Ejecución 24



(j) Ejecución 25



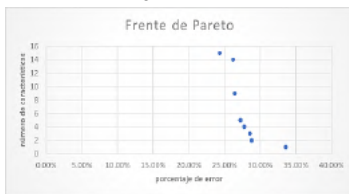
(k) Ejecución 26



(l) Ejecución 27



(m) Ejecución 28



(n) Ejecución 29



(ñ) Ejecución 30

Figura A.18: Frentes de Pareto del Conjunto de datos cylinder-bands con FSGDE3. (2 - 2)

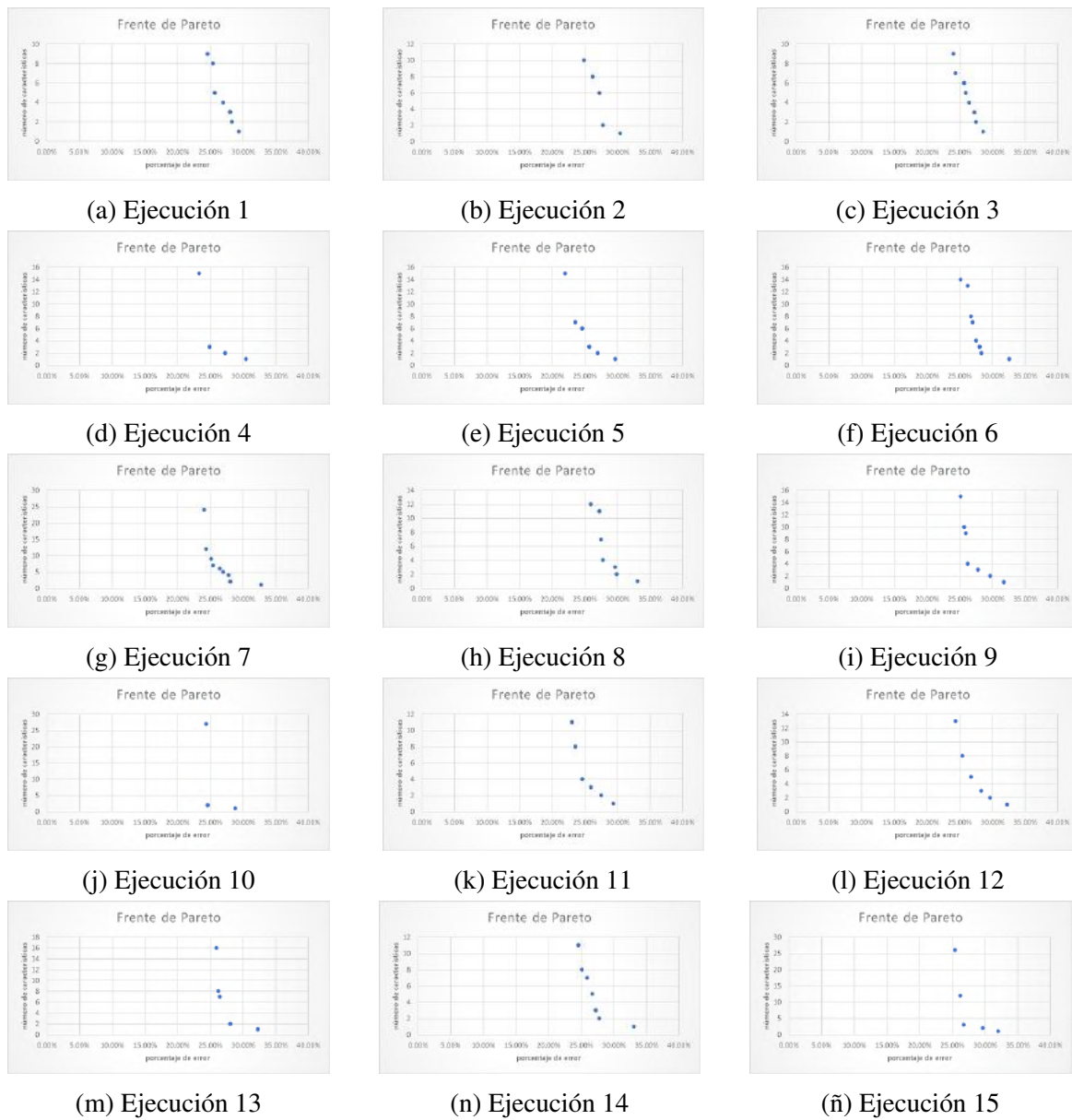


Figura A.19: Frentes de Pareto del Conjunto de datos cylinder-bands con FSNSGA II. (1 - 2)

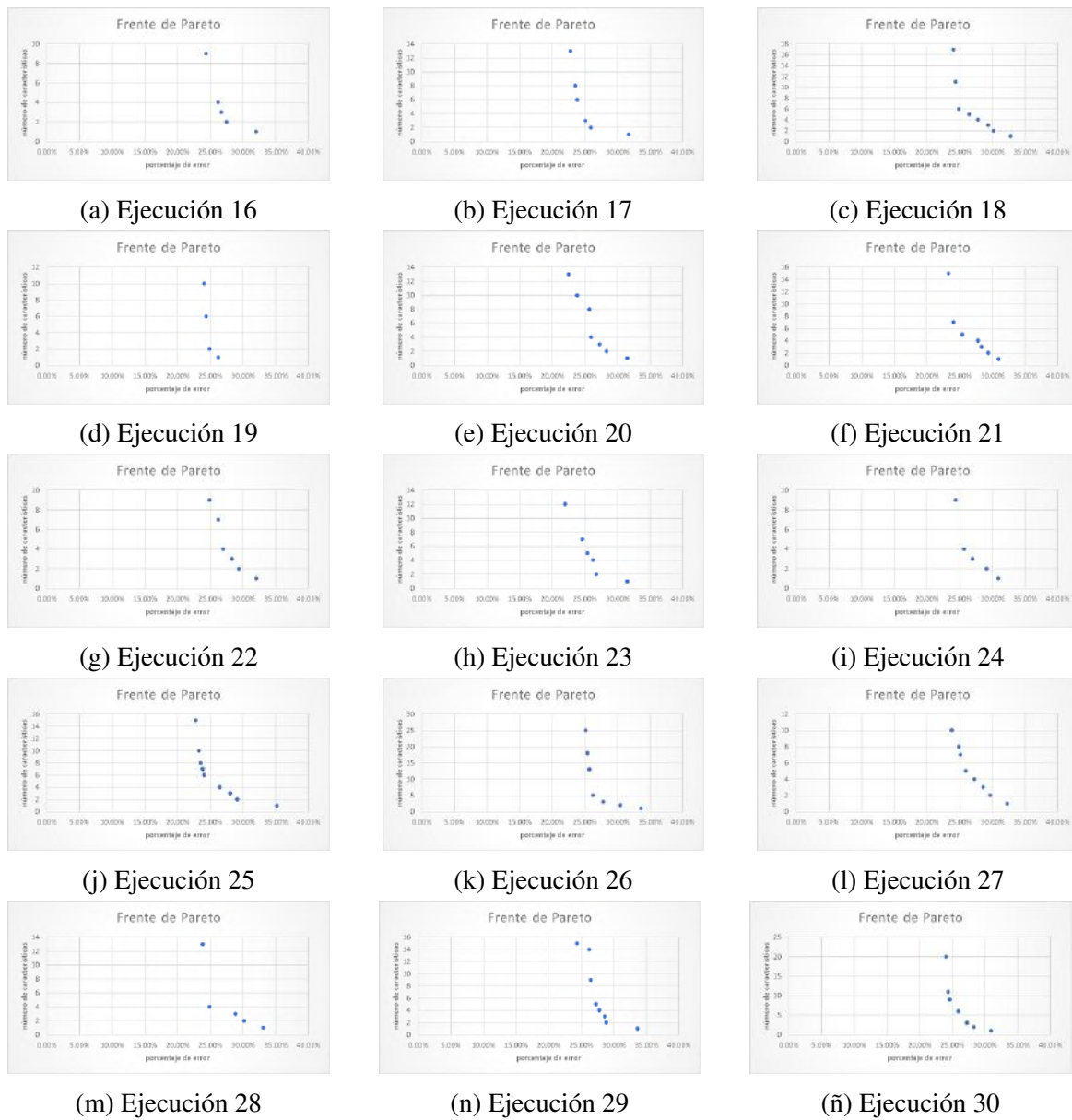


Figura A.20: Frentes de Pareto del Conjunto de datos cylinder-bands con FSNSGA II. (2 - 2)

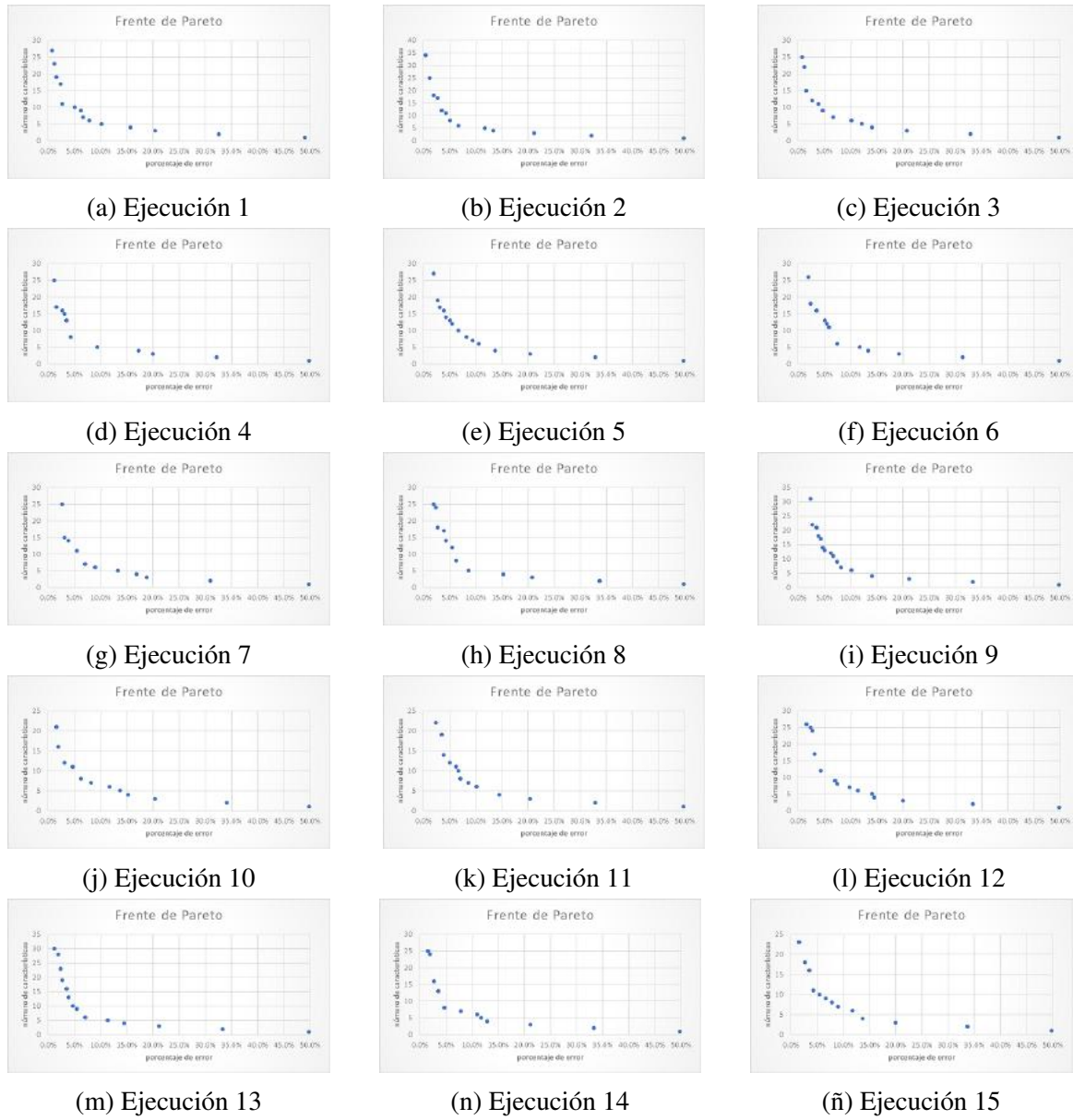


Figura A.21: Frentes de Pareto del Conjunto de datos dermatology con FSGDE3. (1 - 2)

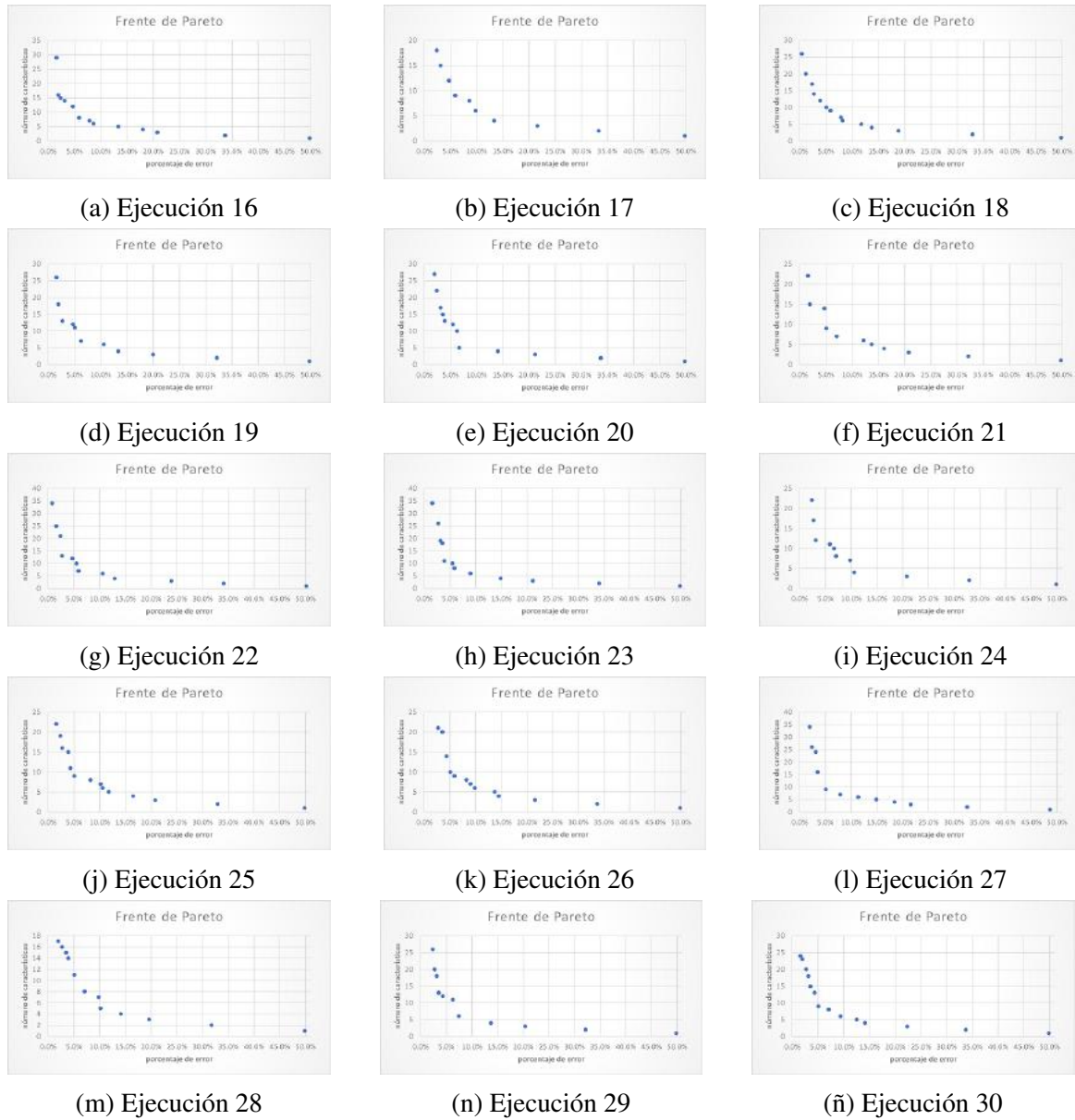


Figura A.22: Frentes de Pareto del Conjunto de datos dermatology con FSGDE3. (2 - 2)

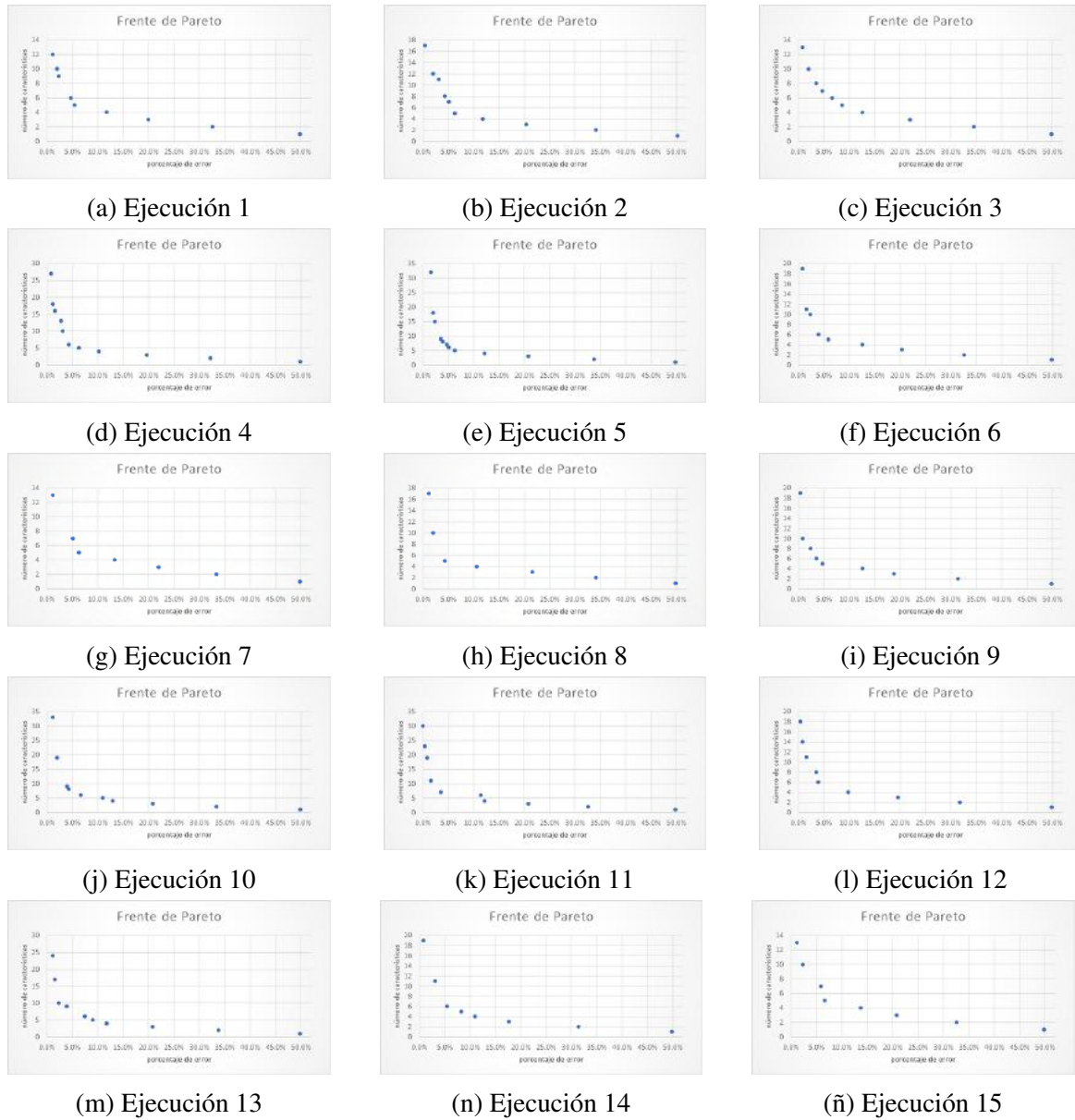
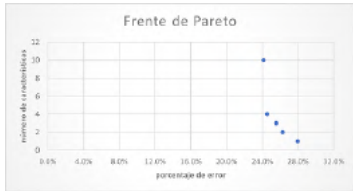


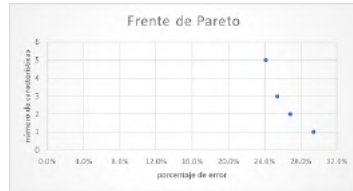
Figura A.23: Frentes de Pareto del Conjunto de datos dermatology con FSNSGA II. (1 - 2)



Figura A.24: Frentes de Pareto del Conjunto de datos dermatology con FSNSGA II. (2 - 2)



(a) Ejecución 1



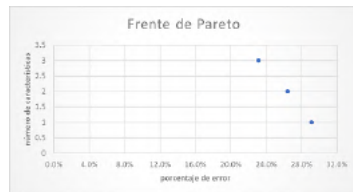
(b) Ejecución 2



(c) Ejecución 3



(d) Ejecución 4



(e) Ejecución 5



(f) Ejecución 6



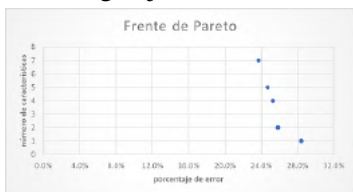
(g) Ejecución 7



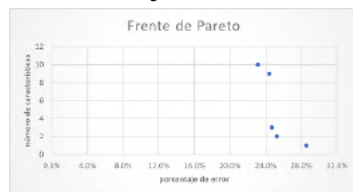
(h) Ejecución 8



(i) Ejecución 9



(j) Ejecución 10



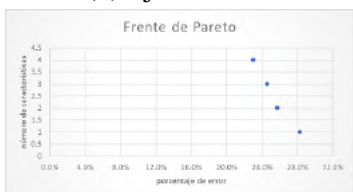
(k) Ejecución 11



(l) Ejecución 12



(m) Ejecución 13

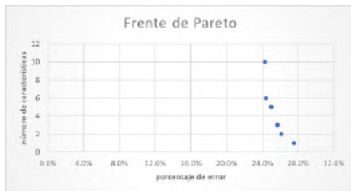


(n) Ejecución 14

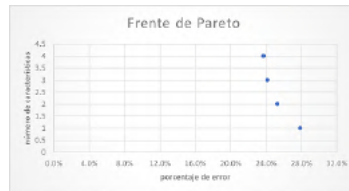


(ñ) Ejecución 15

Figura A.25: Frentes de Pareto del Conjunto de datos german-c con FSGDE3. (1 - 2)



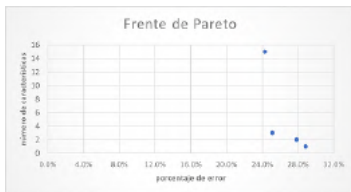
(a) Ejecución 16



(b) Ejecución 17



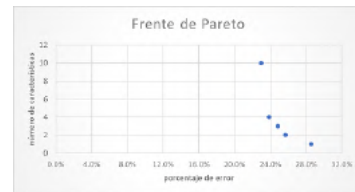
(c) Ejecución 18



(d) Ejecución 19



(e) Ejecución 20



(f) Ejecución 21



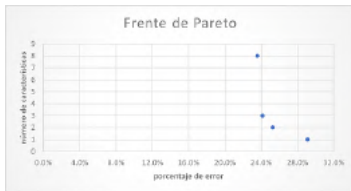
(g) Ejecución 22



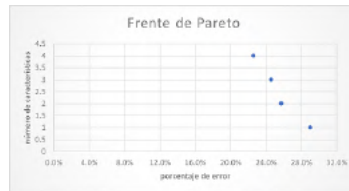
(h) Ejecución 23



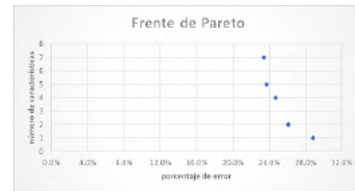
(i) Ejecución 24



(j) Ejecución 25



(k) Ejecución 26



(l) Ejecución 27



(m) Ejecución 28

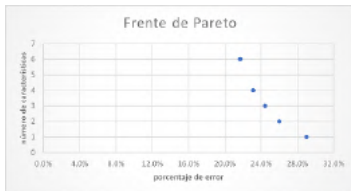


(n) Ejecución 29

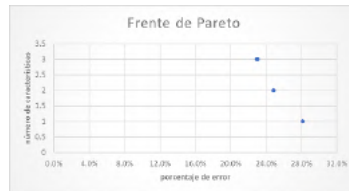


(ñ) Ejecución 30

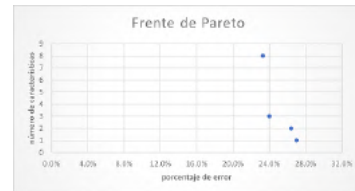
Figura A.26: Frentes de Pareto del Conjunto de datos german-c con FSGDE3. (2 - 2)



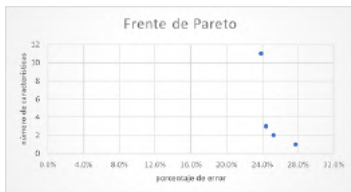
(a) Ejecución 1



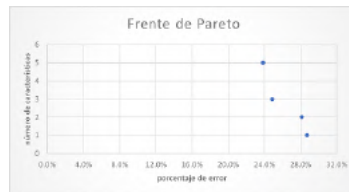
(b) Ejecución 2



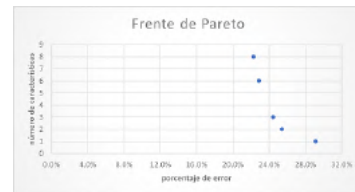
(c) Ejecución 3



(d) Ejecución 4



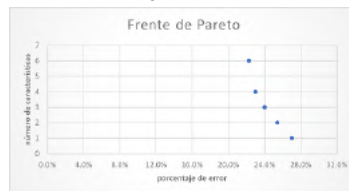
(e) Ejecución 5



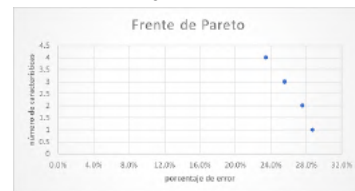
(f) Ejecución 6



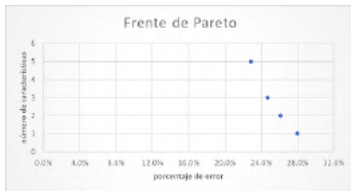
(g) Ejecución 7



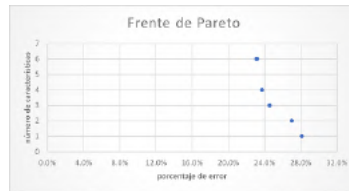
(h) Ejecución 8



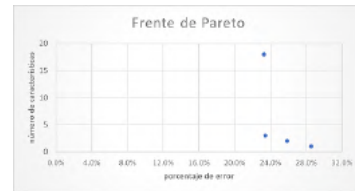
(i) Ejecución 9



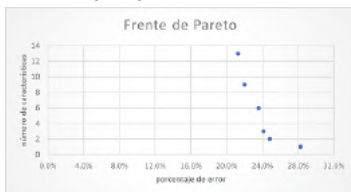
(j) Ejecución 10



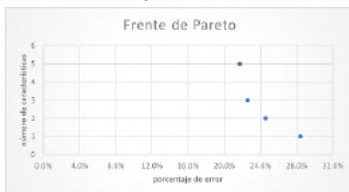
(k) Ejecución 11



(l) Ejecución 12



(m) Ejecución 13



(n) Ejecución 14



(ñ) Ejecución 15

Figura A.27: Frentes de Pareto del Conjunto de datos german-c con FSNSGA II. (1 - 2)



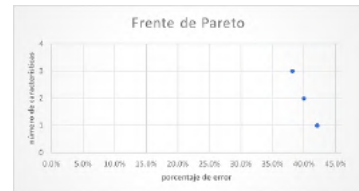
Figura A.28: Frentes de Pareto del Conjunto de datos german-c con FSNSGA II. (2 - 2)



(a) Ejecución 1



(b) Ejecución 2



(c) Ejecución 3



(d) Ejecución 4



(e) Ejecución 5



(f) Ejecución 6



(g) Ejecución 7



(h) Ejecución 8



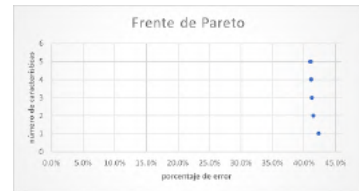
(i) Ejecución 9



(j) Ejecución 10



(k) Ejecución 11



(l) Ejecución 12



(m) Ejecución 13



(n) Ejecución 14



(ñ) Ejecución 15

Figura A.29: Frentes de Pareto del Conjunto de datos h-valley con FSGDE3. (1 - 2)



(a) Ejecución 16



(b) Ejecución 17



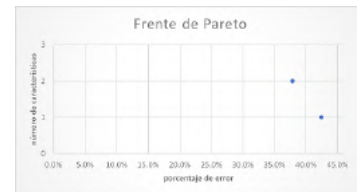
(c) Ejecución 18



(d) Ejecución 19



(e) Ejecución 20



(f) Ejecución 21



(g) Ejecución 22



(h) Ejecución 23



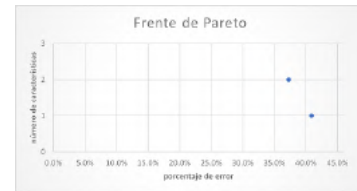
(i) Ejecución 24



(j) Ejecución 25



(k) Ejecución 26



(l) Ejecución 27



(m) Ejecución 28



(n) Ejecución 29



(ñ) Ejecución 30

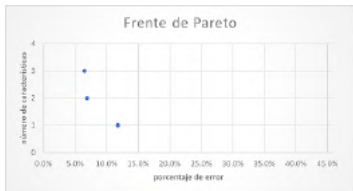
Figura A.30: Frentes de Pareto del Conjunto de datos h-valley con FSGDE3. (2 - 2)



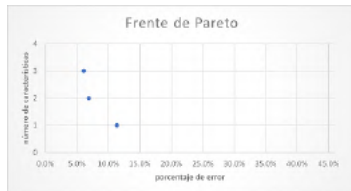
Figura A.31: Frentes de Pareto del Conjunto de datos h-valley con FSNSGA II. (1 - 2)



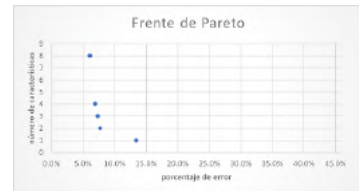
Figura A.32: Frentes de Pareto del Conjunto de datos h-valley con FSNSGA II. (2 - 2)



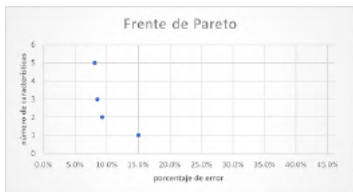
(a) Ejecución 1



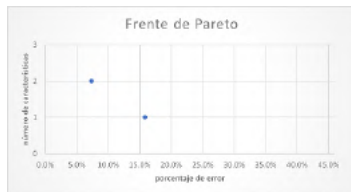
(b) Ejecución 2



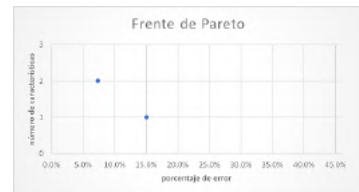
(c) Ejecución 3



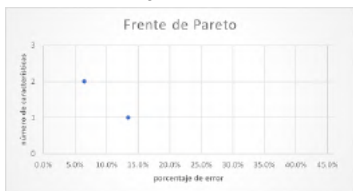
(d) Ejecución 4



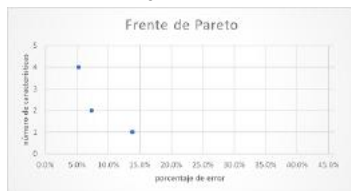
(e) Ejecución 5



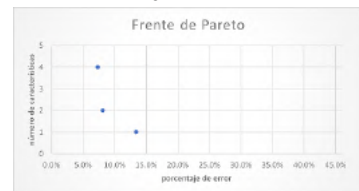
(f) Ejecución 6



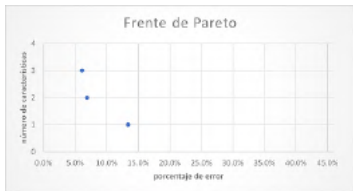
(g) Ejecución 7



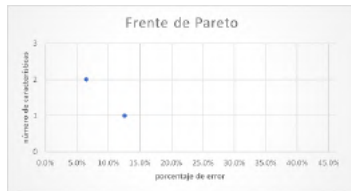
(h) Ejecución 8



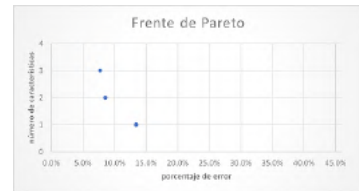
(i) Ejecución 9



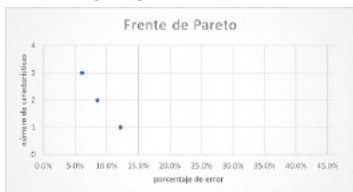
(j) Ejecución 10



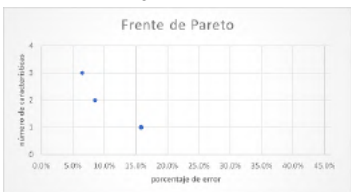
(k) Ejecución 11



(l) Ejecución 12



(m) Ejecución 13

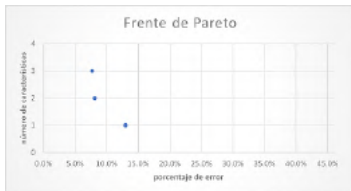


(n) Ejecución 14

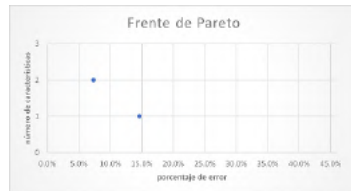


(ñ) Ejecución 15

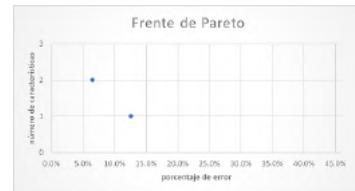
Figura A.33: Frentes de Pareto del Conjunto de datos ionosphere con FSGDE3. (1 - 2)



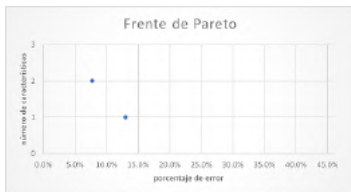
(a) Ejecución 16



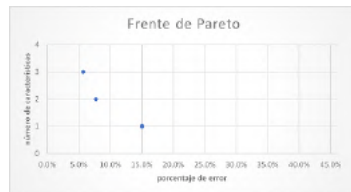
(b) Ejecución 17



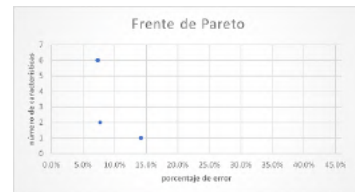
(c) Ejecución 18



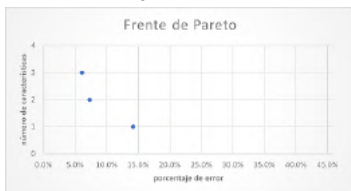
(d) Ejecución 19



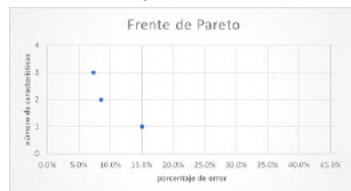
(e) Ejecución 20



(f) Ejecución 21



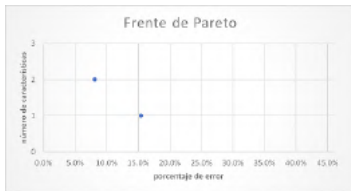
(g) Ejecución 22



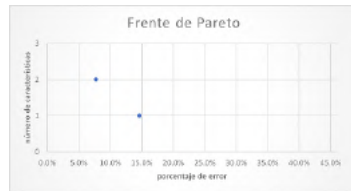
(h) Ejecución 23



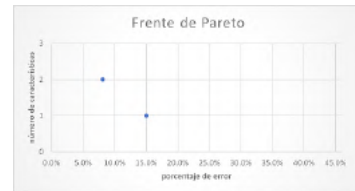
(i) Ejecución 24



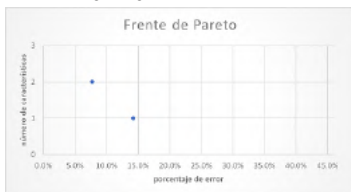
(j) Ejecución 25



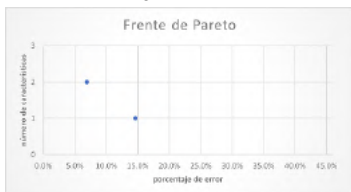
(k) Ejecución 26



(l) Ejecución 27



(m) Ejecución 28

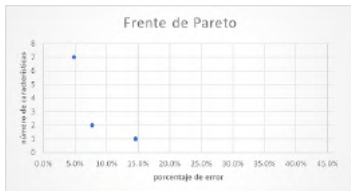


(n) Ejecución 29

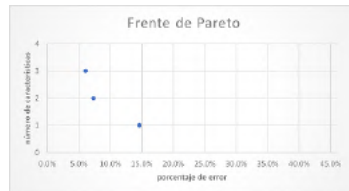


(ñ) Ejecución 30

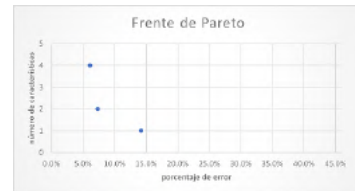
Figura A.34: Frentes de Pareto del Conjunto de datos ionosphere con FSGDE3. (2 - 2)



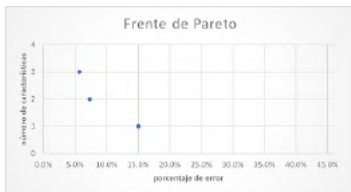
(a) Ejecución 1



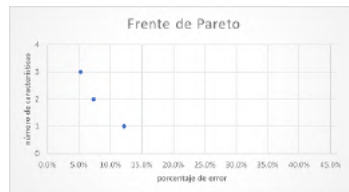
(b) Ejecución 2



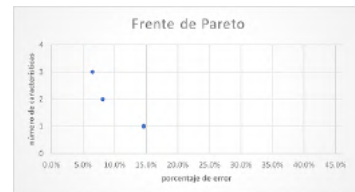
(c) Ejecución 3



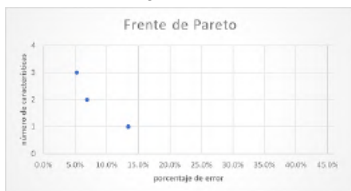
(d) Ejecución 4



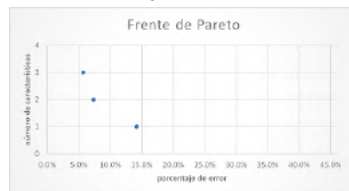
(e) Ejecución 5



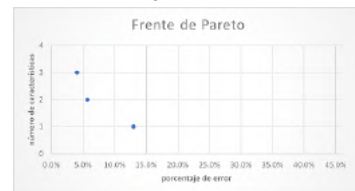
(f) Ejecución 6



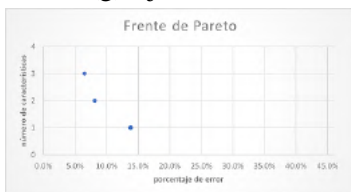
(g) Ejecución 7



(h) Ejecución 8



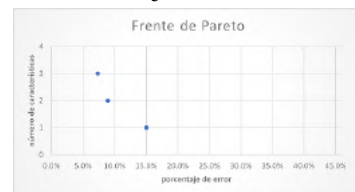
(i) Ejecución 9



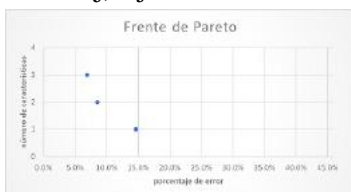
(j) Ejecución 10



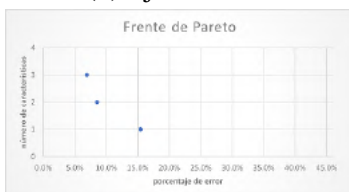
(k) Ejecución 11



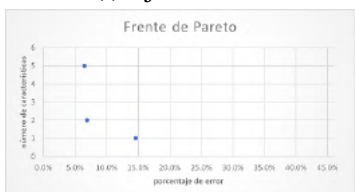
(l) Ejecución 12



(m) Ejecución 13



(n) Ejecución 14



(ñ) Ejecución 15

Figura A.35: Frentes de Pareto del Conjunto de datos ionosphere con FSNSGA II. (1 - 2)



Figura A.36: Frentes de Pareto del Conjunto de datos ionosphere con FSNSGA II. (2 - 2)

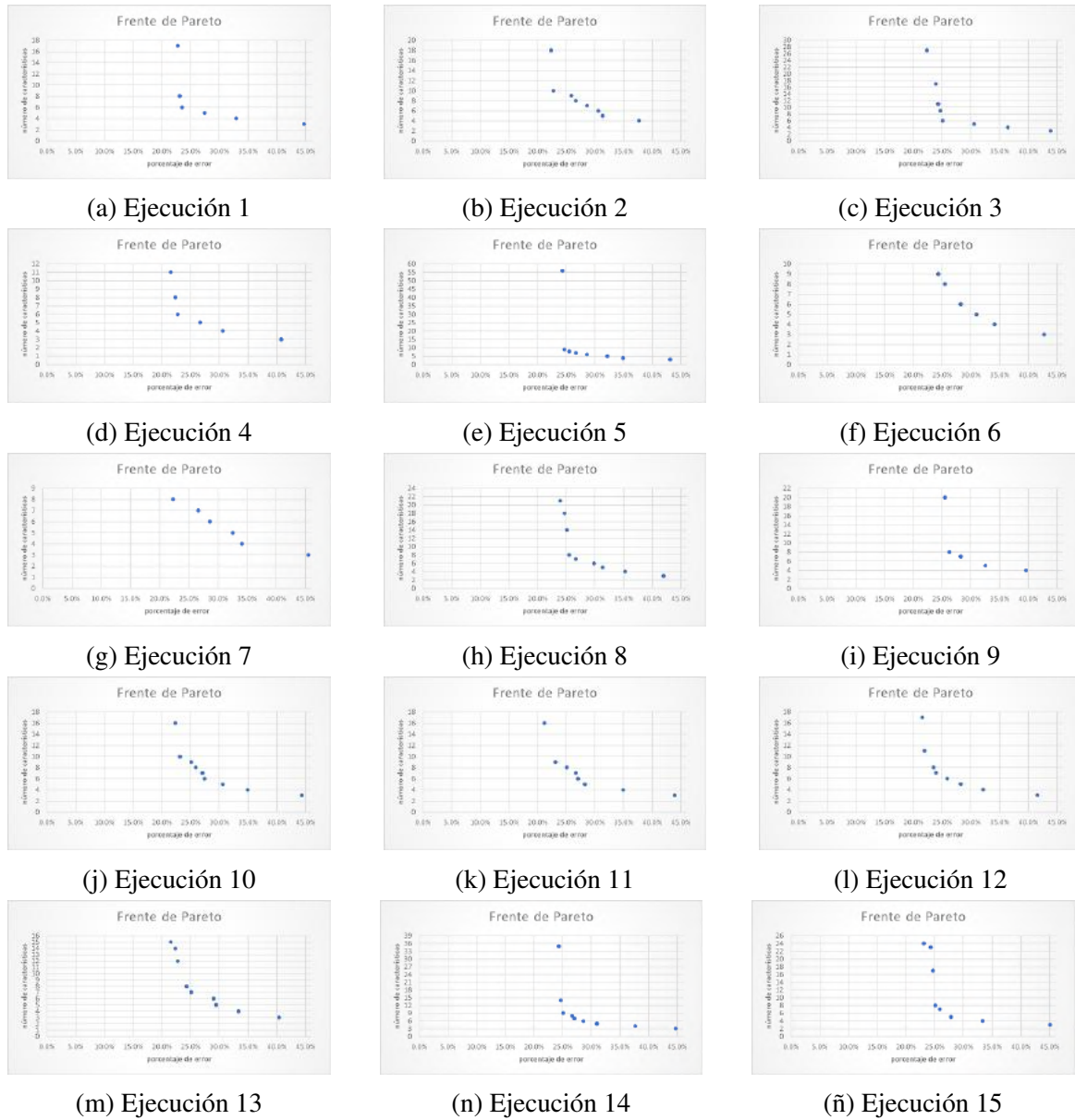


Figura A.37: Frentes de Pareto del Conjunto de datos m-libras con FSGDE3. (1 - 2)

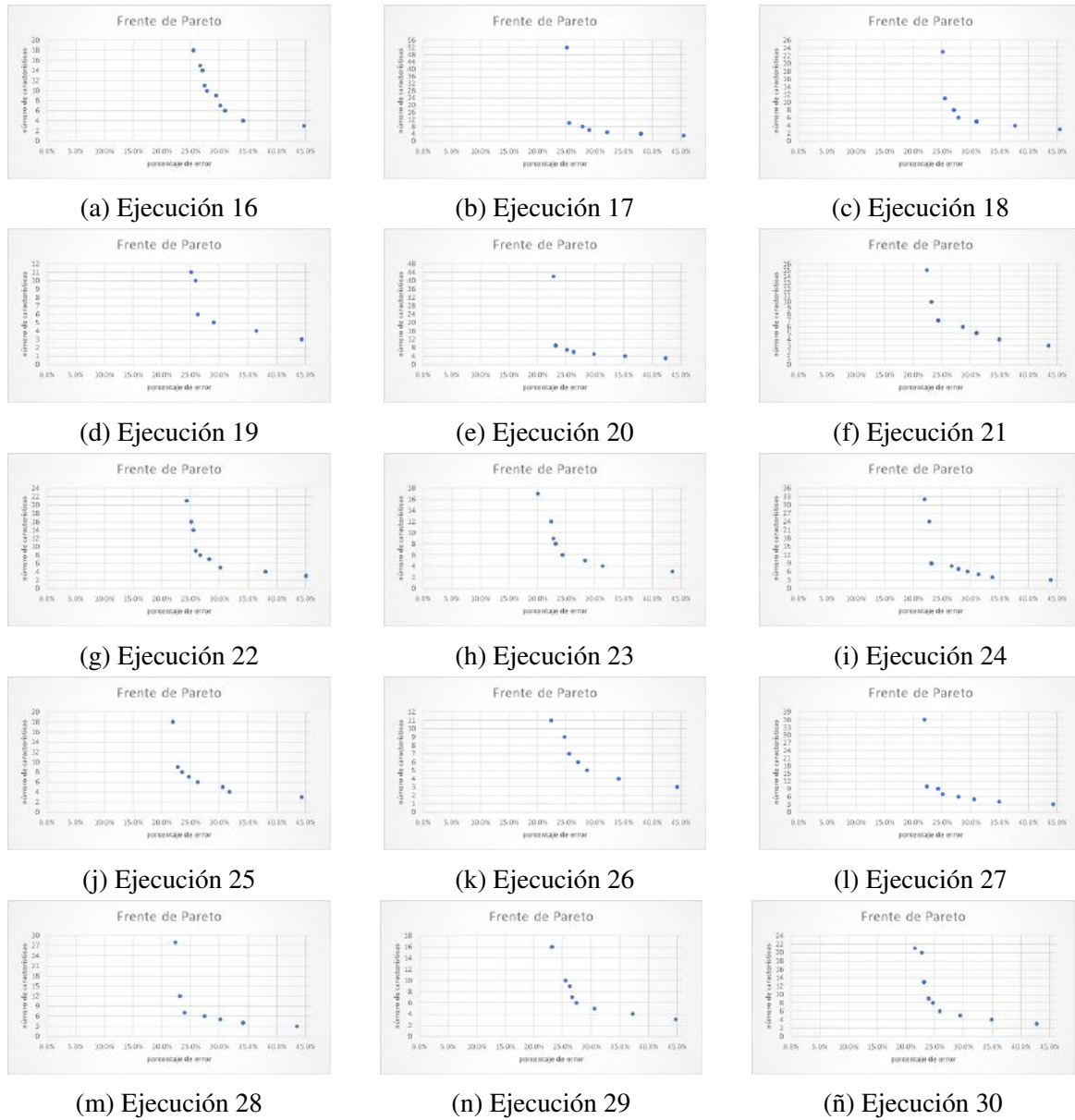
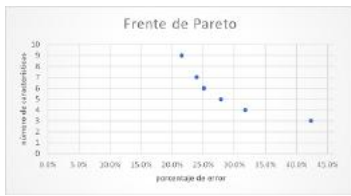


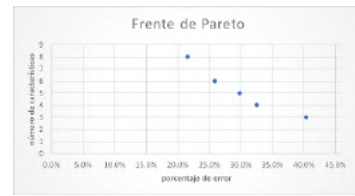
Figura A.38: Frentes de Pareto del Conjunto de datos m-libras con FSGDE3. (2 - 2)



(a) Ejecución 1



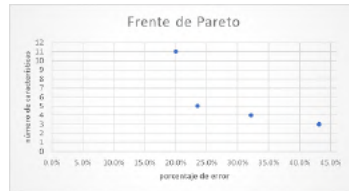
(b) Ejecución 2



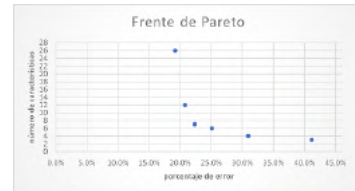
(c) Ejecución 3



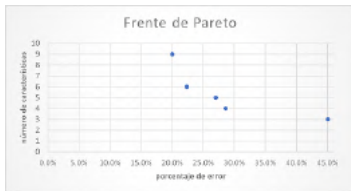
(d) Ejecución 4



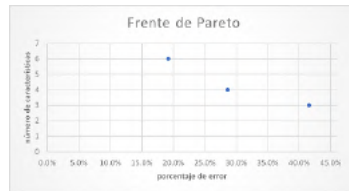
(e) Ejecución 5



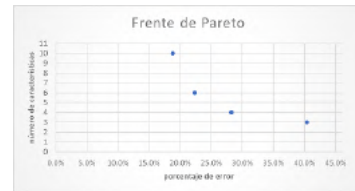
(f) Ejecución 6



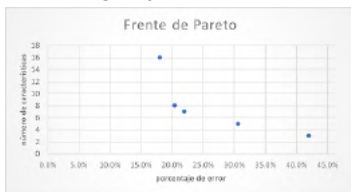
(g) Ejecución 7



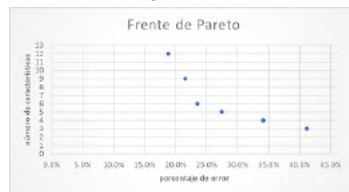
(h) Ejecución 8



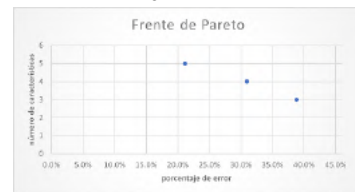
(i) Ejecución 9



(j) Ejecución 10



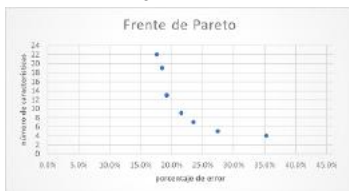
(k) Ejecución 11



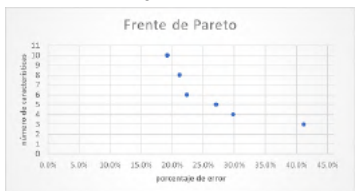
(l) Ejecución 12



(m) Ejecución 13



(n) Ejecución 14



(ñ) Ejecución 15

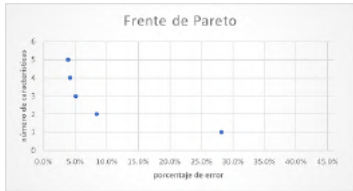
Figura A.39: Frentes de Pareto del Conjunto de datos m-libras con FSNSGA II. (1 - 2)



Figura A.40: Frentes de Pareto del Conjunto de datos m-libras con FSNSGA II. (2 - 2)

Capítulo B

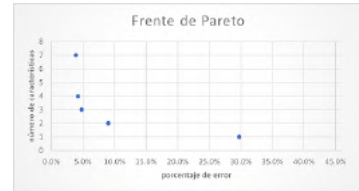
Anexos-Continuación



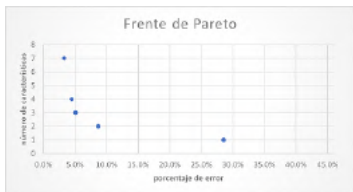
(a) Ejecución 1



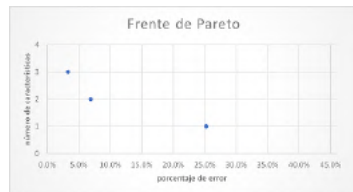
(b) Ejecución 2



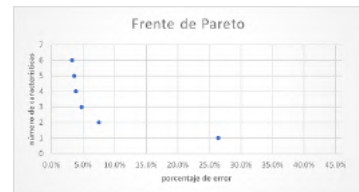
(c) Ejecución 3



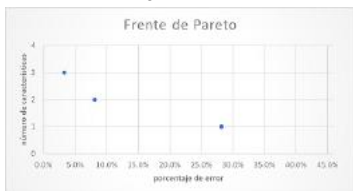
(d) Ejecución 4



(e) Ejecución 5



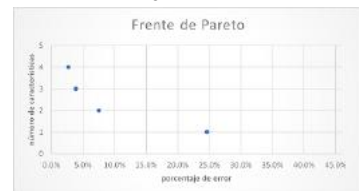
(f) Ejecución 6



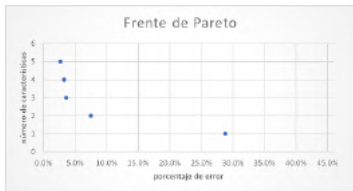
(g) Ejecución 7



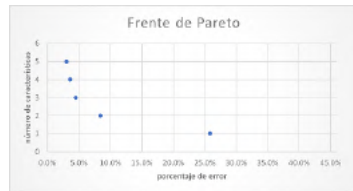
(h) Ejecución 8



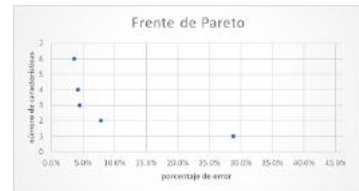
(i) Ejecución 9



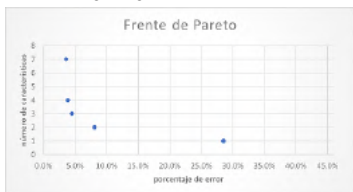
(j) Ejecución 10



(k) Ejecución 11



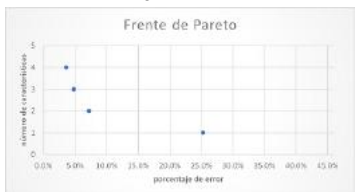
(l) Ejecución 12



(m) Ejecución 13

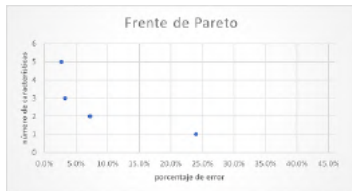


(n) Ejecución 14

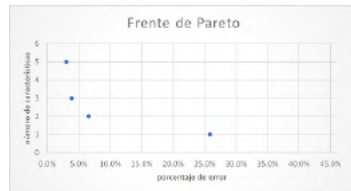


(ñ) Ejecución 15

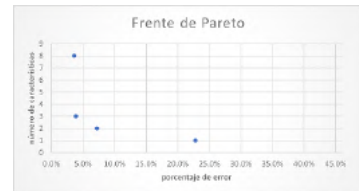
Figura B.1: Frentes de Pareto del Conjunto de datos musk-1 con FSGDE3. (1 - 2)



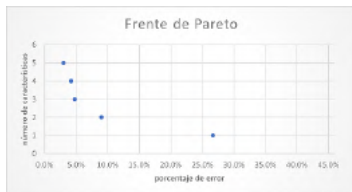
(a) Ejecución 16



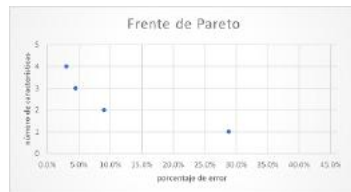
(b) Ejecución 17



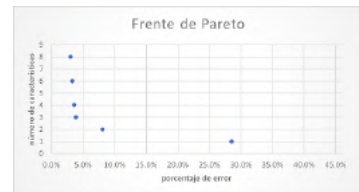
(c) Ejecución 18



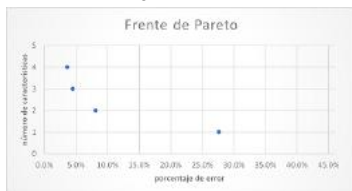
(d) Ejecución 19



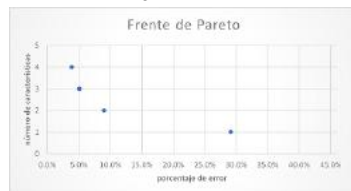
(e) Ejecución 20



(f) Ejecución 21



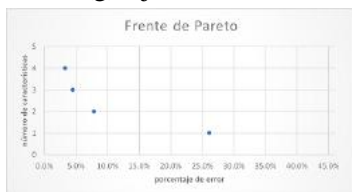
(g) Ejecución 22



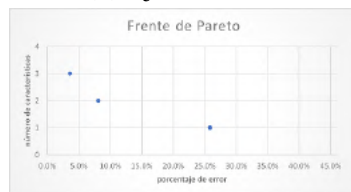
(h) Ejecución 23



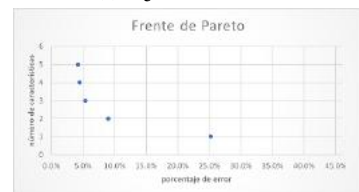
(i) Ejecución 24



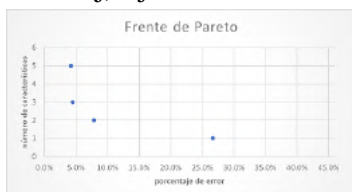
(j) Ejecución 25



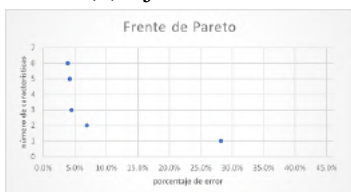
(k) Ejecución 26



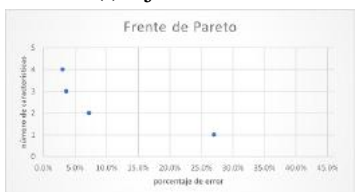
(l) Ejecución 27



(m) Ejecución 28

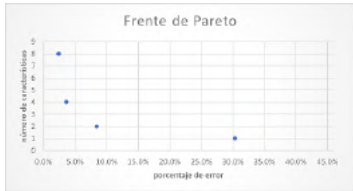


(n) Ejecución 29

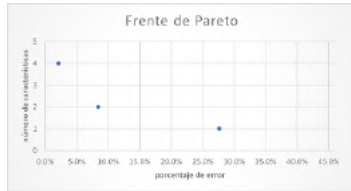


(ñ) Ejecución 30

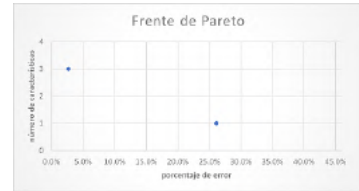
Figura B.2: Frentes de Pareto del Conjunto de datos musk-1 con FSGDE3. (2 - 2)



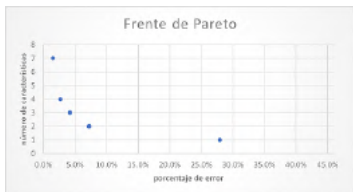
(a) Ejecución 1



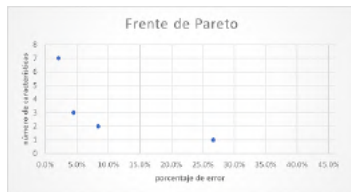
(b) Ejecución 2



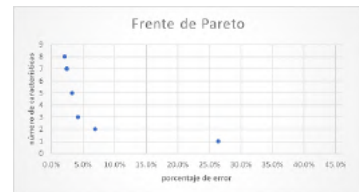
(c) Ejecución 3



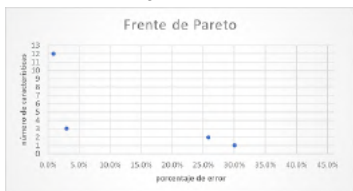
(d) Ejecución 4



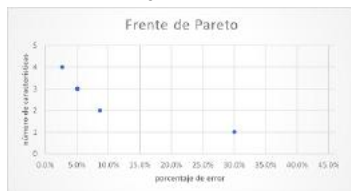
(e) Ejecución 5



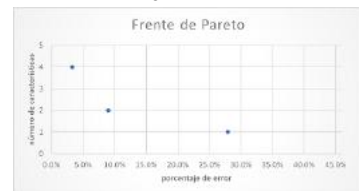
(f) Ejecución 6



(g) Ejecución 7



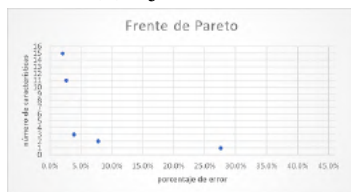
(h) Ejecución 8



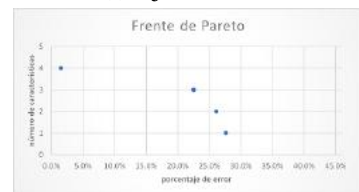
(i) Ejecución 9



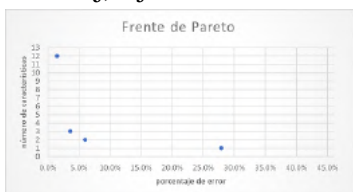
(j) Ejecución 10



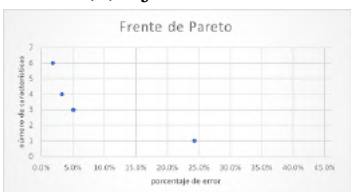
(k) Ejecución 11



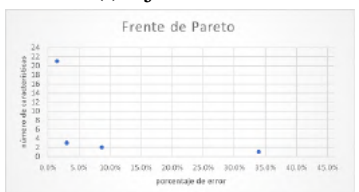
(l) Ejecución 12



(m) Ejecución 13

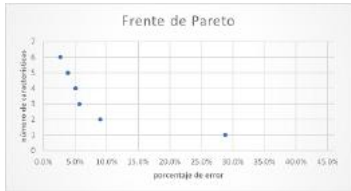


(n) Ejecución 14

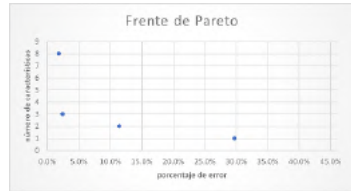


(ñ) Ejecución 15

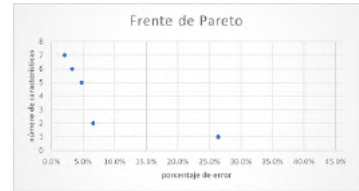
Figura B.3: Frentes de Pareto del Conjunto de datos musk-1 con FSNSGA II. (1 - 2)



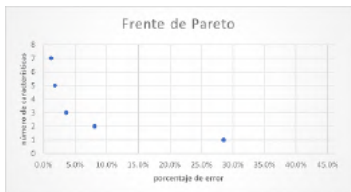
(a) Ejecución 16



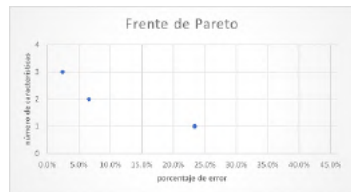
(b) Ejecución 17



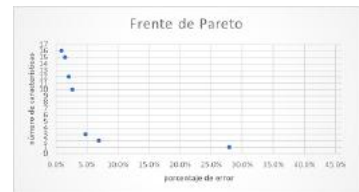
(c) Ejecución 18



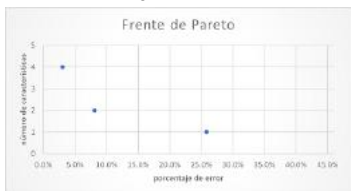
(d) Ejecución 19



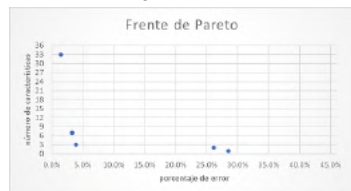
(e) Ejecución 20



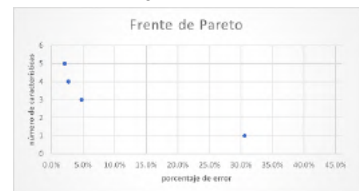
(f) Ejecución 21



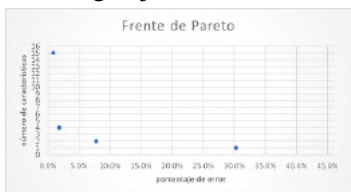
(g) Ejecución 22



(h) Ejecución 23



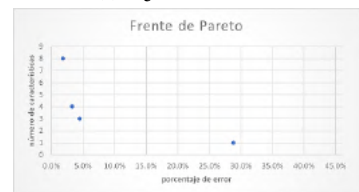
(i) Ejecución 24



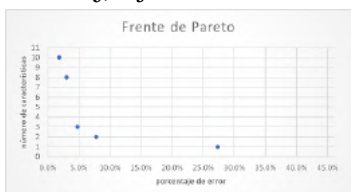
(j) Ejecución 25



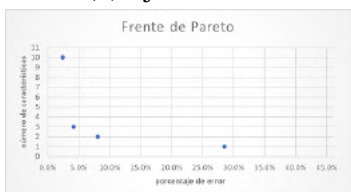
(k) Ejecución 26



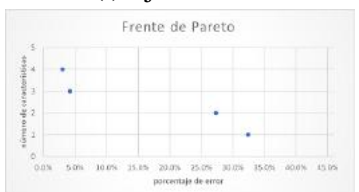
(l) Ejecución 27



(m) Ejecución 28



(n) Ejecución 29



(ñ) Ejecución 30

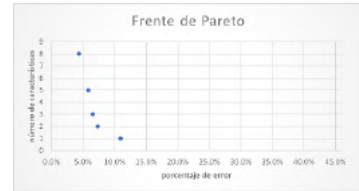
Figura B.4: Frentes de Pareto del Conjunto de datos musk-1 con FSNSGA II. (2 - 2)



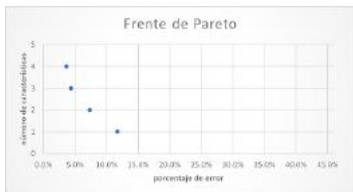
(a) Ejecución 1



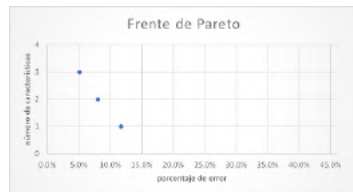
(b) Ejecución 2



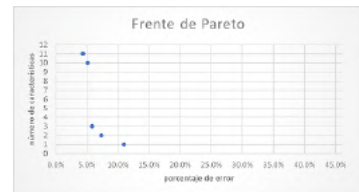
(c) Ejecución 3



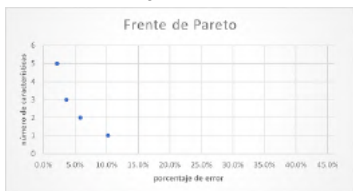
(d) Ejecución 4



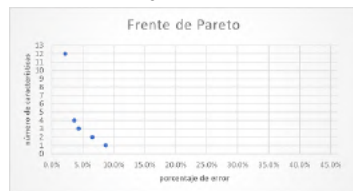
(e) Ejecución 5



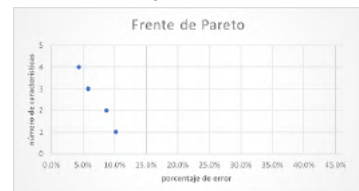
(f) Ejecución 6



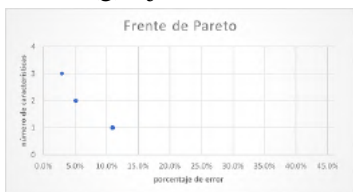
(g) Ejecución 7



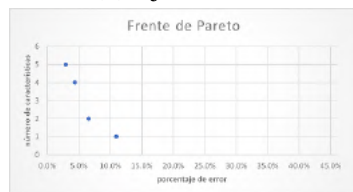
(h) Ejecución 8



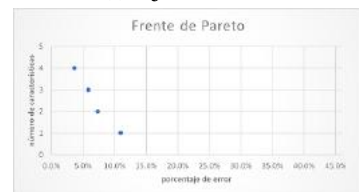
(i) Ejecución 9



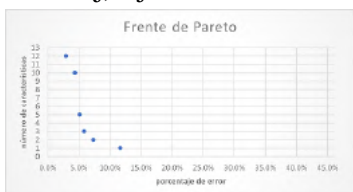
(j) Ejecución 10



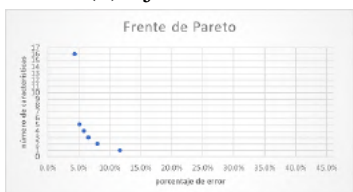
(k) Ejecución 11



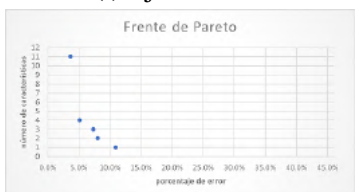
(l) Ejecución 12



(m) Ejecución 13

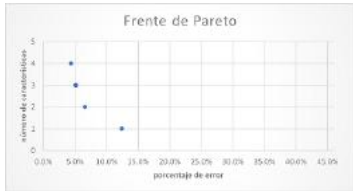


(n) Ejecución 14

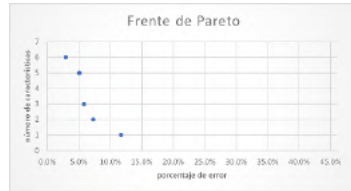


(ñ) Ejecución 15

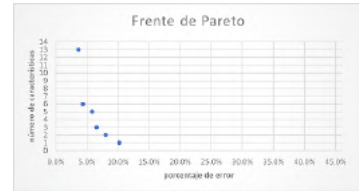
Figura B.5: Frentes de Pareto del Conjunto de datos parkinsons con FSGDE3. (1 - 2)



(a) Ejecución 16



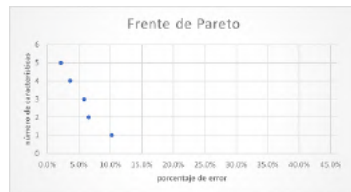
(b) Ejecución 17



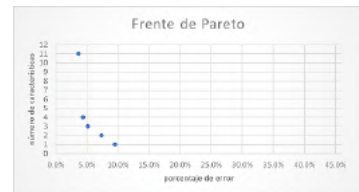
(c) Ejecución 18



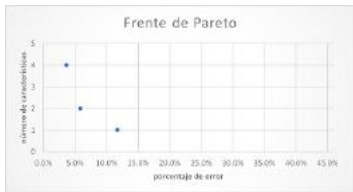
(d) Ejecución 19



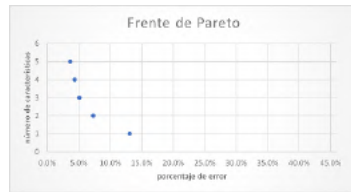
(e) Ejecución 20



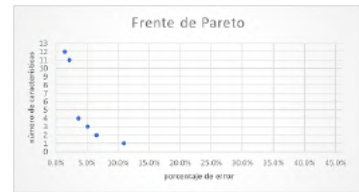
(f) Ejecución 21



(g) Ejecución 22



(h) Ejecución 23



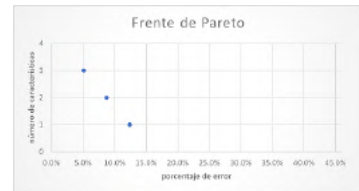
(i) Ejecución 24



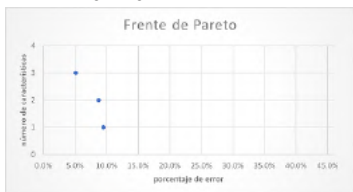
(j) Ejecución 25



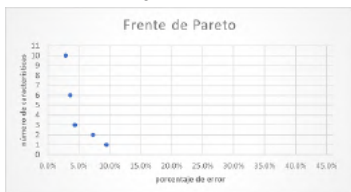
(k) Ejecución 26



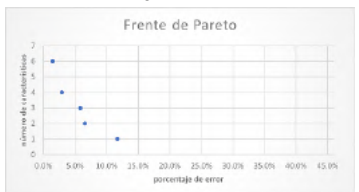
(l) Ejecución 27



(m) Ejecución 28

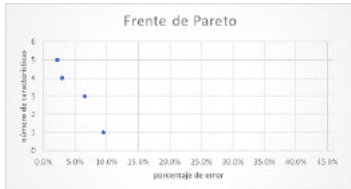


(n) Ejecución 29

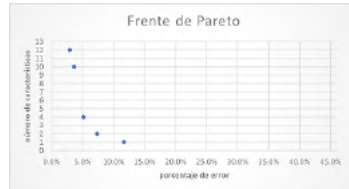


(ñ) Ejecución 30

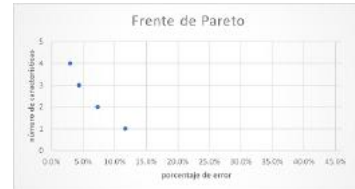
Figura B.6: Frentes de Pareto del Conjunto de datos parkinsons con FSGDE3. (2 - 2)



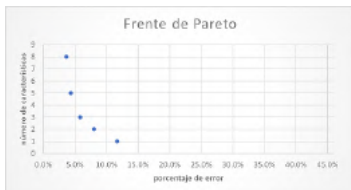
(a) Ejecución 1



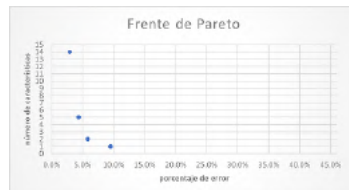
(b) Ejecución 2



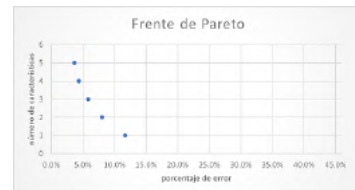
(c) Ejecución 3



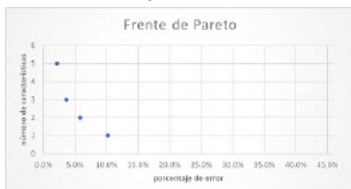
(d) Ejecución 4



(e) Ejecución 5



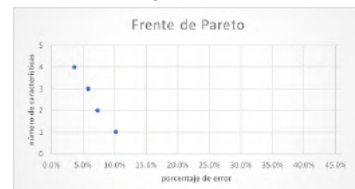
(f) Ejecución 6



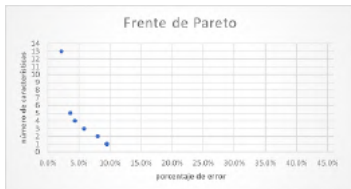
(g) Ejecución 7



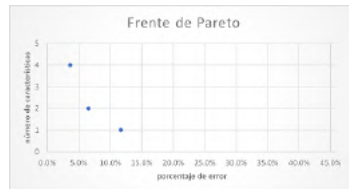
(h) Ejecución 8



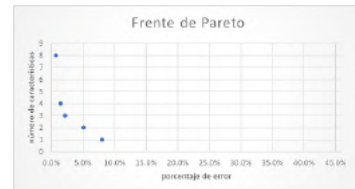
(i) Ejecución 9



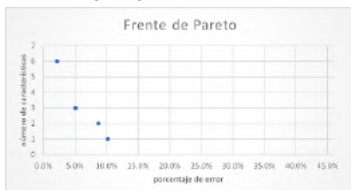
(j) Ejecución 10



(k) Ejecución 11



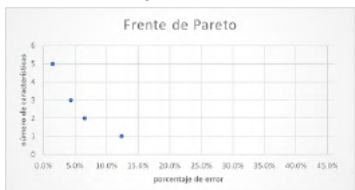
(l) Ejecución 12



(m) Ejecución 13



(n) Ejecución 14

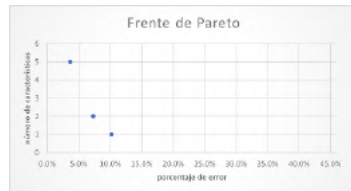


(ñ) Ejecución 15

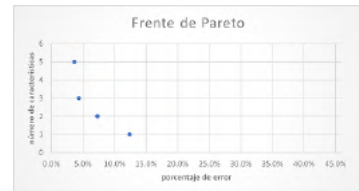
Figura B.7: Frentes de Pareto del Conjunto de datos parkinsons con FSNSGA II. (1 - 2)



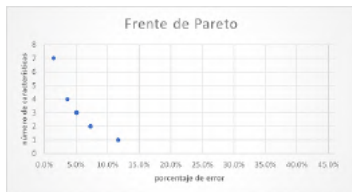
(a) Ejecución 16



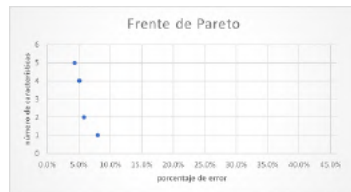
(b) Ejecución 17



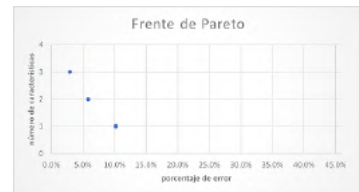
(c) Ejecución 18



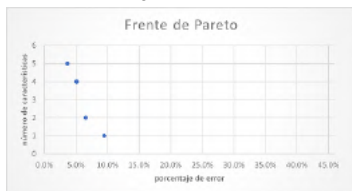
(d) Ejecución 19



(e) Ejecución 20



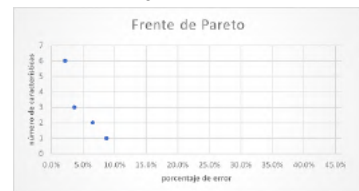
(f) Ejecución 21



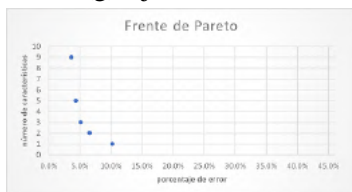
(g) Ejecución 22



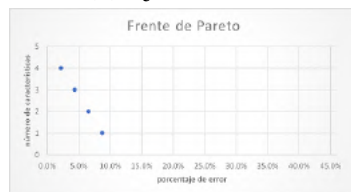
(h) Ejecución 23



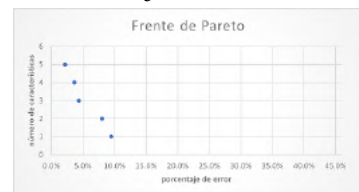
(i) Ejecución 24



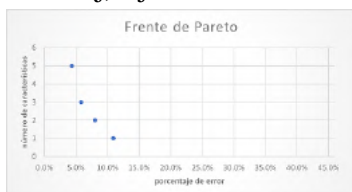
(j) Ejecución 25



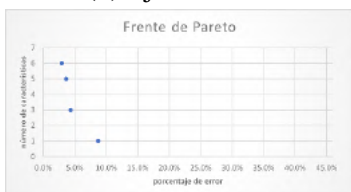
(k) Ejecución 26



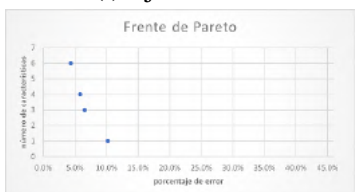
(l) Ejecución 27



(m) Ejecución 28



(n) Ejecución 29



(ñ) Ejecución 30

Figura B.8: Frentes de Pareto del Conjunto de datos parkinsons con FSNSGA II. (2 - 2)



(a) Ejecución 1



(b) Ejecución 2



(c) Ejecución 3



(d) Ejecución 4



(e) Ejecución 5



(f) Ejecución 6



(g) Ejecución 7



(h) Ejecución 8



(i) Ejecución 9



(j) Ejecución 10



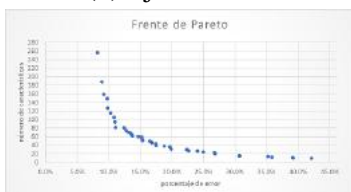
(k) Ejecución 11



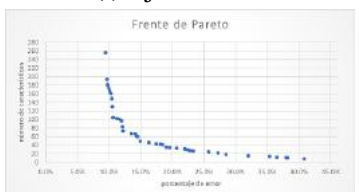
(l) Ejecución 12



(m) Ejecución 13



(n) Ejecución 14

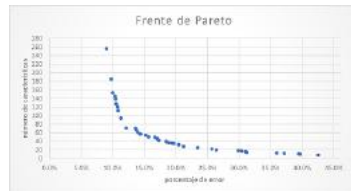


(ñ) Ejecución 15

Figura B.9: Frentes de Pareto del Conjunto de datos semeion con FSGDE3. (1 - 2)



(a) Ejecución 16



(b) Ejecución 17



(c) Ejecución 18



(d) Ejecución 19



(e) Ejecución 20



(f) Ejecución 21



(g) Ejecución 22



(h) Ejecución 23



(i) Ejecución 24



(j) Ejecución 25



(k) Ejecución 26



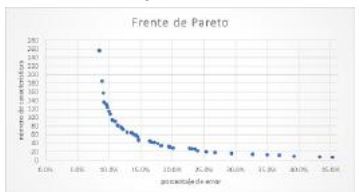
(l) Ejecución 27



(m) Ejecución 28

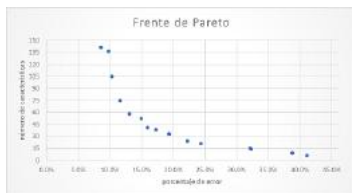


(n) Ejecución 29

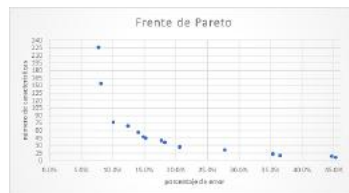


(ñ) Ejecución 30

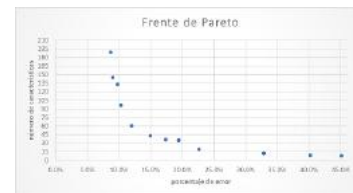
Figura B.10: Frentes de Pareto del Conjunto de datos semeion con FSGDE3. (2 - 2)



(a) Ejecución 1



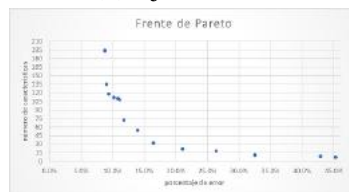
(b) Ejecución 2



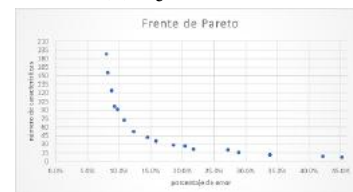
(c) Ejecución 3



(d) Ejecución 4



(e) Ejecución 5



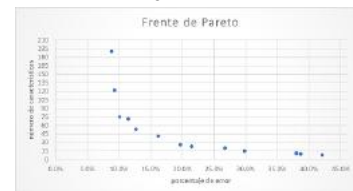
(f) Ejecución 6



(g) Ejecución 7



(h) Ejecución 8



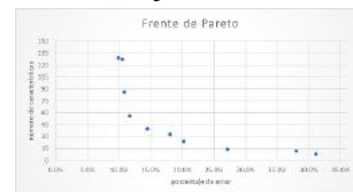
(i) Ejecución 9



(j) Ejecución 10



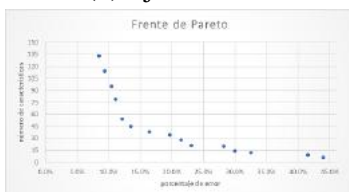
(k) Ejecución 11



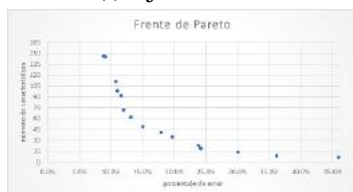
(l) Ejecución 12



(m) Ejecución 13



(n) Ejecución 14



(ñ) Ejecución 15

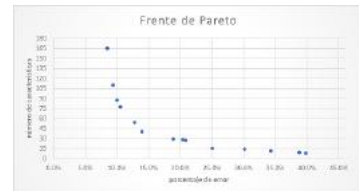
Figura B.11: Frentes de Pareto del Conjunto de datos semeion con FSNSGA II. (1 - 2)



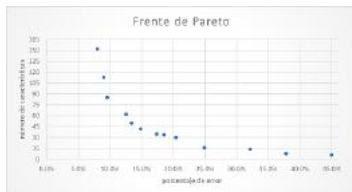
(a) Ejecución 16



(b) Ejecución 17



(c) Ejecución 18



(d) Ejecución 19



(e) Ejecución 20



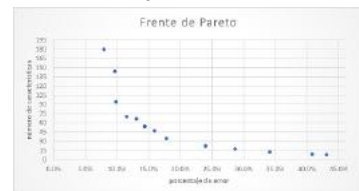
(f) Ejecución 21



(g) Ejecución 22



(h) Ejecución 23



(i) Ejecución 24



(j) Ejecución 25



(k) Ejecución 26



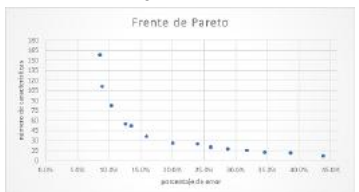
(l) Ejecución 27



(m) Ejecución 28

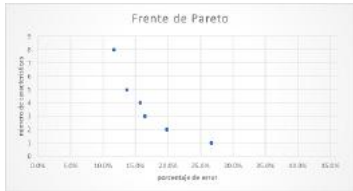


(n) Ejecución 29

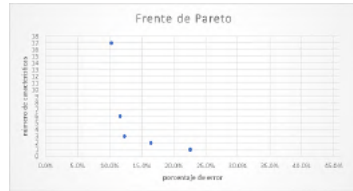


(ñ) Ejecución 30

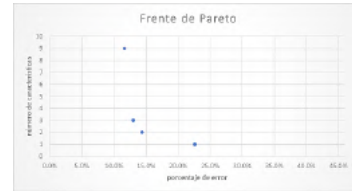
Figura B.12: Frentes de Pareto del Conjunto de datos semeion con FSNSGA II. (2 - 2)



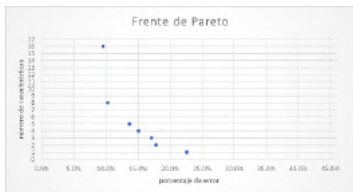
(a) Ejecución 1



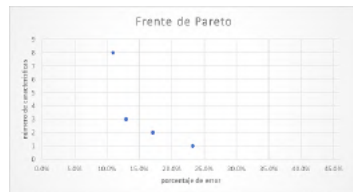
(b) Ejecución 2



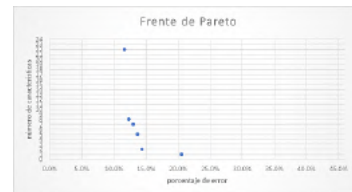
(c) Ejecución 3



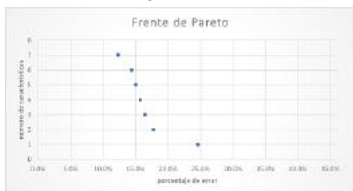
(d) Ejecución 4



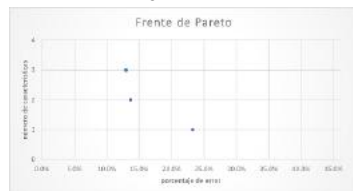
(e) Ejecución 5



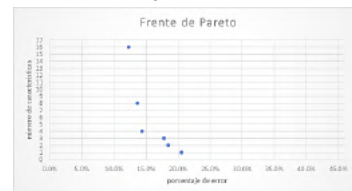
(f) Ejecución 6



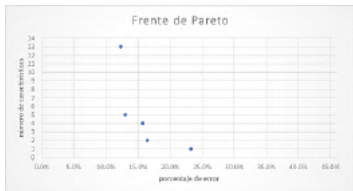
(g) Ejecución 7



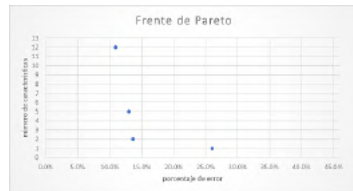
(h) Ejecución 8



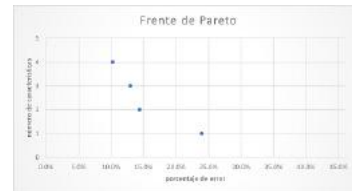
(i) Ejecución 9



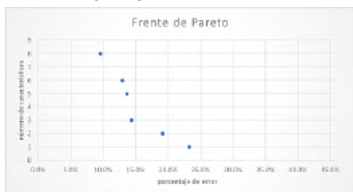
(j) Ejecución 10



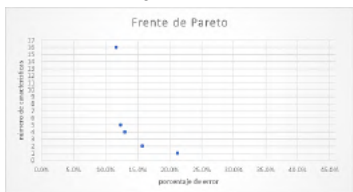
(k) Ejecución 11



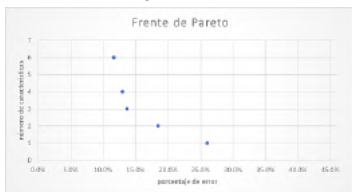
(l) Ejecución 12



(m) Ejecución 13

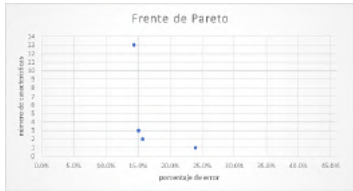


(n) Ejecución 14

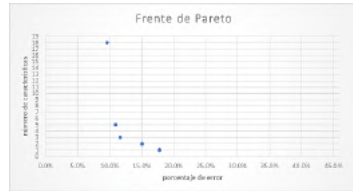


(ñ) Ejecución 15

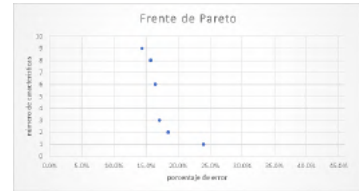
Figura B.13: Frentes de Pareto del Conjunto de datos sonar con FSGDE3. (1 - 2)



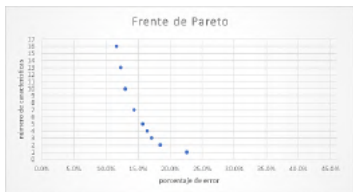
(a) Ejecución 16



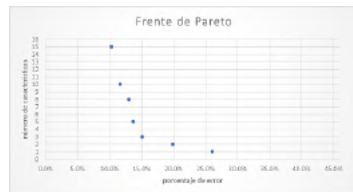
(b) Ejecución 17



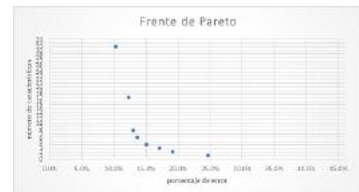
(c) Ejecución 18



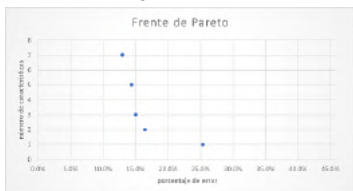
(d) Ejecución 19



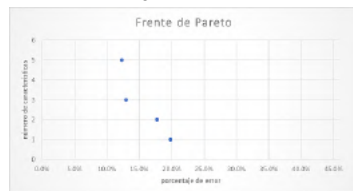
(e) Ejecución 20



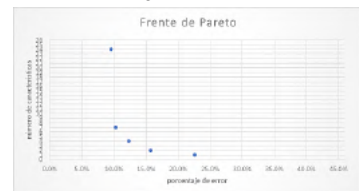
(f) Ejecución 21



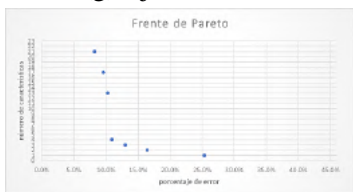
(g) Ejecución 22



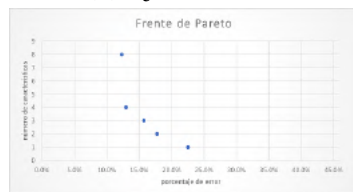
(h) Ejecución 23



(i) Ejecución 24



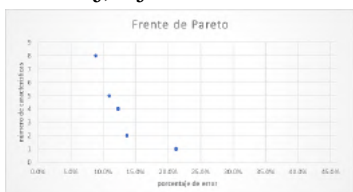
(j) Ejecución 25



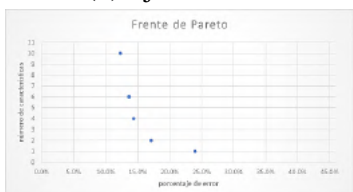
(k) Ejecución 26



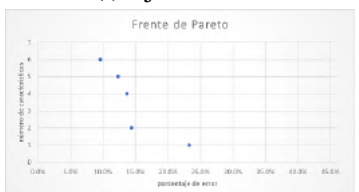
(l) Ejecución 27



(m) Ejecución 28

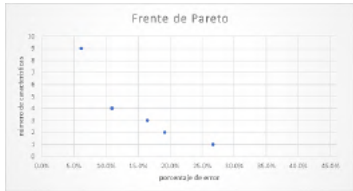


(n) Ejecución 29

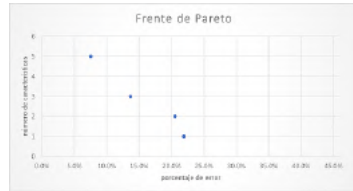


(ñ) Ejecución 30

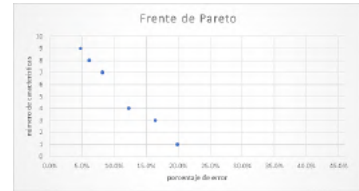
Figura B.14: Frentes de Pareto del Conjunto de datos sonar con FSGDE3. (2 - 2)



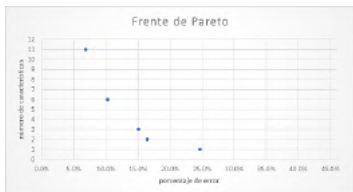
(a) Ejecución 1



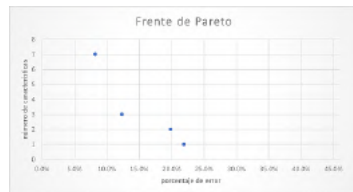
(b) Ejecución 2



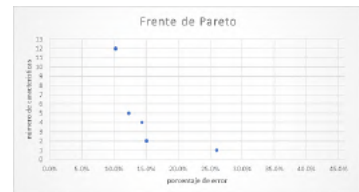
(c) Ejecución 3



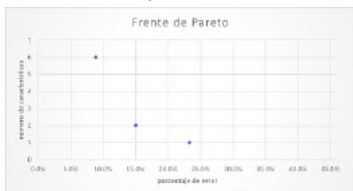
(d) Ejecución 4



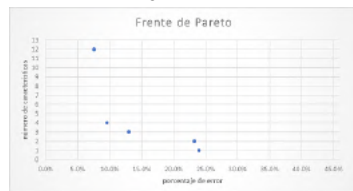
(e) Ejecución 5



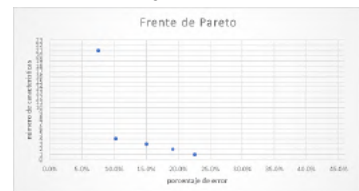
(f) Ejecución 6



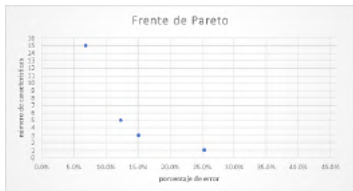
(g) Ejecución 7



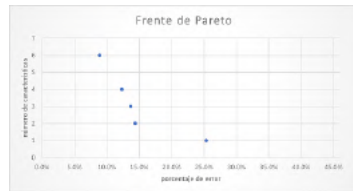
(h) Ejecución 8



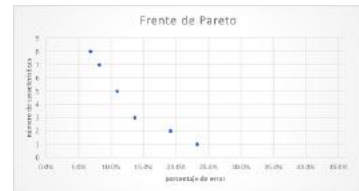
(i) Ejecución 9



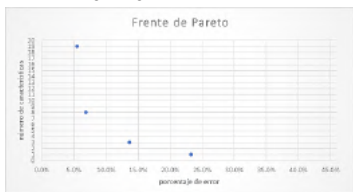
(j) Ejecución 10



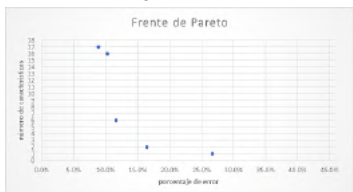
(k) Ejecución 11



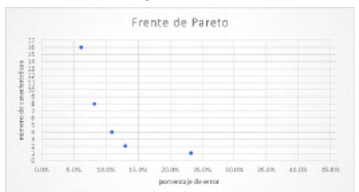
(l) Ejecución 12



(m) Ejecución 13

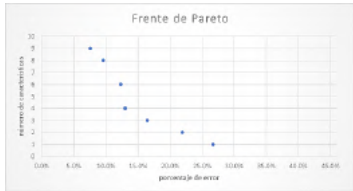


(n) Ejecución 14

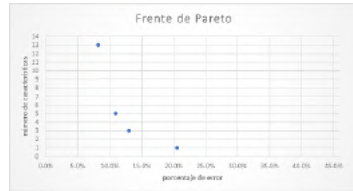


(ñ) Ejecución 15

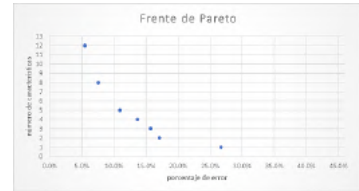
Figura B.15: Frentes de Pareto del Conjunto de datos sonar con FSNSGA II. (1 - 2)



(a) Ejecución 16



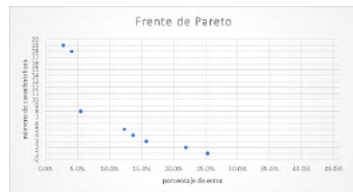
(b) Ejecución 17



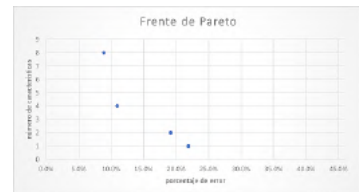
(c) Ejecución 18



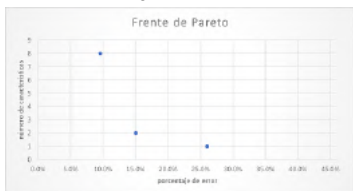
(d) Ejecución 19



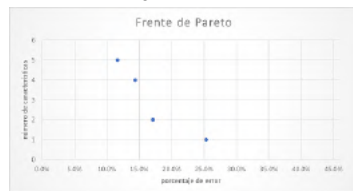
(e) Ejecución 20



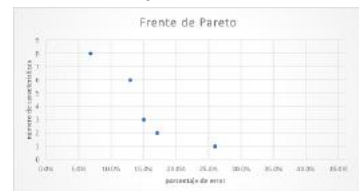
(f) Ejecución 21



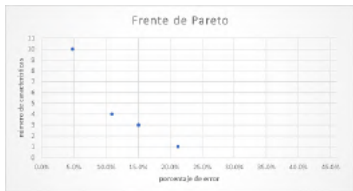
(g) Ejecución 22



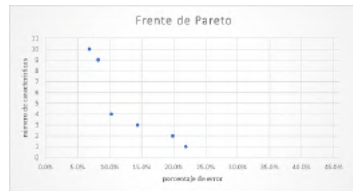
(h) Ejecución 23



(i) Ejecución 24



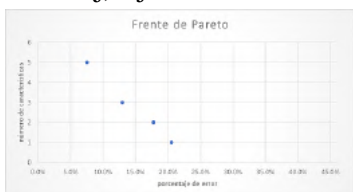
(j) Ejecución 25



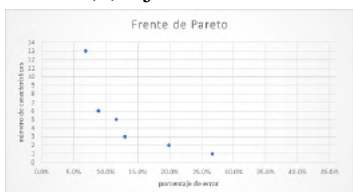
(k) Ejecución 26



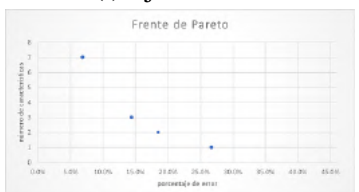
(l) Ejecución 27



(m) Ejecución 28

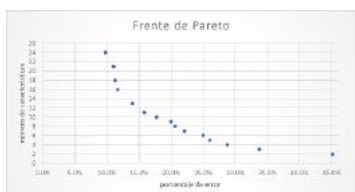


(n) Ejecución 29

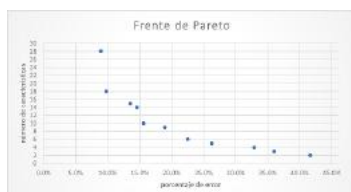


(ñ) Ejecución 30

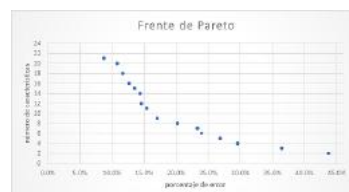
Figura B.16: Frentes de Pareto del Conjunto de datos sonar con FSNSGA II. (2 - 2)



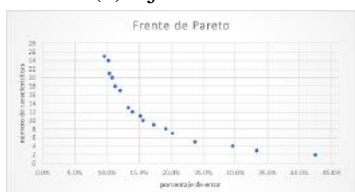
(a) Ejecución 1



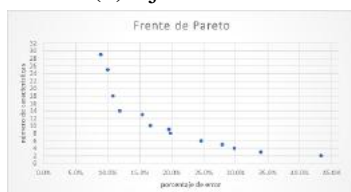
(b) Ejecución 2



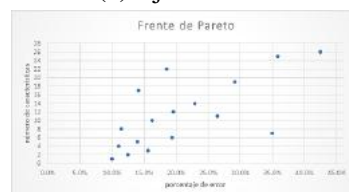
(c) Ejecución 3



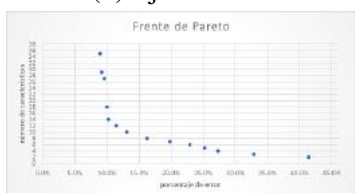
(d) Ejecución 4



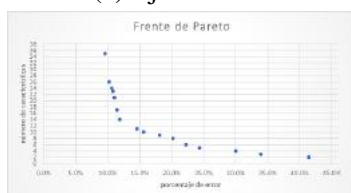
(e) Ejecución 5



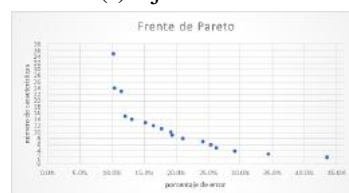
(f) Ejecución 6



(g) Ejecución 7



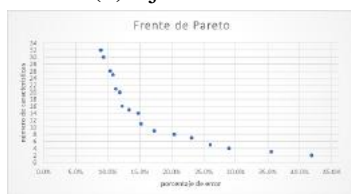
(h) Ejecución 8



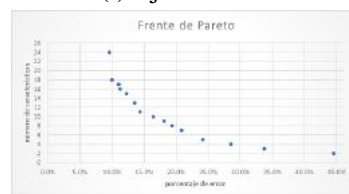
(i) Ejecución 9



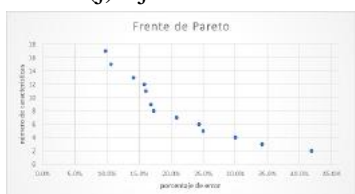
(j) Ejecución 10



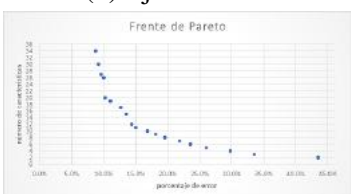
(k) Ejecución 11



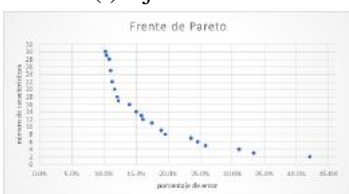
(l) Ejecución 12



(m) Ejecución 13

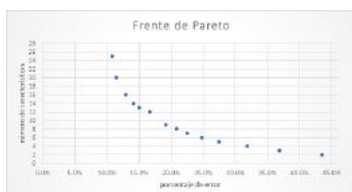


(n) Ejecución 14

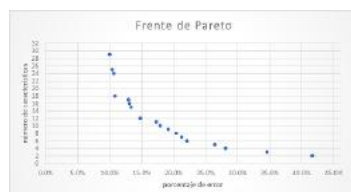


(ñ) Ejecución 15

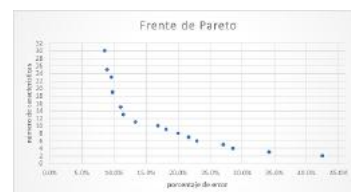
Figura B.17: Frentes de Pareto del Conjunto de datos soybean con FSGDE3. (1 - 2)



(a) Ejecución 16



(b) Ejecución 17



(c) Ejecución 18



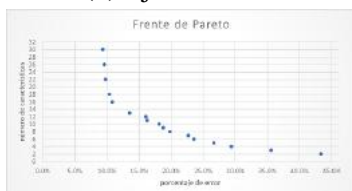
(d) Ejecución 19



(e) Ejecución 20



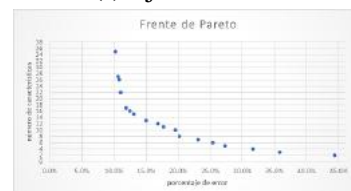
(f) Ejecución 21



(g) Ejecución 22



(h) Ejecución 23



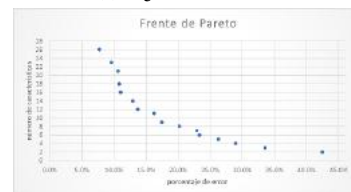
(i) Ejecución 24



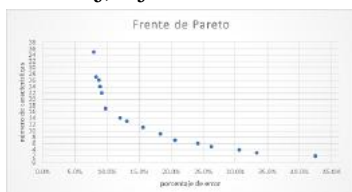
(j) Ejecución 25



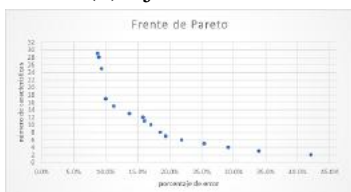
(k) Ejecución 26



(l) Ejecución 27



(m) Ejecución 28

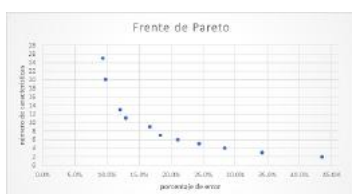


(n) Ejecución 29



(ñ) Ejecución 30

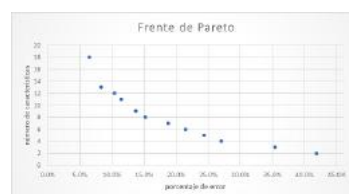
Figura B.18: Frentes de Pareto del Conjunto de datos soybean con FSGDE3. (2 - 2)



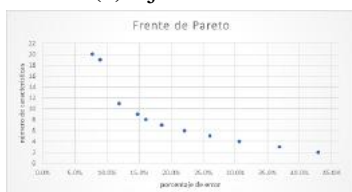
(a) Ejecución 1



(b) Ejecución 2



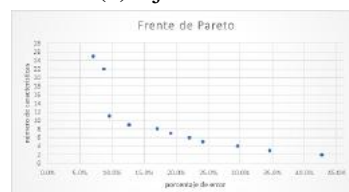
(c) Ejecución 3



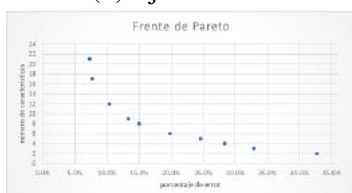
(d) Ejecución 4



(e) Ejecución 5



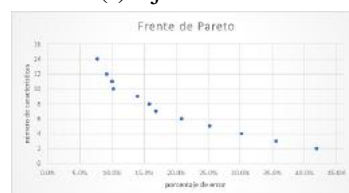
(f) Ejecución 6



(g) Ejecución 7



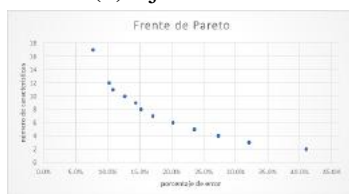
(h) Ejecución 8



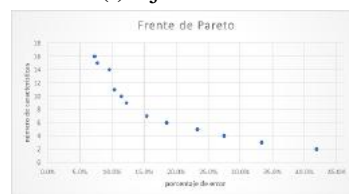
(i) Ejecución 9



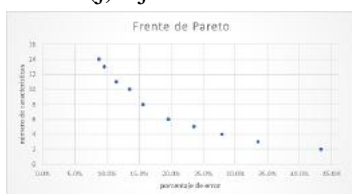
(j) Ejecución 10



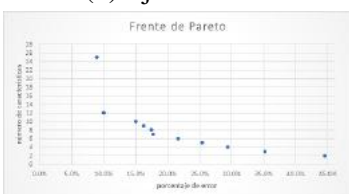
(k) Ejecución 11



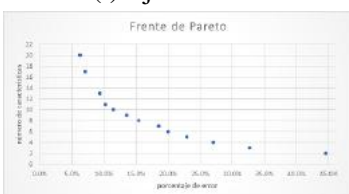
(l) Ejecución 12



(m) Ejecución 13



(n) Ejecución 14



(ñ) Ejecución 15

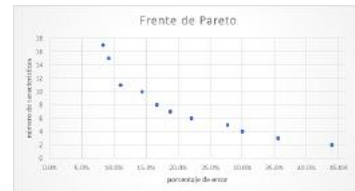
Figura B.19: Frentes de Pareto del Conjunto de datos soybean con FSNSGA II. (1 - 2)



(a) Ejecución 16



(b) Ejecución 17



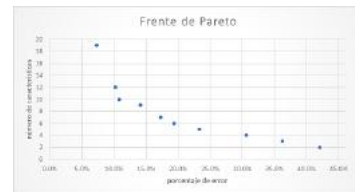
(c) Ejecución 18



(d) Ejecución 19



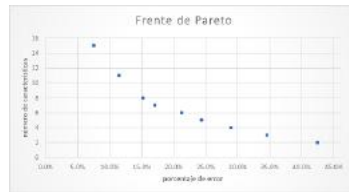
(e) Ejecución 20



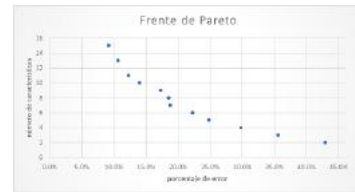
(f) Ejecución 21



(g) Ejecución 22



(h) Ejecución 23



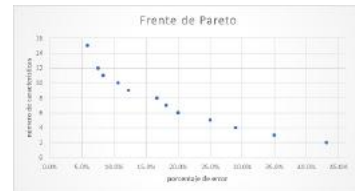
(i) Ejecución 24



(j) Ejecución 25



(k) Ejecución 26



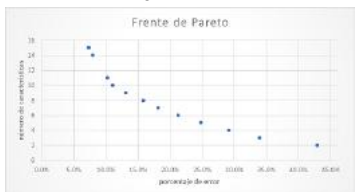
(l) Ejecución 27



(m) Ejecución 28

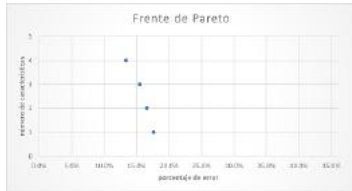


(n) Ejecución 29

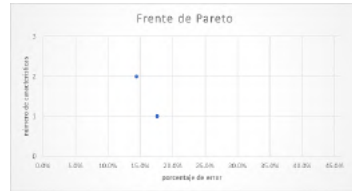


(ñ) Ejecución 30

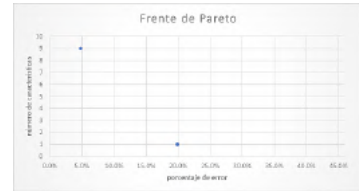
Figura B.20: Frentes de Pareto del Conjunto de datos soybean con FSNSGA II. (2 - 2)



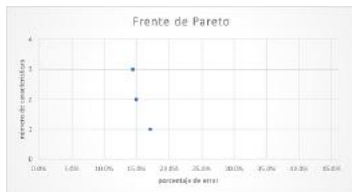
(a) Ejecución 1



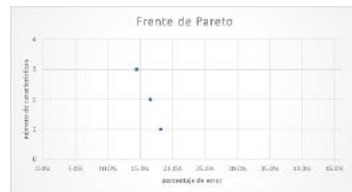
(b) Ejecución 2



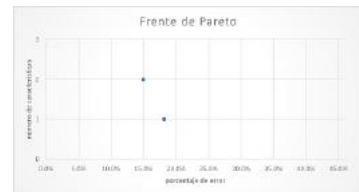
(c) Ejecución 3



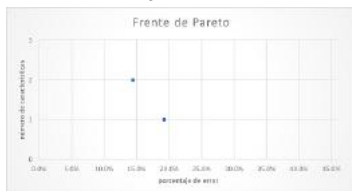
(d) Ejecución 4



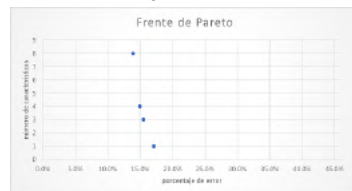
(e) Ejecución 5



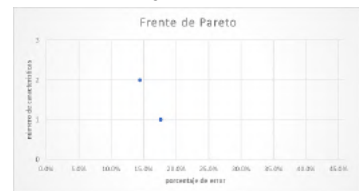
(f) Ejecución 6



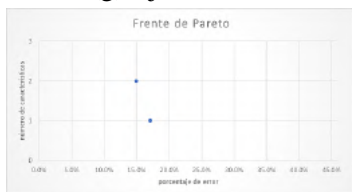
(g) Ejecución 7



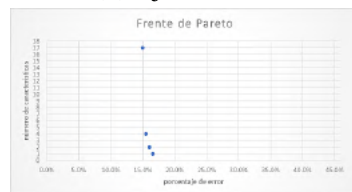
(h) Ejecución 8



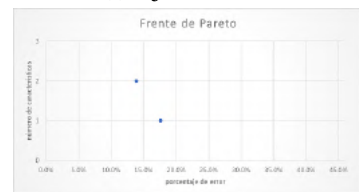
(i) Ejecución 9



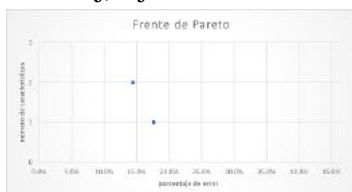
(j) Ejecución 10



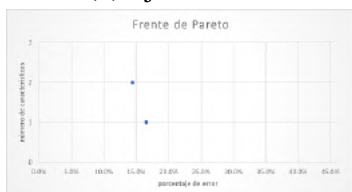
(k) Ejecución 11



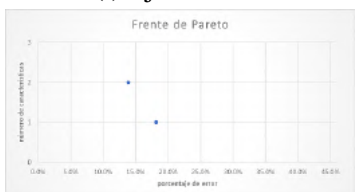
(l) Ejecución 12



(m) Ejecución 13

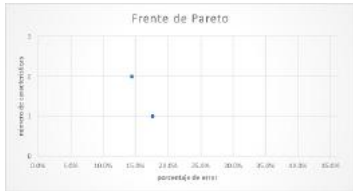


(n) Ejecución 14



(ñ) Ejecución 15

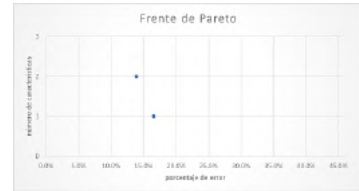
Figura B.21: Frentes de Pareto del Conjunto de datos spectf con FSGDE3. (1 - 2)



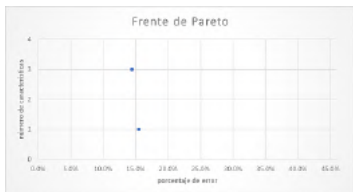
(a) Ejecución 16



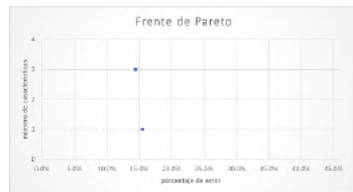
(b) Ejecución 17



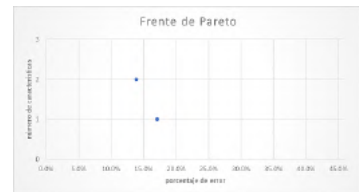
(c) Ejecución 18



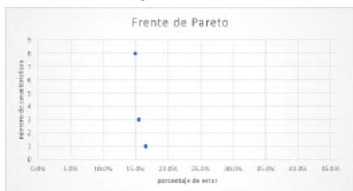
(d) Ejecución 19



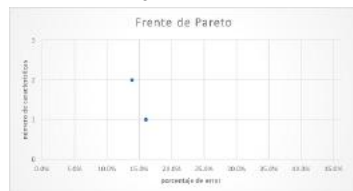
(e) Ejecución 20



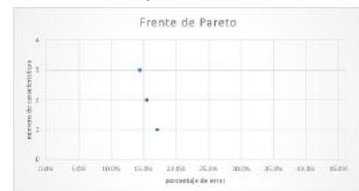
(f) Ejecución 21



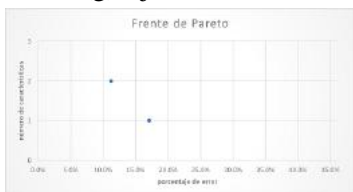
(g) Ejecución 22



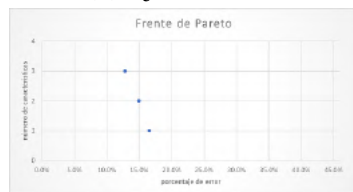
(h) Ejecución 23



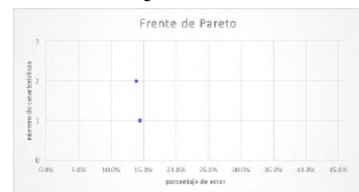
(i) Ejecución 24



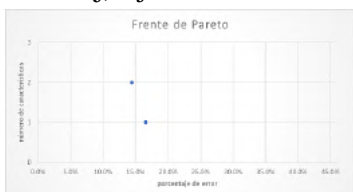
(j) Ejecución 25



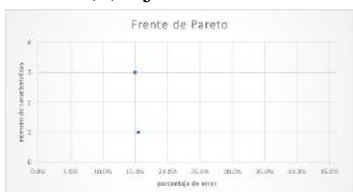
(k) Ejecución 26



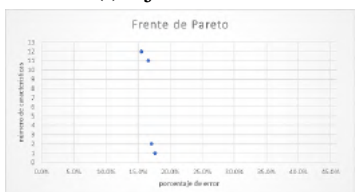
(l) Ejecución 27



(m) Ejecución 28

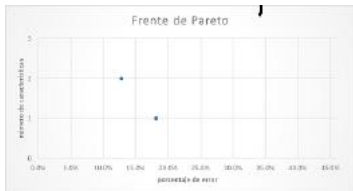


(n) Ejecución 29

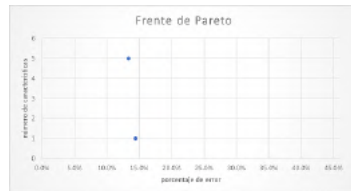


(ñ) Ejecución 30

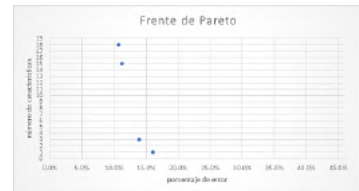
Figura B.22: Frentes de Pareto del Conjunto de datos spectf con FSGDE3. (2 - 2)



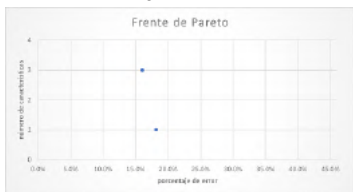
(a) Ejecución 1



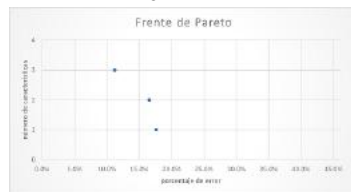
(b) Ejecución 2



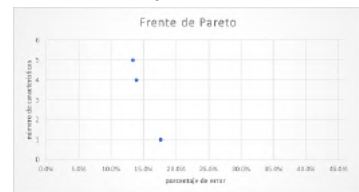
(c) Ejecución 3



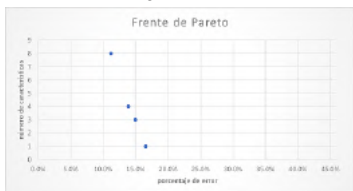
(d) Ejecución 4



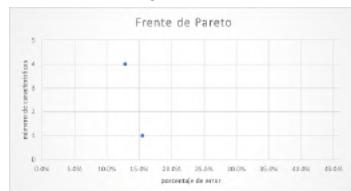
(e) Ejecución 5



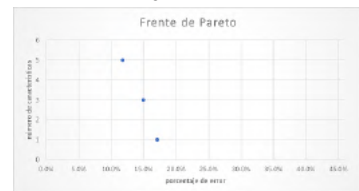
(f) Ejecución 6



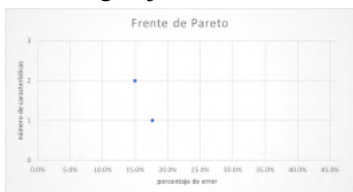
(g) Ejecución 7



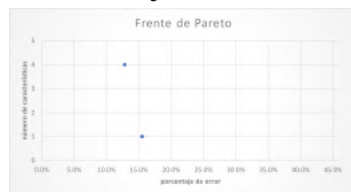
(h) Ejecución 8



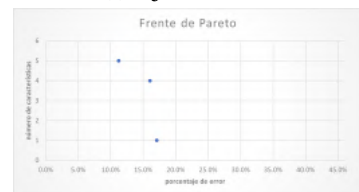
(i) Ejecución 9



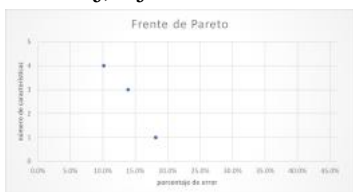
(j) Ejecución 10



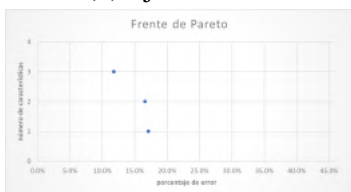
(k) Ejecución 11



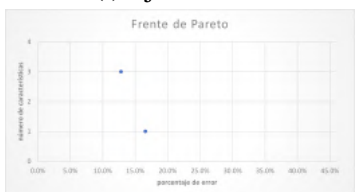
(l) Ejecución 12



(m) Ejecución 13

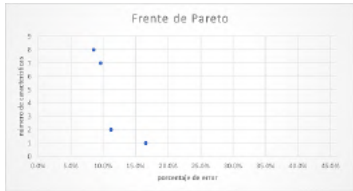


(n) Ejecución 14

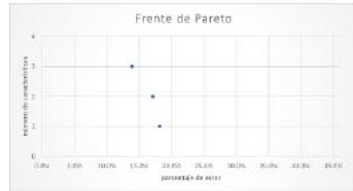


(ñ) Ejecución 15

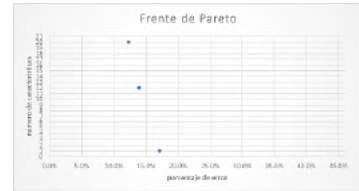
Figura B.23: Frentes de Pareto del Conjunto de datos spectf con FSNSGA II. (1 - 2)



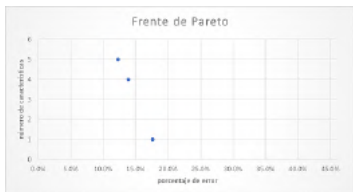
(a) Ejecución 16



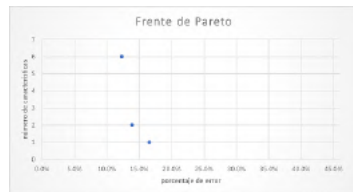
(b) Ejecución 17



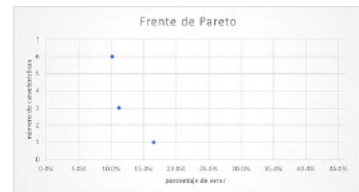
(c) Ejecución 18



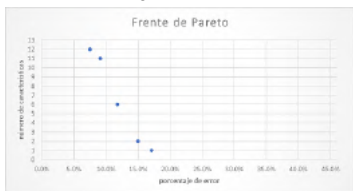
(d) Ejecución 19



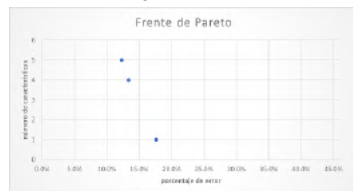
(e) Ejecución 20



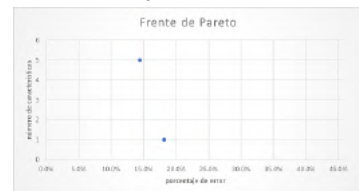
(f) Ejecución 21



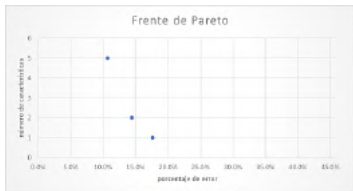
(g) Ejecución 22



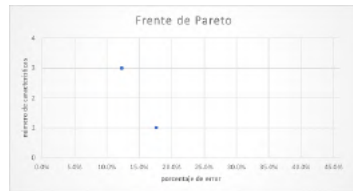
(h) Ejecución 23



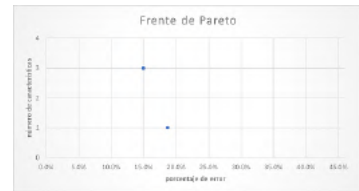
(i) Ejecución 24



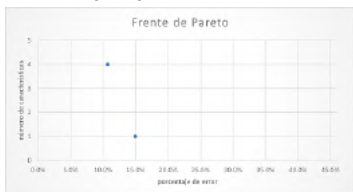
(j) Ejecución 25



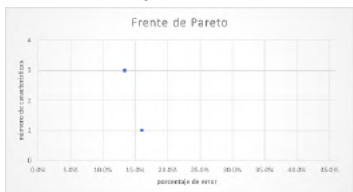
(k) Ejecución 26



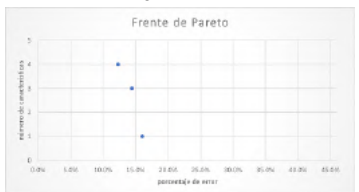
(l) Ejecución 27



(m) Ejecución 28

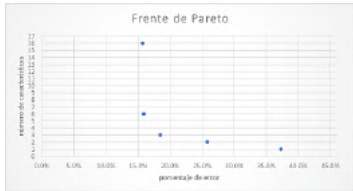


(n) Ejecución 29

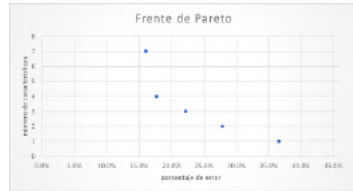


(ñ) Ejecución 30

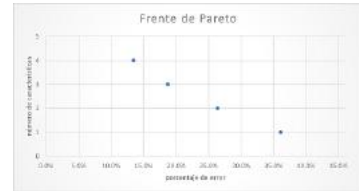
Figura B.24: Frentes de Pareto del Conjunto de datos spectf con FSNSGA II. (2 - 2)



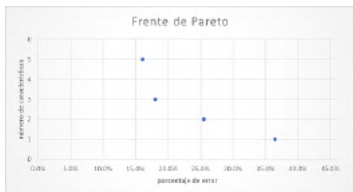
(a) Ejecución 1



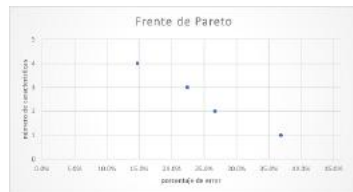
(b) Ejecución 2



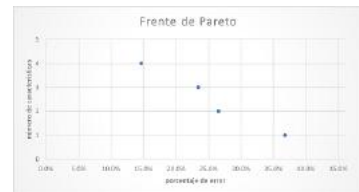
(c) Ejecución 3



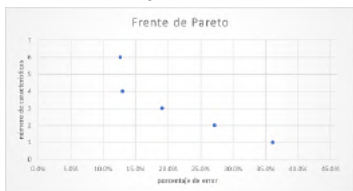
(d) Ejecución 4



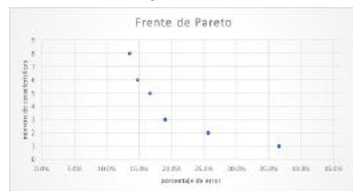
(e) Ejecución 5



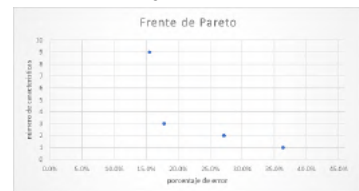
(f) Ejecución 6



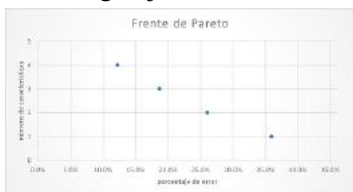
(g) Ejecución 7



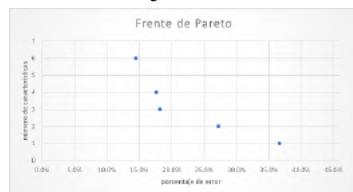
(h) Ejecución 8



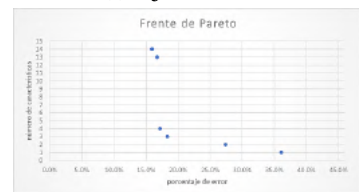
(i) Ejecución 9



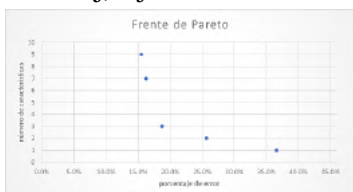
(j) Ejecución 10



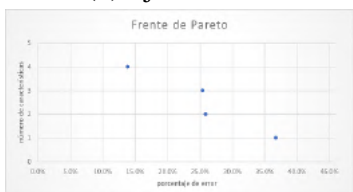
(k) Ejecución 11



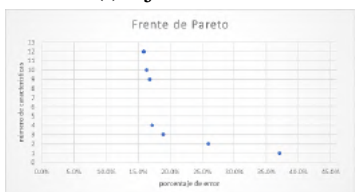
(l) Ejecución 12



(m) Ejecución 13

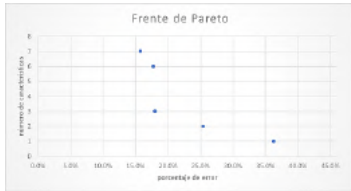


(n) Ejecución 14

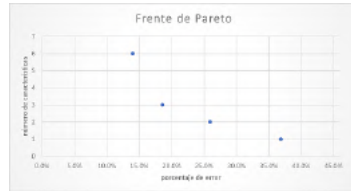


(ñ) Ejecución 15

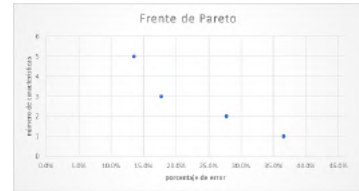
Figura B.25: Frentes de Pareto del Conjunto de datos splice con FSGDE3. (1 - 2)



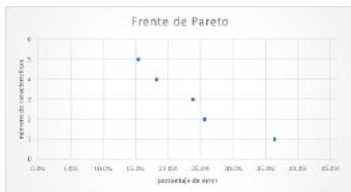
(a) Ejecución 16



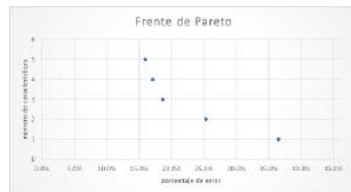
(b) Ejecución 17



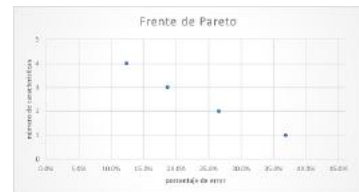
(c) Ejecución 18



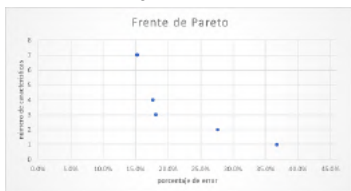
(d) Ejecución 19



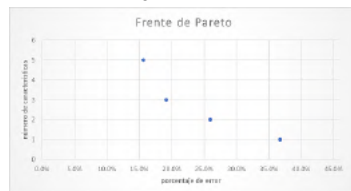
(e) Ejecución 20



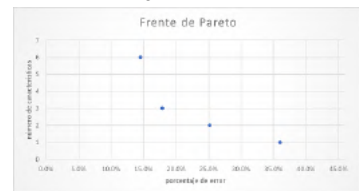
(f) Ejecución 21



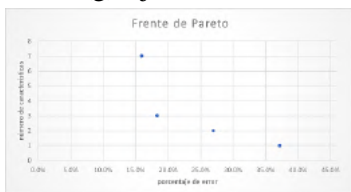
(g) Ejecución 22



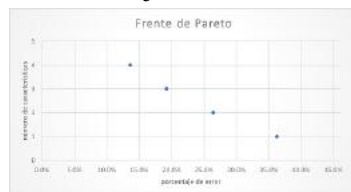
(h) Ejecución 23



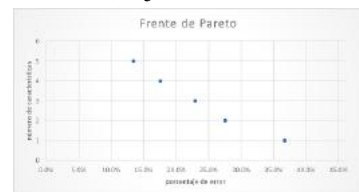
(i) Ejecución 24



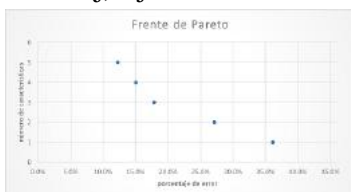
(j) Ejecución 25



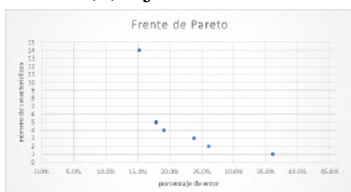
(k) Ejecución 26



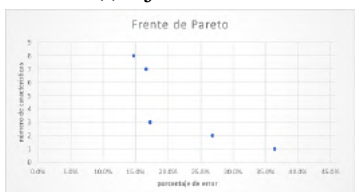
(l) Ejecución 27



(m) Ejecución 28

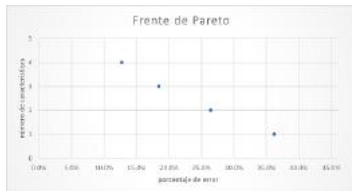


(n) Ejecución 29

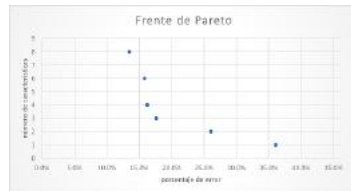


(ñ) Ejecución 30

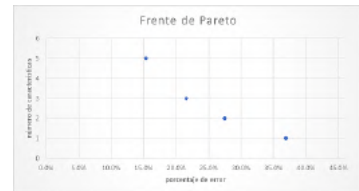
Figura B.26: Frentes de Pareto del Conjunto de datos splice con FSGDE3. (2 - 2)



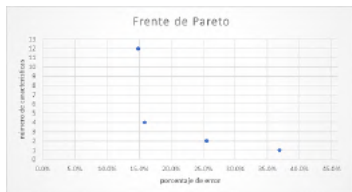
(a) Ejecución 1



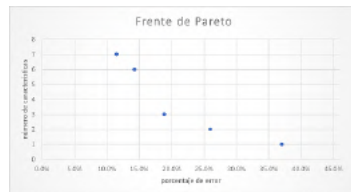
(b) Ejecución 2



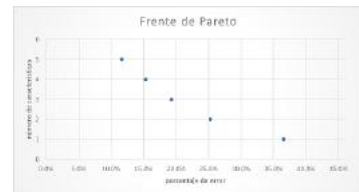
(c) Ejecución 3



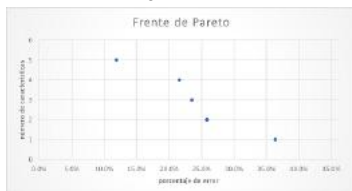
(d) Ejecución 4



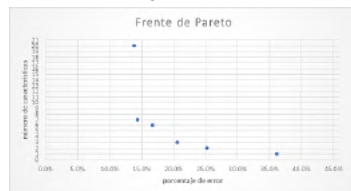
(e) Ejecución 5



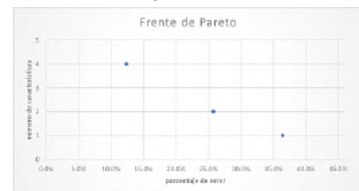
(f) Ejecución 6



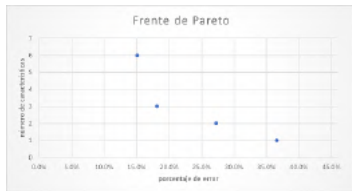
(g) Ejecución 7



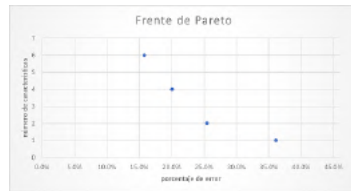
(h) Ejecución 8



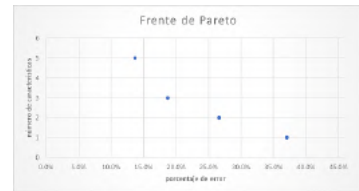
(i) Ejecución 9



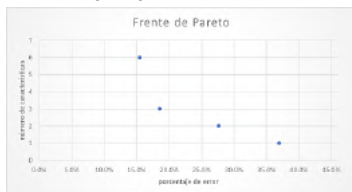
(j) Ejecución 10



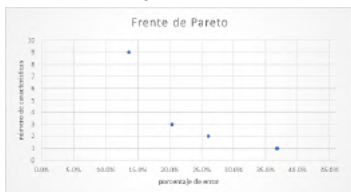
(k) Ejecución 11



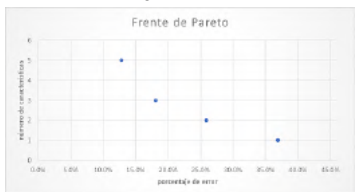
(l) Ejecución 12



(m) Ejecución 13

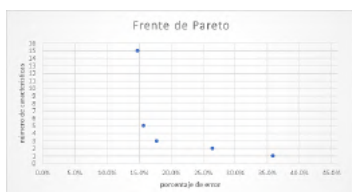


(n) Ejecución 14

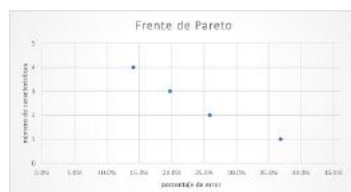


(ñ) Ejecución 15

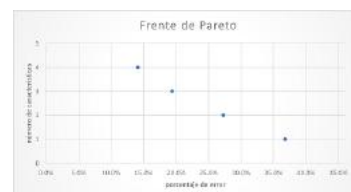
Figura B.27: Frentes de Pareto del Conjunto de datos splice con FSNSGA II. (1 - 2)



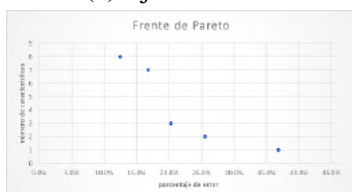
(a) Ejecución 16



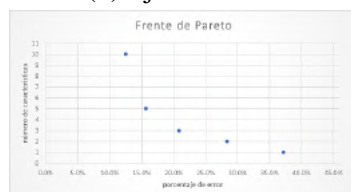
(b) Ejecución 17



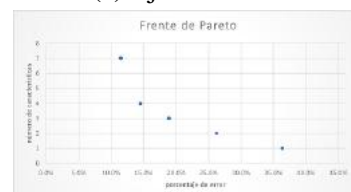
(c) Ejecución 18



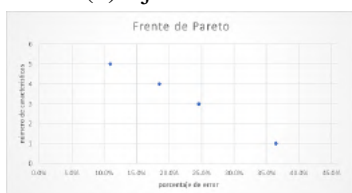
(d) Ejecución 19



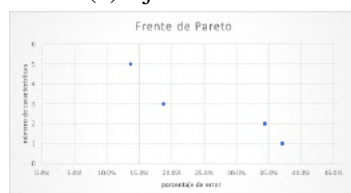
(e) Ejecución 20



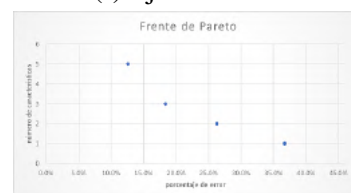
(f) Ejecución 21



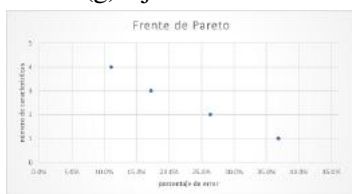
(g) Ejecución 22



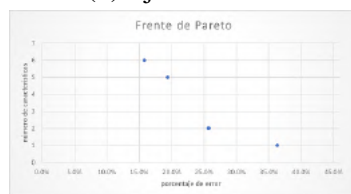
(h) Ejecución 23



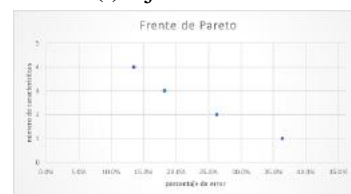
(i) Ejecución 24



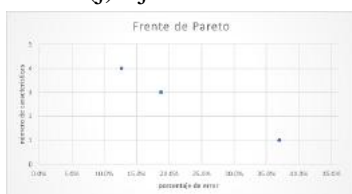
(j) Ejecución 25



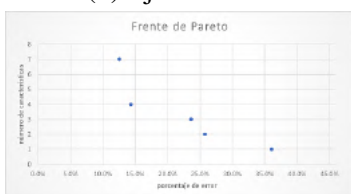
(k) Ejecución 26



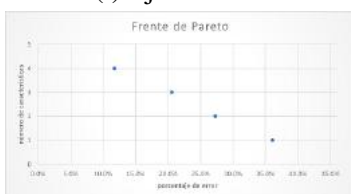
(l) Ejecución 27



(m) Ejecución 28

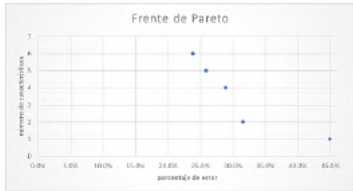


(n) Ejecución 29



(ñ) Ejecución 30

Figura B.28: Frentes de Pareto del Conjunto de datos splice con FSNSGA II. (2 - 2)



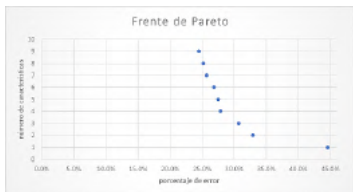
(a) Ejecución 1



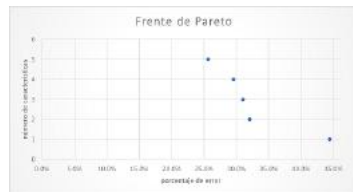
(b) Ejecución 2



(c) Ejecución 3



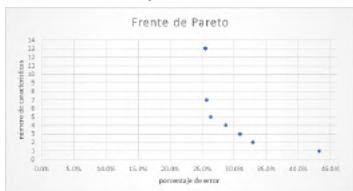
(d) Ejecución 4



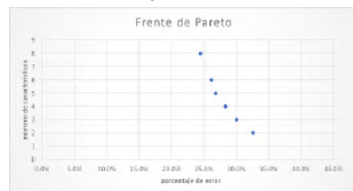
(e) Ejecución 5



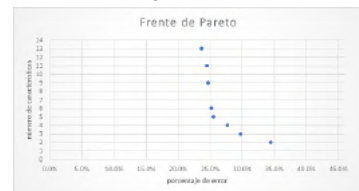
(f) Ejecución 6



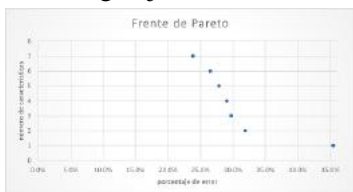
(g) Ejecución 7



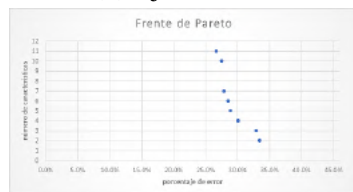
(h) Ejecución 8



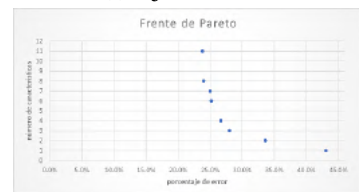
(i) Ejecución 9



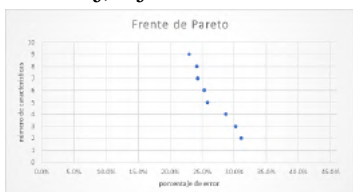
(j) Ejecución 10



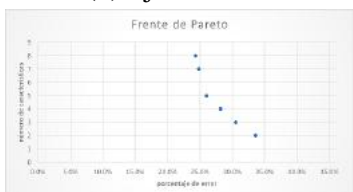
(k) Ejecución 11



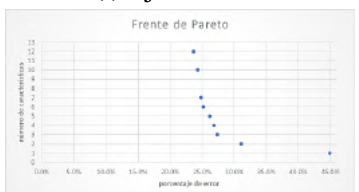
(l) Ejecución 12



(m) Ejecución 13

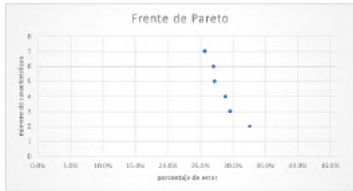


(n) Ejecución 14

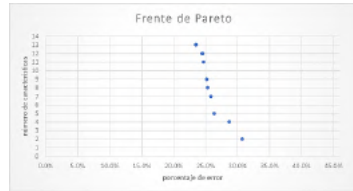


(ñ) Ejecución 15

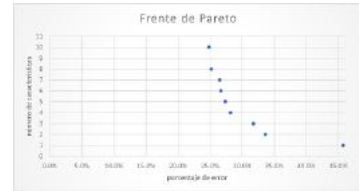
Figura B.29: Frentes de Pareto del Conjunto de datos vehicle con FSGDE3. (1 - 2)



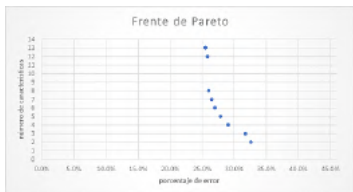
(a) Ejecución 16



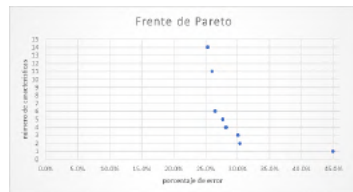
(b) Ejecución 17



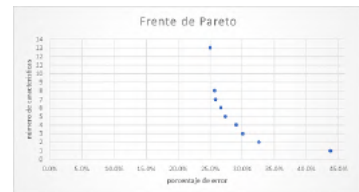
(c) Ejecución 18



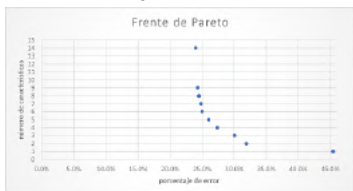
(d) Ejecución 19



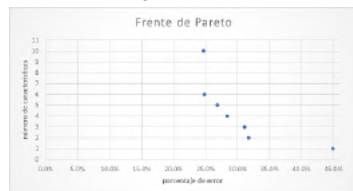
(e) Ejecución 20



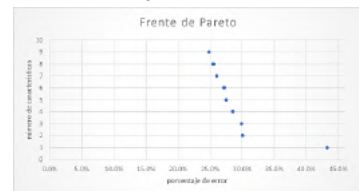
(f) Ejecución 21



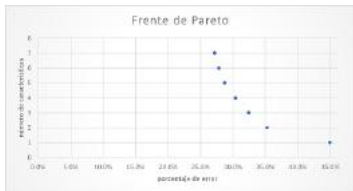
(g) Ejecución 22



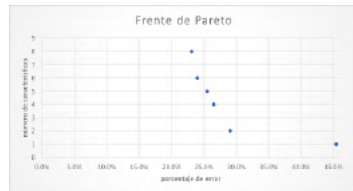
(h) Ejecución 23



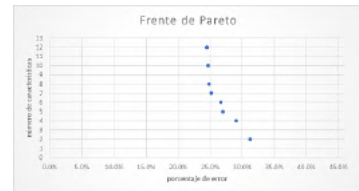
(i) Ejecución 24



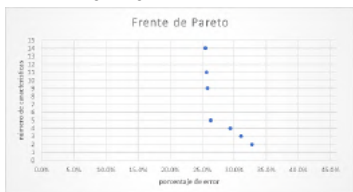
(j) Ejecución 25



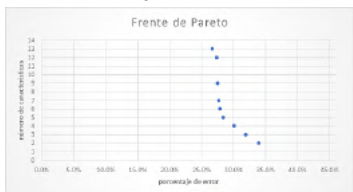
(k) Ejecución 26



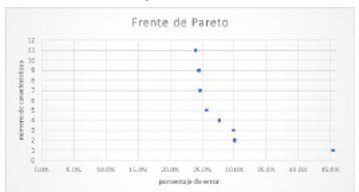
(l) Ejecución 27



(m) Ejecución 28

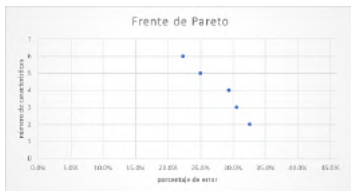


(n) Ejecución 29

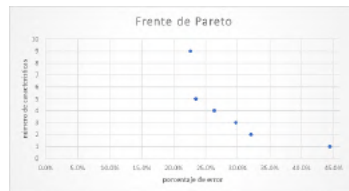


(ñ) Ejecución 30

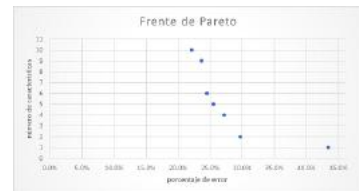
Figura B.30: Frentes de Pareto del Conjunto de datos vehicle con FSGDE3. (2 - 2)



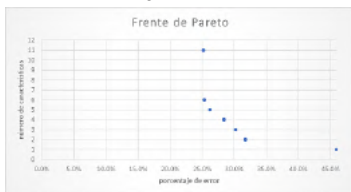
(a) Ejecución 1



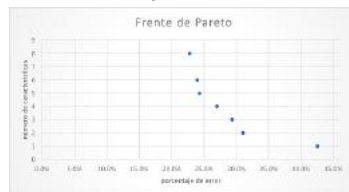
(b) Ejecución 2



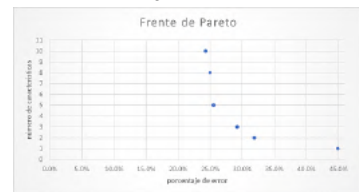
(c) Ejecución 3



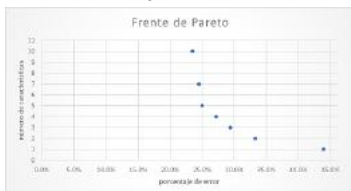
(d) Ejecución 4



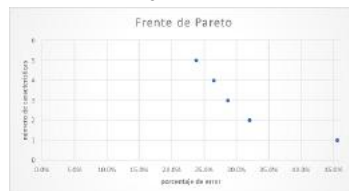
(e) Ejecución 5



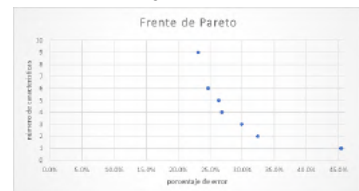
(f) Ejecución 6



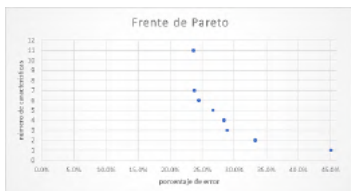
(g) Ejecución 7



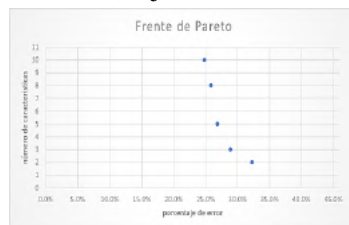
(h) Ejecución 8



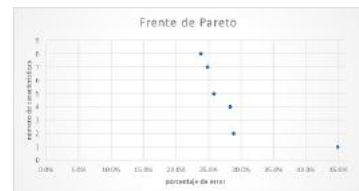
(i) Ejecución 9



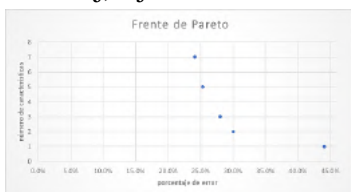
(j) Ejecución 10



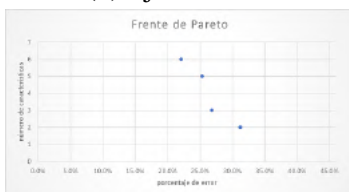
(k) Ejecución 11



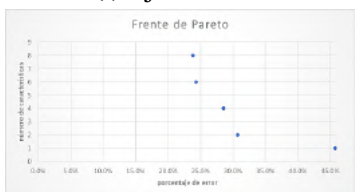
(l) Ejecución 12



(m) Ejecución 13

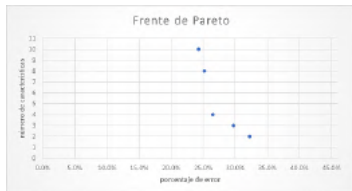


(n) Ejecución 14

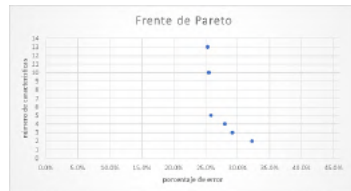


(ñ) Ejecución 15

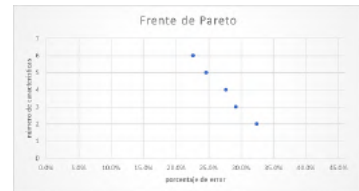
Figura B.31: Frentes de Pareto del Conjunto de datos vehicle con FSNSGA II. (1 - 2)



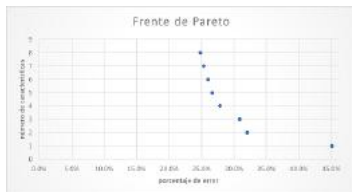
(a) Ejecución 16



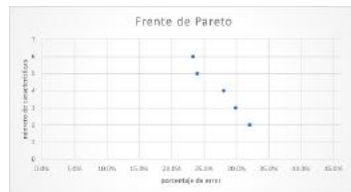
(b) Ejecución 17



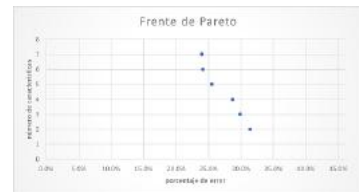
(c) Ejecución 18



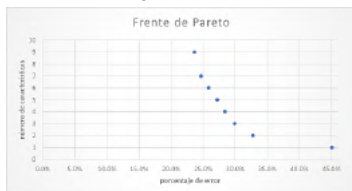
(d) Ejecución 19



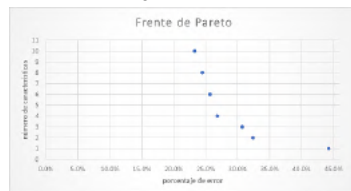
(e) Ejecución 20



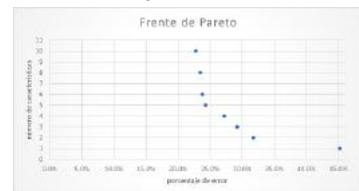
(f) Ejecución 21



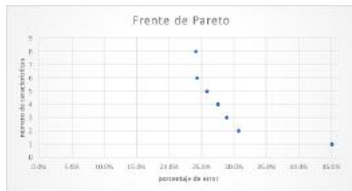
(g) Ejecución 22



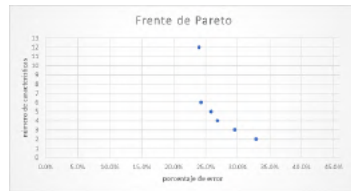
(h) Ejecución 23



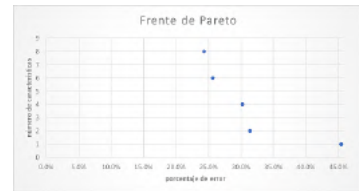
(i) Ejecución 24



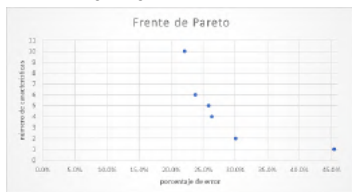
(j) Ejecución 25



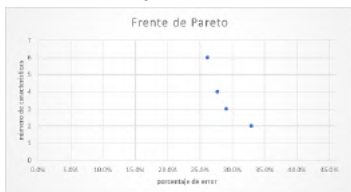
(k) Ejecución 26



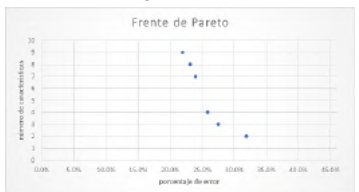
(l) Ejecución 27



(m) Ejecución 28



(n) Ejecución 29



(ñ) Ejecución 30

Figura B.32: Frentes de Pareto del Conjunto de datos vehicle con FSNSGA II. (2 - 2)



(a) Ejecución 1



(b) Ejecución 2



(c) Ejecución 3



(d) Ejecución 4



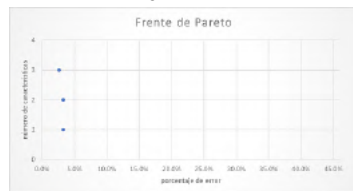
(e) Ejecución 5



(f) Ejecución 6



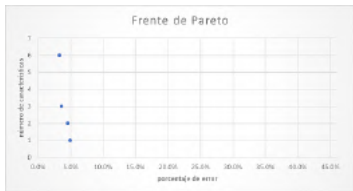
(g) Ejecución 7



(h) Ejecución 8



(i) Ejecución 9



(j) Ejecución 10



(k) Ejecución 11



(l) Ejecución 12



(m) Ejecución 13

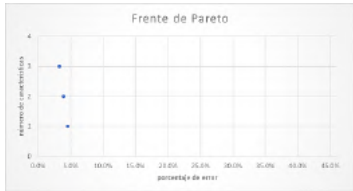


(n) Ejecución 14



(ñ) Ejecución 15

Figura B.33: Frentes de Pareto del Conjunto de datos vote con FSGDE3. (1 - 2)



(a) Ejecución 16



(b) Ejecución 17



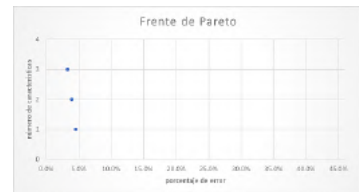
(c) Ejecución 18



(d) Ejecución 19



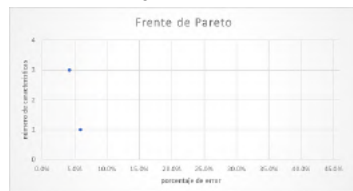
(e) Ejecución 20



(f) Ejecución 21



(g) Ejecución 22



(h) Ejecución 23



(i) Ejecución 24



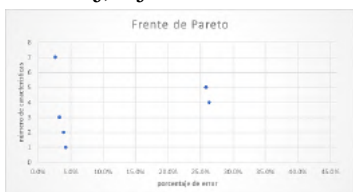
(j) Ejecución 25



(k) Ejecución 26



(l) Ejecución 27



(m) Ejecución 28



(n) Ejecución 29

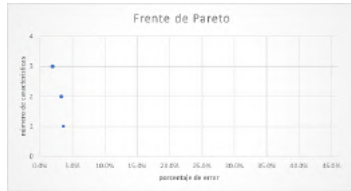


(ñ) Ejecución 30

Figura B.34: Frentes de Pareto del Conjunto de datos vote con FSGDE3. (2 - 2)



(a) Ejecución 1



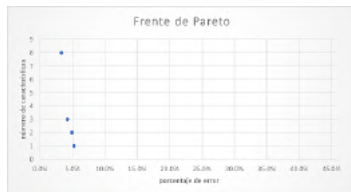
(b) Ejecución 2



(c) Ejecución 3



(d) Ejecución 4



(e) Ejecución 5



(f) Ejecución 6



(g) Ejecución 7



(h) Ejecución 8



(i) Ejecución 9



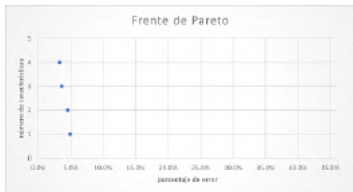
(j) Ejecución 10



(k) Ejecución 11



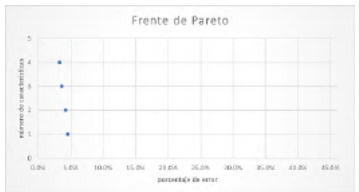
(l) Ejecución 12



(m) Ejecución 13



(n) Ejecución 14



(ñ) Ejecución 15

Figura B.35: Frentes de Pareto del Conjunto de datos vote con FSNSGA II. (1 - 2)



(a) Ejecución 16



(b) Ejecución 17



(c) Ejecución 18



(d) Ejecución 19



(e) Ejecución 20



(f) Ejecución 21



(g) Ejecución 22



(h) Ejecución 23



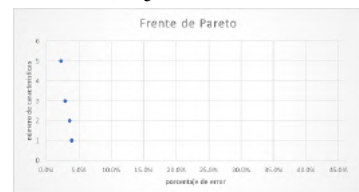
(i) Ejecución 24



(j) Ejecución 25



(k) Ejecución 26



(l) Ejecución 27



(m) Ejecución 28

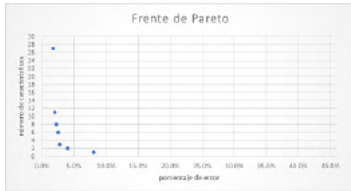


(n) Ejecución 29

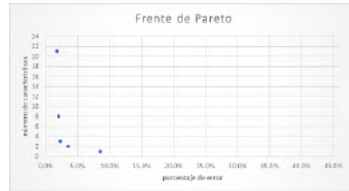


(ñ) Ejecución 30

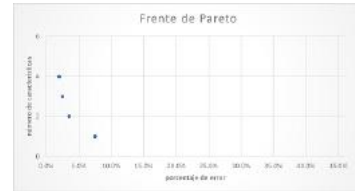
Figura B.36: Frentes de Pareto del Conjunto de datos vote con FSNSGA II. (2 - 2)



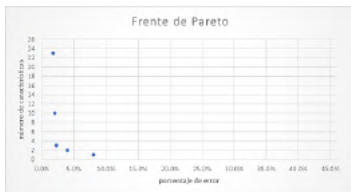
(a) Ejecución 1



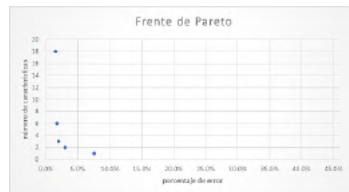
(b) Ejecución 2



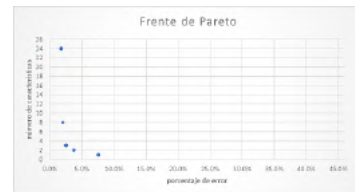
(c) Ejecución 3



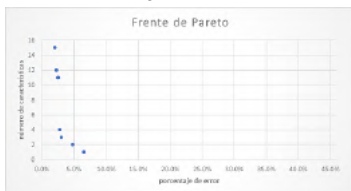
(d) Ejecución 4



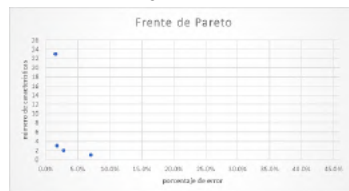
(e) Ejecución 5



(f) Ejecución 6



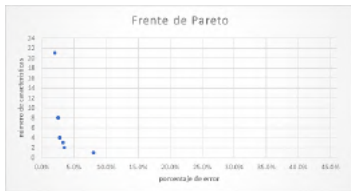
(g) Ejecución 7



(h) Ejecución 8



(i) Ejecución 9



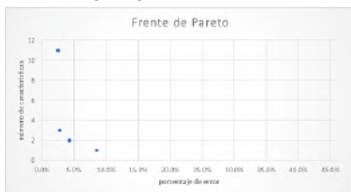
(j) Ejecución 10



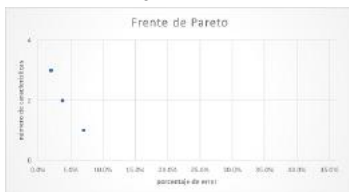
(k) Ejecución 11



(l) Ejecución 12



(m) Ejecución 13

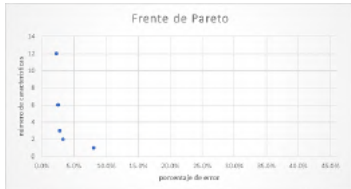


(n) Ejecución 14

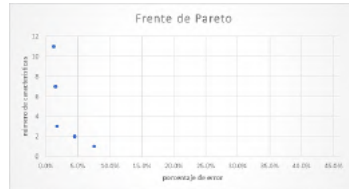


(ñ) Ejecución 15

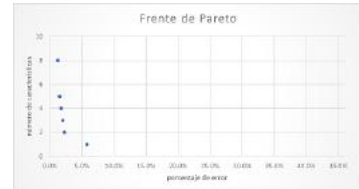
Figura B.37: Frentes de Pareto del Conjunto de datos wdbc con FSGDE3. (1 - 2)



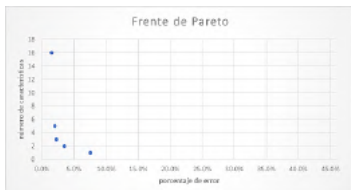
(a) Ejecución 16



(b) Ejecución 17



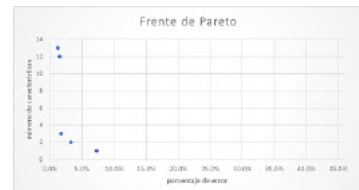
(c) Ejecución 18



(d) Ejecución 19



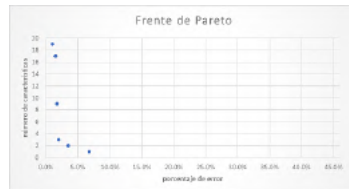
(e) Ejecución 20



(f) Ejecución 21



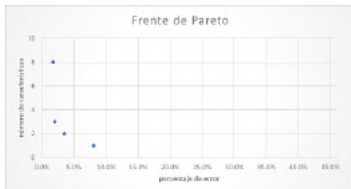
(g) Ejecución 22



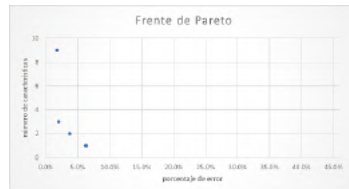
(h) Ejecución 23



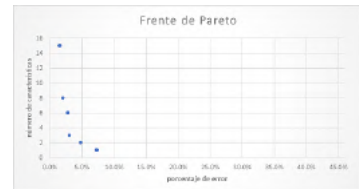
(i) Ejecución 24



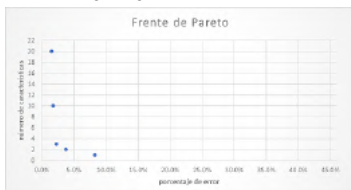
(j) Ejecución 25



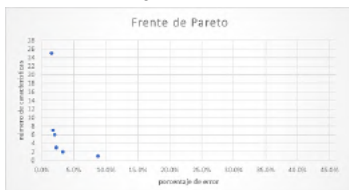
(k) Ejecución 26



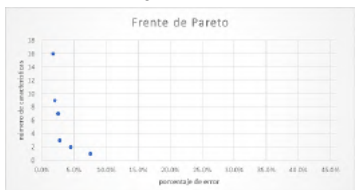
(l) Ejecución 27



(m) Ejecución 28



(n) Ejecución 29



(ñ) Ejecución 30

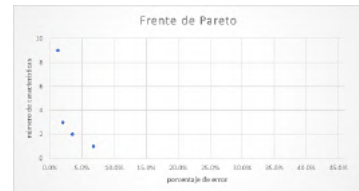
Figura B.38: Frentes de Pareto del Conjunto de datos wdbc con FSGDE3. (2 - 2)



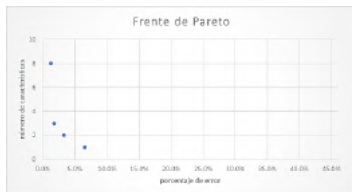
(a) Ejecución 1



(b) Ejecución 2



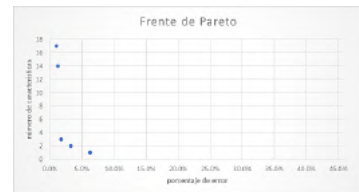
(c) Ejecución 3



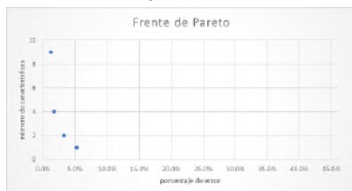
(d) Ejecución 4



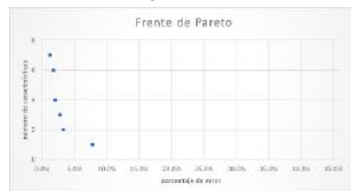
(e) Ejecución 5



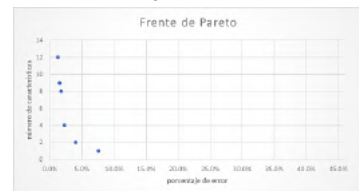
(f) Ejecución 6



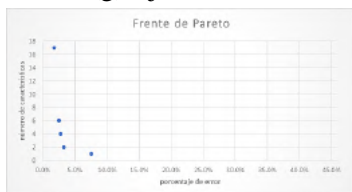
(g) Ejecución 7



(h) Ejecución 8



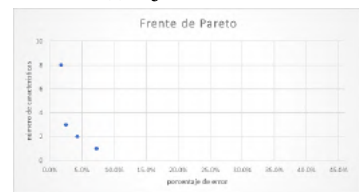
(i) Ejecución 9



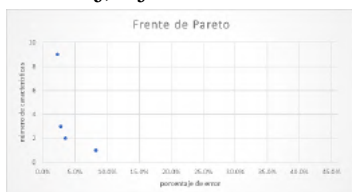
(j) Ejecución 10



(k) Ejecución 11



(l) Ejecución 12



(m) Ejecución 13

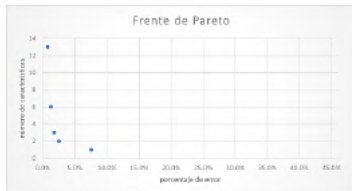


(n) Ejecución 14

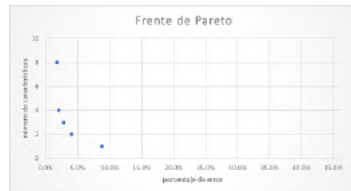


(ñ) Ejecución 15

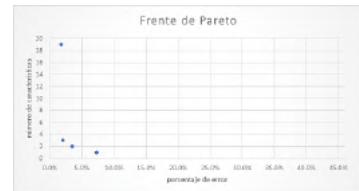
Figura B.39: Frentes de Pareto del Conjunto de datos wdcb con FSNSGA II. (1 - 2)



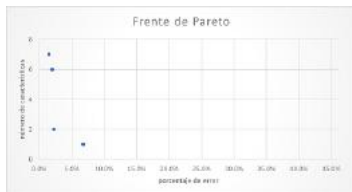
(a) Ejecución 16



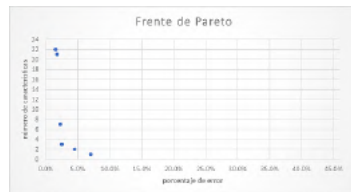
(b) Ejecución 17



(c) Ejecución 18



(d) Ejecución 19



(e) Ejecución 20



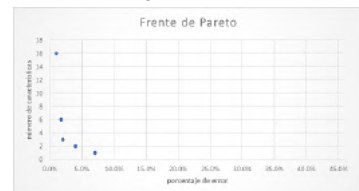
(f) Ejecución 21



(g) Ejecución 22



(h) Ejecución 23



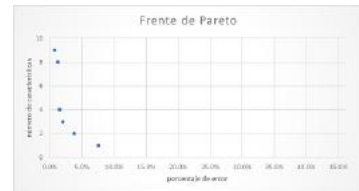
(i) Ejecución 24



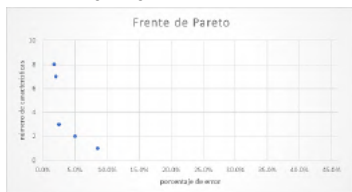
(j) Ejecución 25



(k) Ejecución 26



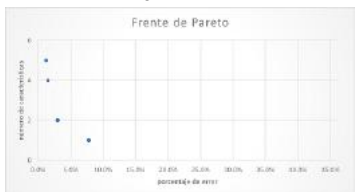
(l) Ejecución 27



(m) Ejecución 28



(n) Ejecución 29



(ñ) Ejecución 30

Figura B.40: Frentes de Pareto del Conjunto de datos wdbc con FSNSGA II. (2 - 2)

Capítulo C

Anexos-microarreglos



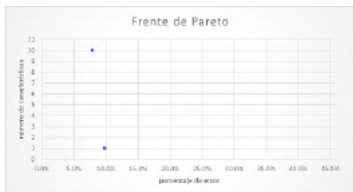
(a) Ejecución 1



(b) Ejecución 2



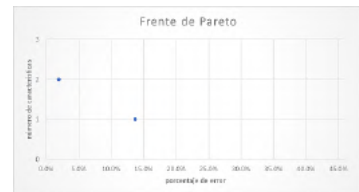
(c) Ejecución 3



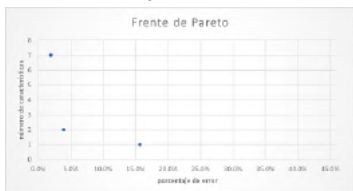
(d) Ejecución 4



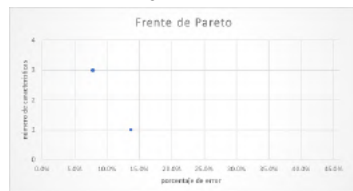
(e) Ejecución 5



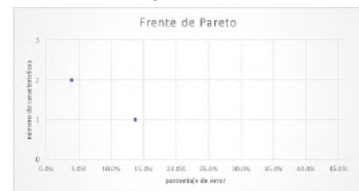
(f) Ejecución 6



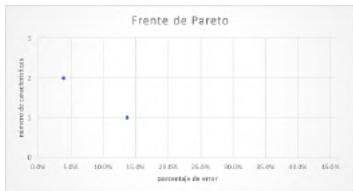
(g) Ejecución 7



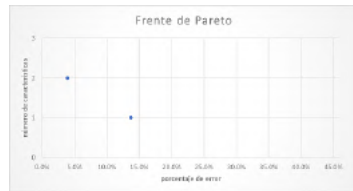
(h) Ejecución 8



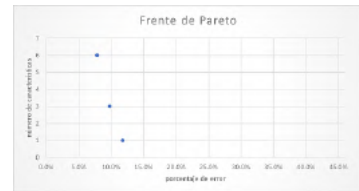
(i) Ejecución 9



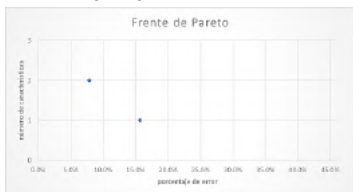
(j) Ejecución 10



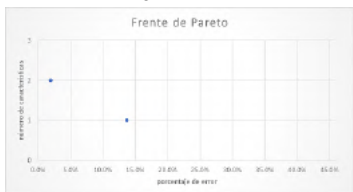
(k) Ejecución 11



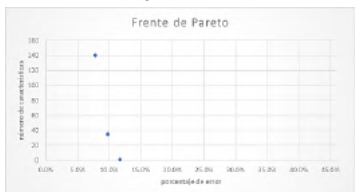
(l) Ejecución 12



(m) Ejecución 13

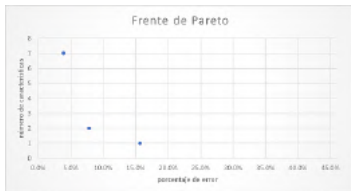


(n) Ejecución 14

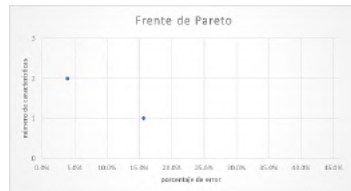


(ñ) Ejecución 15

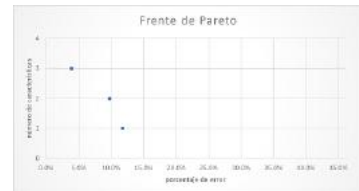
Figura C.1: Frentes de Pareto del Conjunto de datos all-aml con FSGDE3. (1 - 2)



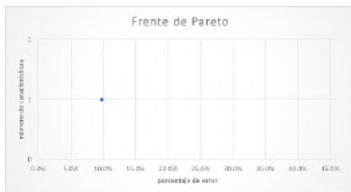
(a) Ejecución 16



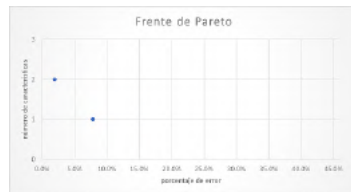
(b) Ejecución 17



(c) Ejecución 18



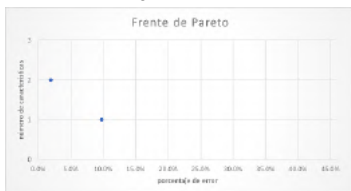
(d) Ejecución 19



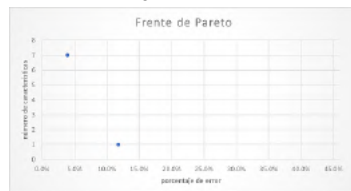
(e) Ejecución 20



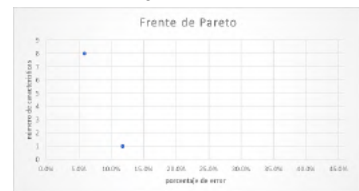
(f) Ejecución 21



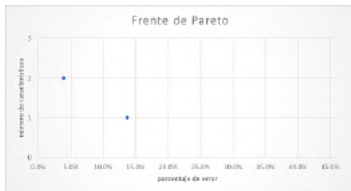
(g) Ejecución 22



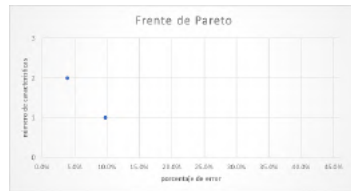
(h) Ejecución 23



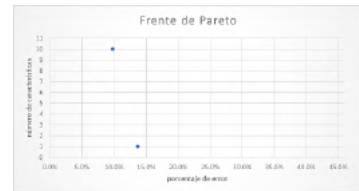
(i) Ejecución 24



(j) Ejecución 25



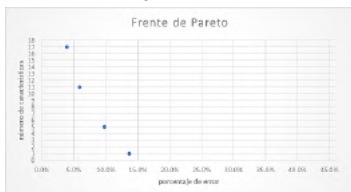
(k) Ejecución 26



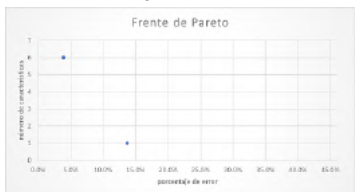
(l) Ejecución 27



(m) Ejecución 28

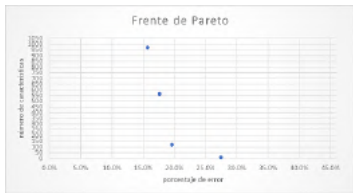


(n) Ejecución 29

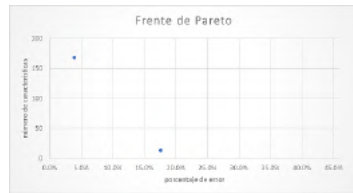


(ñ) Ejecución 30

Figura C.2: Frentes de Pareto del Conjunto de datos all-aml con FSGDE3. (2 - 2)



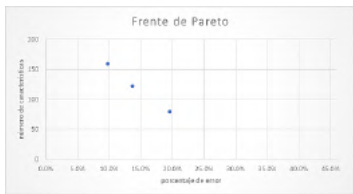
(a) Ejecución 1



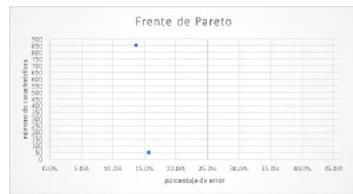
(b) Ejecución 2



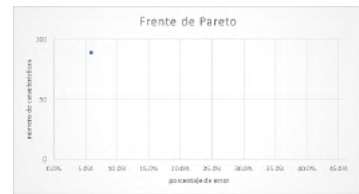
(c) Ejecución 3



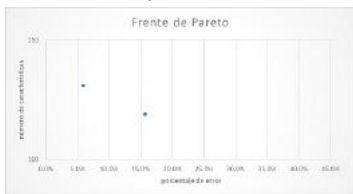
(d) Ejecución 4



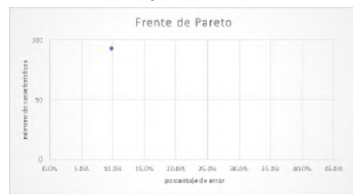
(e) Ejecución 5



(f) Ejecución 6



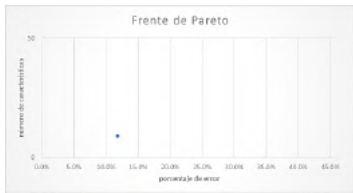
(g) Ejecución 7



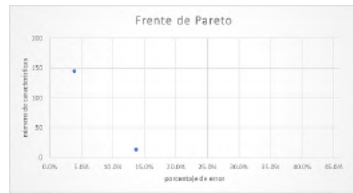
(h) Ejecución 8



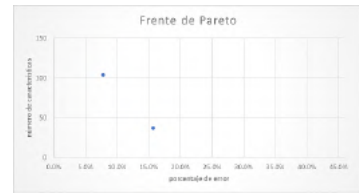
(i) Ejecución 9



(j) Ejecución 10



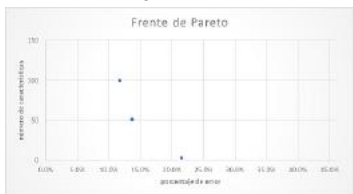
(k) Ejecución 11



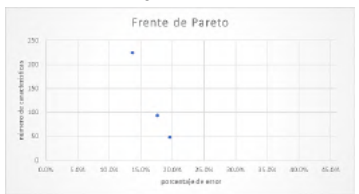
(l) Ejecución 12



(m) Ejecución 13



(n) Ejecución 14

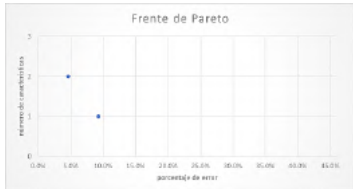


(ñ) Ejecución 15

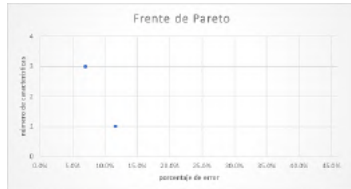
Figura C.3: Frentes de Pareto del Conjunto de datos all-aml con FSNSGA II. (1 - 2)



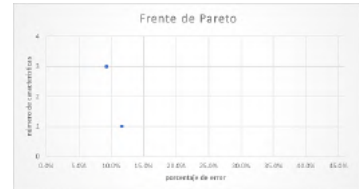
Figura C.4: Frentes de Pareto del Conjunto de datos all-aml con FSNSGA II. (2 - 2)



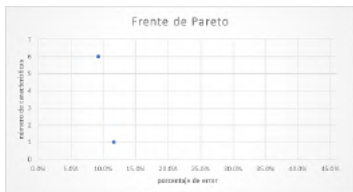
(a) Ejecución 1



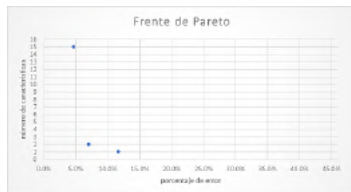
(b) Ejecución 2



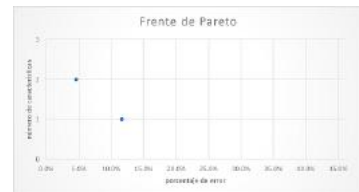
(c) Ejecución 3



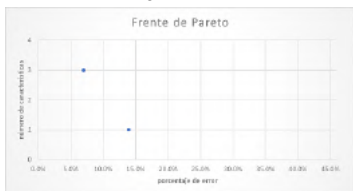
(d) Ejecución 4



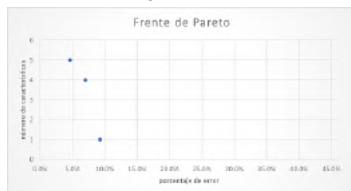
(e) Ejecución 5



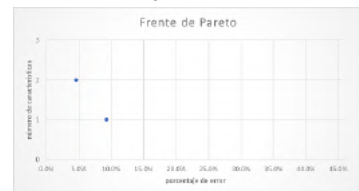
(f) Ejecución 6



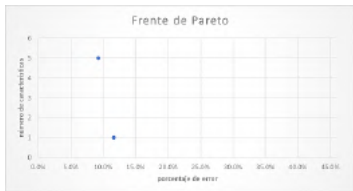
(g) Ejecución 7



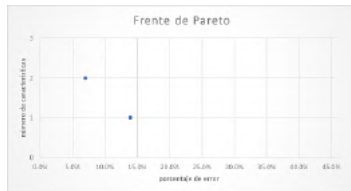
(h) Ejecución 8



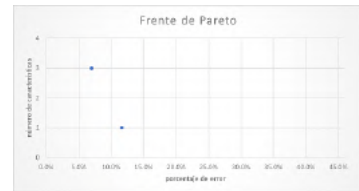
(i) Ejecución 9



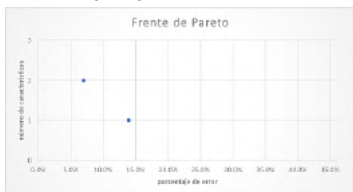
(j) Ejecución 10



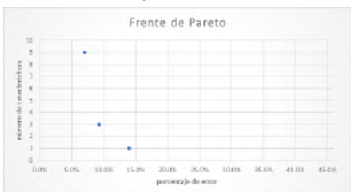
(k) Ejecución 11



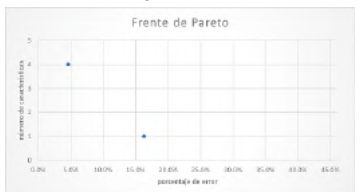
(l) Ejecución 12



(m) Ejecución 13



(n) Ejecución 14

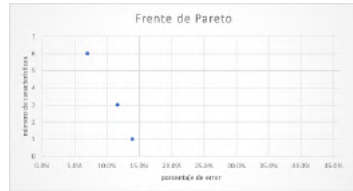


(ñ) Ejecución 15

Figura C.5: Frentes de Pareto del Conjunto de datos colon con FSGDE3. (1 - 2)



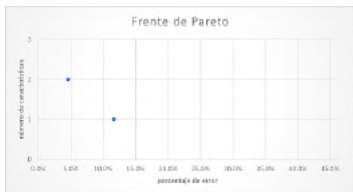
(a) Ejecución 16



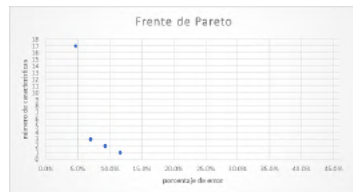
(b) Ejecución 17



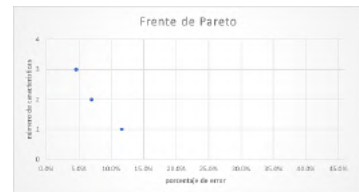
(c) Ejecución 18



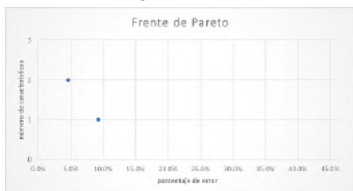
(d) Ejecución 19



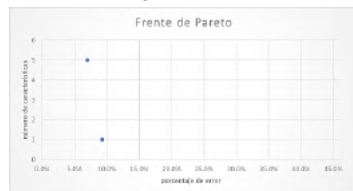
(e) Ejecución 20



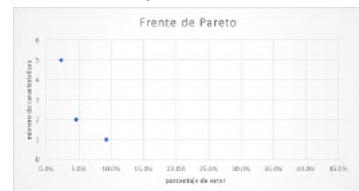
(f) Ejecución 21



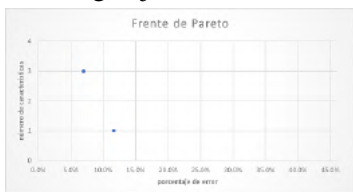
(g) Ejecución 22



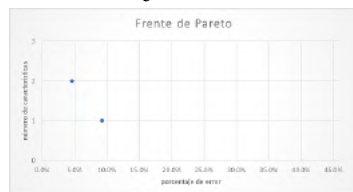
(h) Ejecución 23



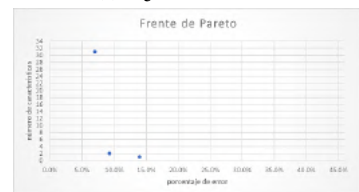
(i) Ejecución 24



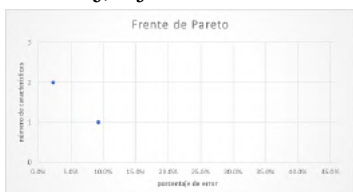
(j) Ejecución 25



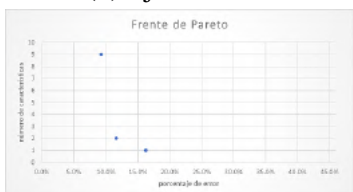
(k) Ejecución 26



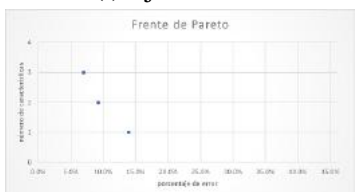
(l) Ejecución 27



(m) Ejecución 28

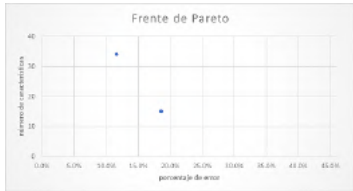


(n) Ejecución 29

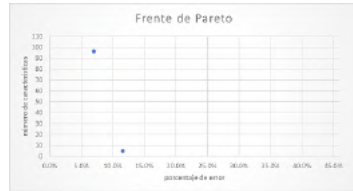


(ñ) Ejecución 30

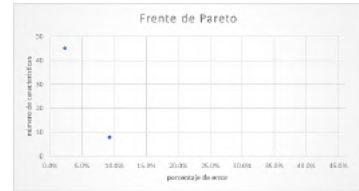
Figura C.6: Frentes de Pareto del Conjunto de datos colon con FSGDE3. (2 - 2)



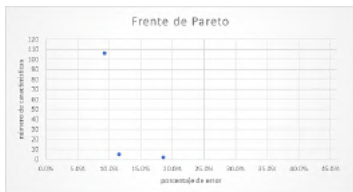
(a) Ejecución 1



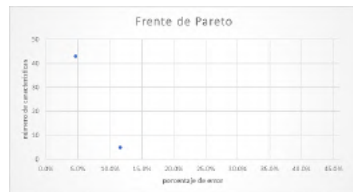
(b) Ejecución 2



(c) Ejecución 3



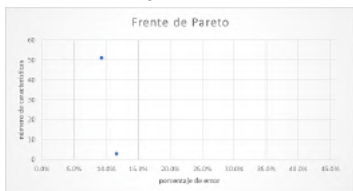
(d) Ejecución 4



(e) Ejecución 5



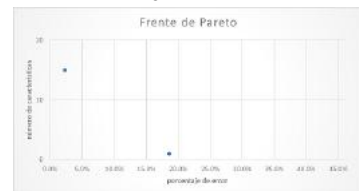
(f) Ejecución 6



(g) Ejecución 7



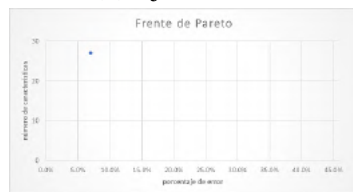
(h) Ejecución 8



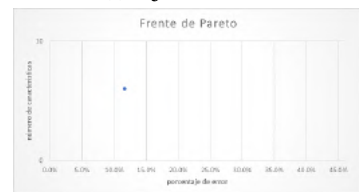
(i) Ejecución 9



(j) Ejecución 10



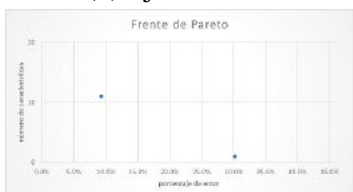
(k) Ejecución 11



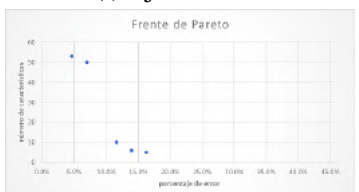
(l) Ejecución 12



(m) Ejecución 13

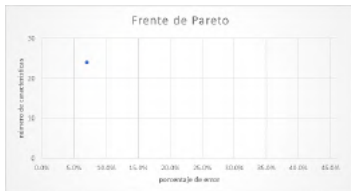


(n) Ejecución 14

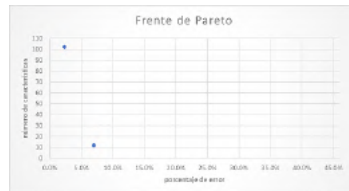


(ñ) Ejecución 15

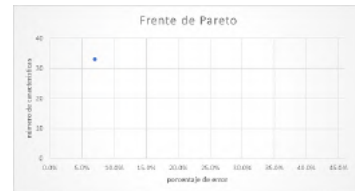
Figura C.7: Frentes de Pareto del Conjunto de datos colon con FSNSGA II. (1 - 2)



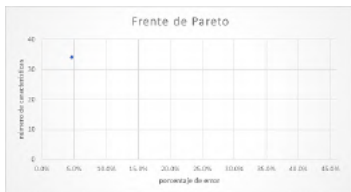
(a) Ejecución 16



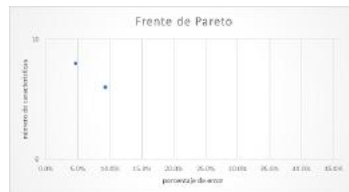
(b) Ejecución 17



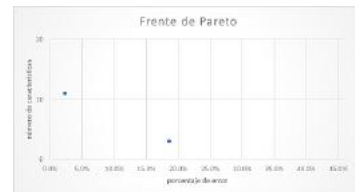
(c) Ejecución 18



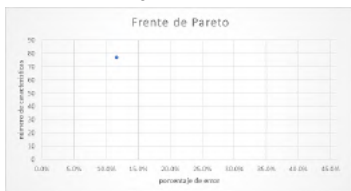
(d) Ejecución 19



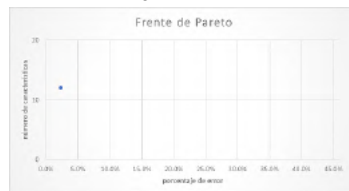
(e) Ejecución 20



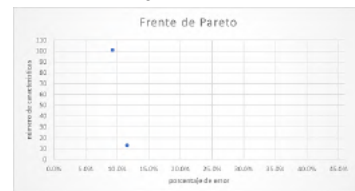
(f) Ejecución 21



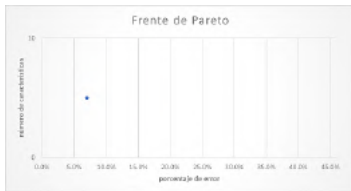
(g) Ejecución 22



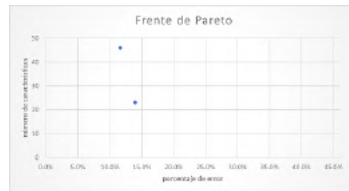
(h) Ejecución 23



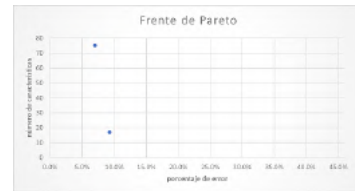
(i) Ejecución 24



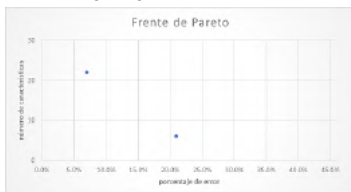
(j) Ejecución 25



(k) Ejecución 26



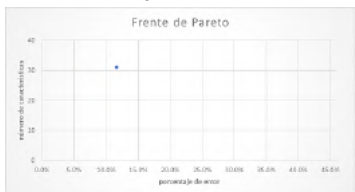
(l) Ejecución 27



(m) Ejecución 28

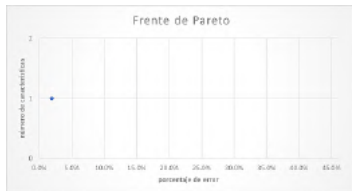


(n) Ejecución 29

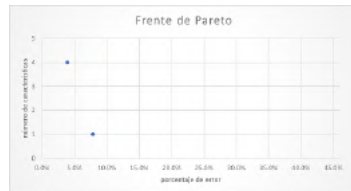


(ñ) Ejecución 30

Figura C.8: Frentes de Pareto del Conjunto de datos colon con FSNSGA II. (2 - 2)



(a) Ejecución 1



(b) Ejecución 2



(c) Ejecución 3



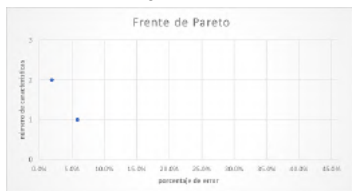
(d) Ejecución 4



(e) Ejecución 5



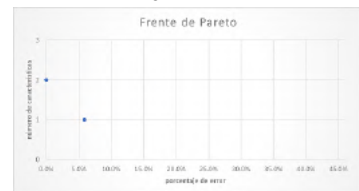
(f) Ejecución 6



(g) Ejecución 7



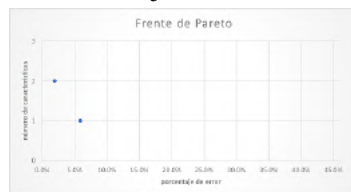
(h) Ejecución 8



(i) Ejecución 9



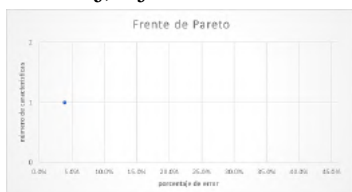
(j) Ejecución 10



(k) Ejecución 11



(l) Ejecución 12



(m) Ejecución 13



(n) Ejecución 14



(ñ) Ejecución 15

Figura C.9: Frentes de Pareto del Conjunto de datos leukemia con FSGDE3. (1 - 2)



(a) Ejecución 16



(b) Ejecución 17



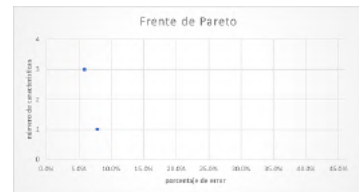
(c) Ejecución 18



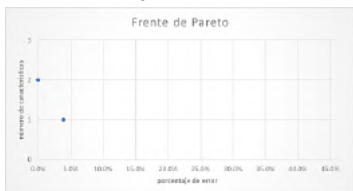
(d) Ejecución 19



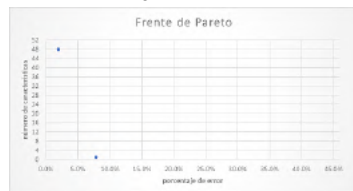
(e) Ejecución 20



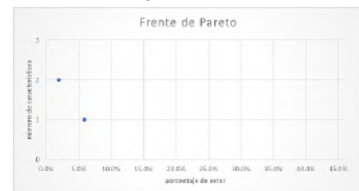
(f) Ejecución 21



(g) Ejecución 22



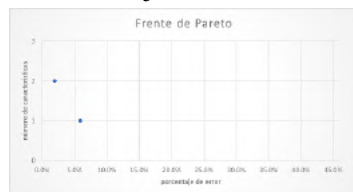
(h) Ejecución 23



(i) Ejecución 24



(j) Ejecución 25



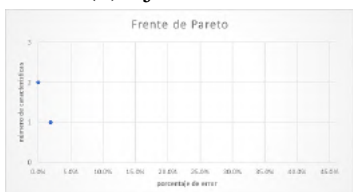
(k) Ejecución 26



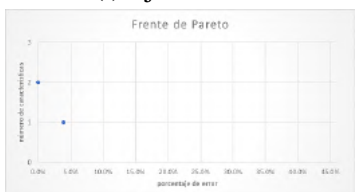
(l) Ejecución 27



(m) Ejecución 28

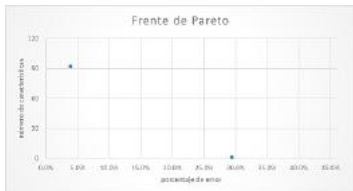


(n) Ejecución 29

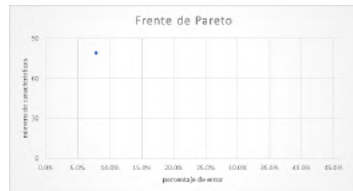


(ñ) Ejecución 30

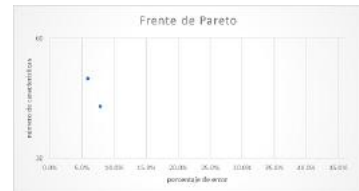
Figura C.10: Frentes de Pareto del Conjunto de datos leukemia con FSGDE3. (2 - 2)



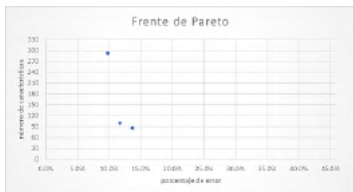
(a) Ejecución 1



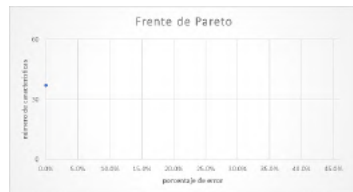
(b) Ejecución 2



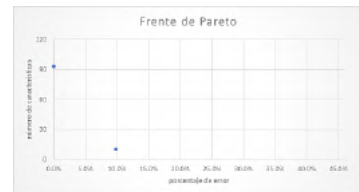
(c) Ejecución 3



(d) Ejecución 4



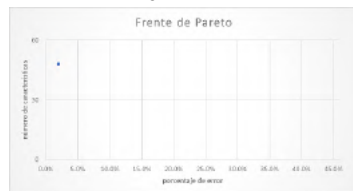
(e) Ejecución 5



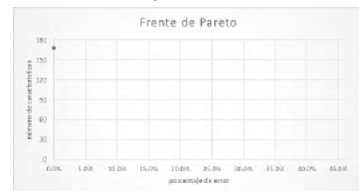
(f) Ejecución 6



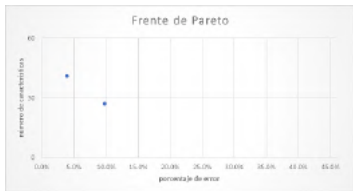
(g) Ejecución 7



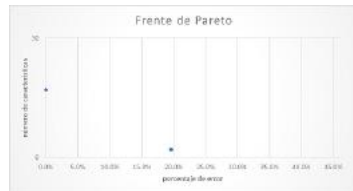
(h) Ejecución 8



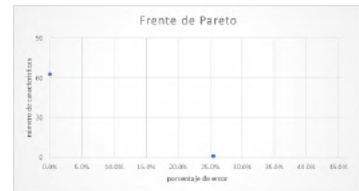
(i) Ejecución 9



(j) Ejecución 10



(k) Ejecución 11



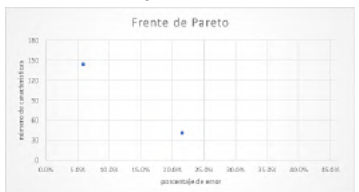
(l) Ejecución 12



(m) Ejecución 13

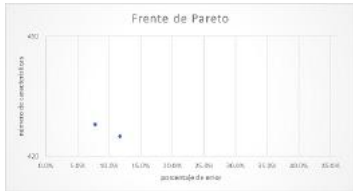


(n) Ejecución 14

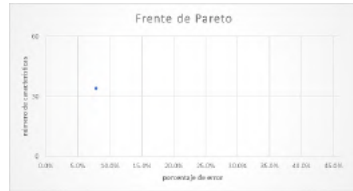


(ñ) Ejecución 15

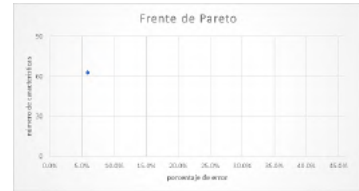
Figura C.11: Frentes de Pareto del Conjunto de datos leukemia con FSNSGA II. (1 - 2)



(a) Ejecución 16



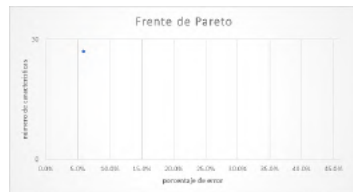
(b) Ejecución 17



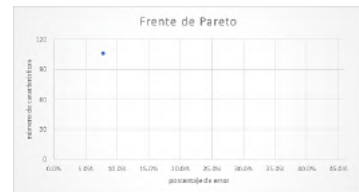
(c) Ejecución 18



(d) Ejecución 19



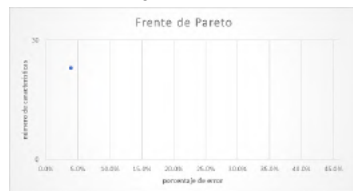
(e) Ejecución 20



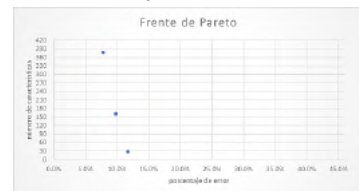
(f) Ejecución 21



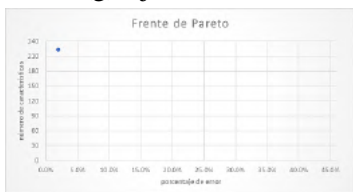
(g) Ejecución 22



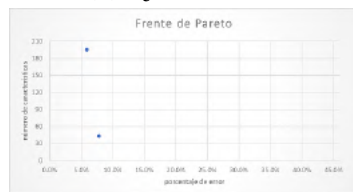
(h) Ejecución 23



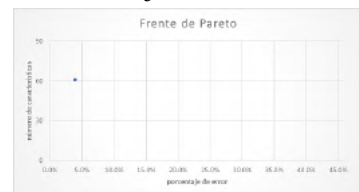
(i) Ejecución 24



(j) Ejecución 25



(k) Ejecución 26



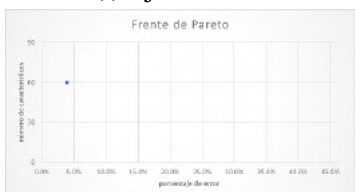
(l) Ejecución 27



(m) Ejecución 28

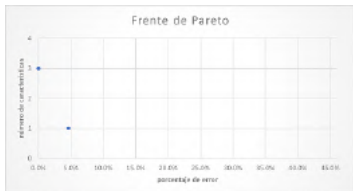


(n) Ejecución 29



(ñ) Ejecución 30

Figura C.12: Frentes de Pareto del Conjunto de datos leukemia con FSNSGA II. (2 - 2)



(a) Ejecución 1



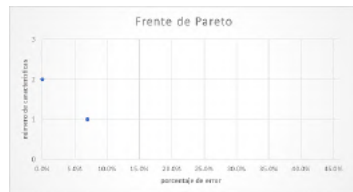
(b) Ejecución 2



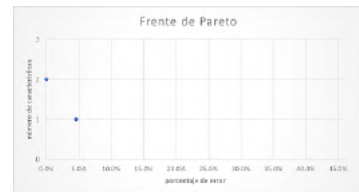
(c) Ejecución 3



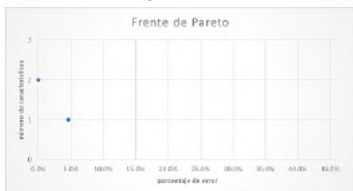
(d) Ejecución 4



(e) Ejecución 5



(f) Ejecución 6



(g) Ejecución 7



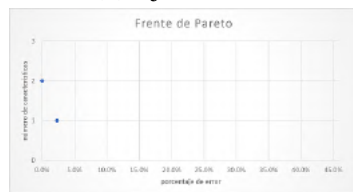
(h) Ejecución 8



(i) Ejecución 9



(j) Ejecución 10



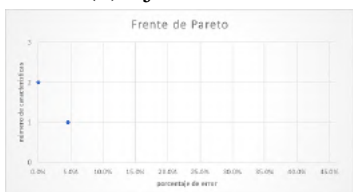
(k) Ejecución 11



(l) Ejecución 12



(m) Ejecución 13

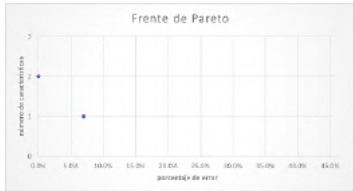


(n) Ejecución 14



(ñ) Ejecución 15

Figura C.13: Frentes de Pareto del Conjunto de datos lymphoma con FSGDE3. (1 - 2)



(a) Ejecución 16



(b) Ejecución 17



(c) Ejecución 18



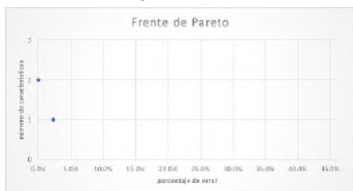
(d) Ejecución 19



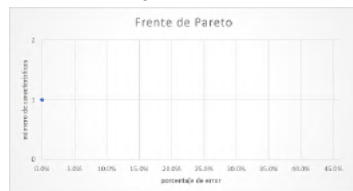
(e) Ejecución 20



(f) Ejecución 21



(g) Ejecución 22



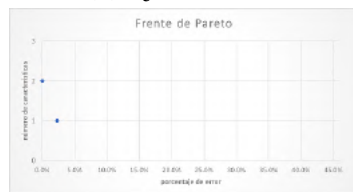
(h) Ejecución 23



(i) Ejecución 24



(j) Ejecución 25



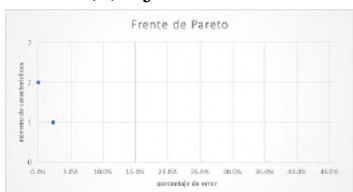
(k) Ejecución 26



(l) Ejecución 27



(m) Ejecución 28

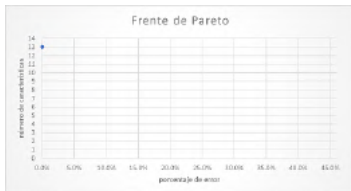


(n) Ejecución 29

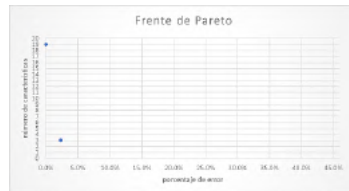


(ñ) Ejecución 30

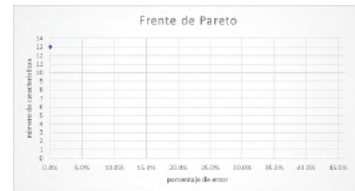
Figura C.14: Frentes de Pareto del Conjunto de datos lymphoma con FSGDE3. (2 - 2)



(a) Ejecución 1



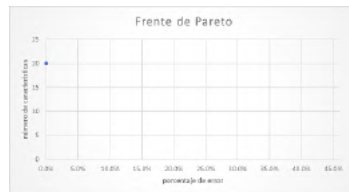
(b) Ejecución 2



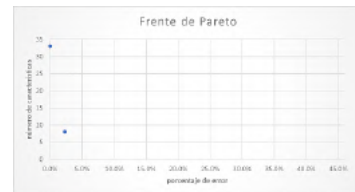
(c) Ejecución 3



(d) Ejecución 4



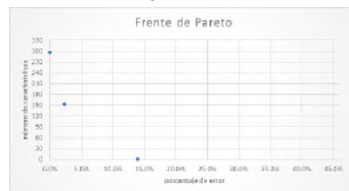
(e) Ejecución 5



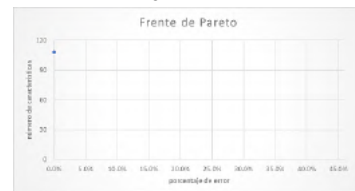
(f) Ejecución 6



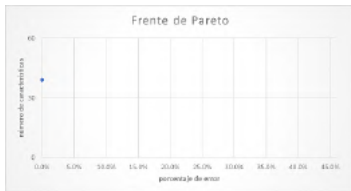
(g) Ejecución 7



(h) Ejecución 8



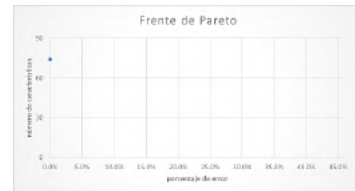
(i) Ejecución 9



(j) Ejecución 10



(k) Ejecución 11



(l) Ejecución 12



(m) Ejecución 13



(n) Ejecución 14



(ñ) Ejecución 15

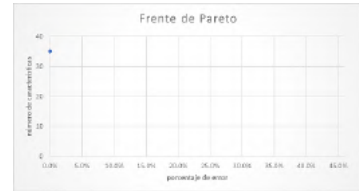
Figura C.15: Frentes de Pareto del Conjunto de datos lymphoma con FSNSGA II. (1 - 2)



(a) Ejecución 16



(b) Ejecución 17



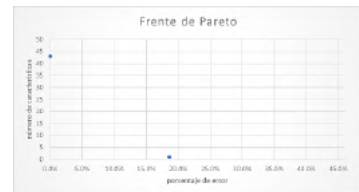
(c) Ejecución 18



(d) Ejecución 19



(e) Ejecución 20



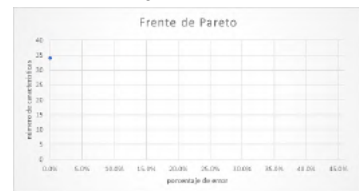
(f) Ejecución 21



(g) Ejecución 22



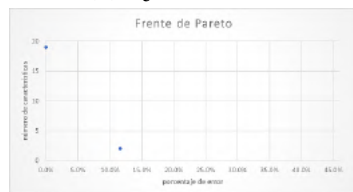
(h) Ejecución 23



(i) Ejecución 24



(j) Ejecución 25



(k) Ejecución 26



(l) Ejecución 27



(m) Ejecución 28

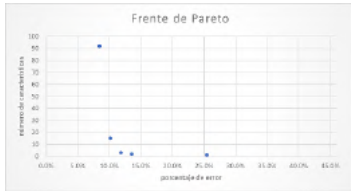


(n) Ejecución 29

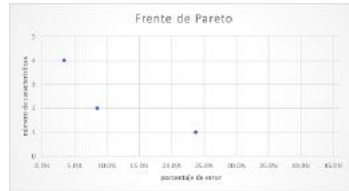


(ñ) Ejecución 30

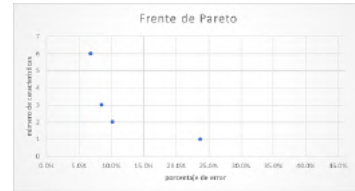
Figura C.16: Frentes de Pareto del Conjunto de datos lymphoma con FSNSGA II. (2 - 2)



(a) Ejecución 1



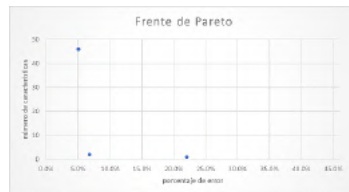
(b) Ejecución 2



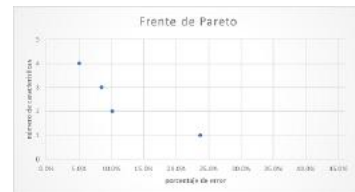
(c) Ejecución 3



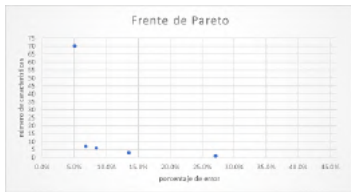
(d) Ejecución 4



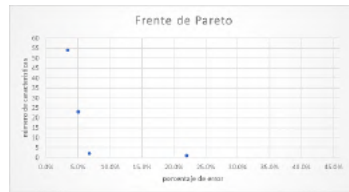
(e) Ejecución 5



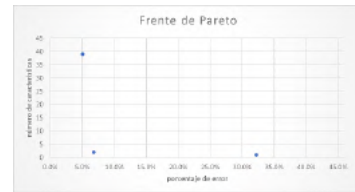
(f) Ejecución 6



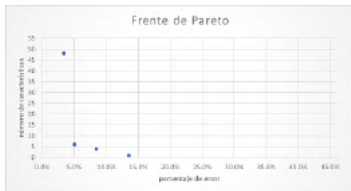
(g) Ejecución 7



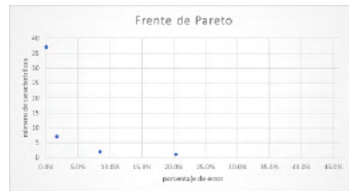
(h) Ejecución 8



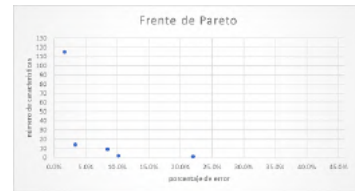
(i) Ejecución 9



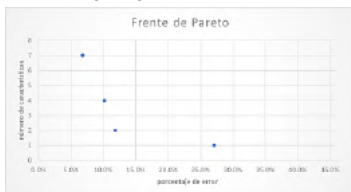
(j) Ejecución 10



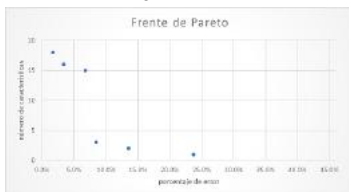
(k) Ejecución 11



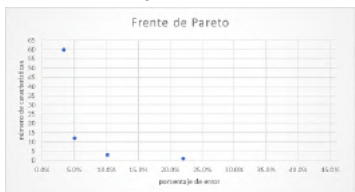
(l) Ejecución 12



(m) Ejecución 13

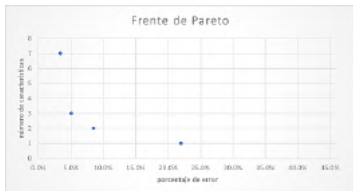


(n) Ejecución 14

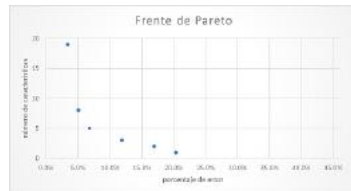


(ñ) Ejecución 15

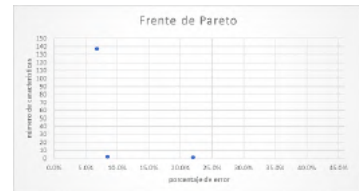
Figura C.17: Frentes de Pareto del Conjunto de datos srbcct con FSGDE3. (1 - 2)



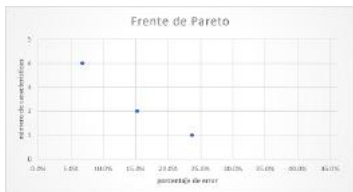
(a) Ejecución 16



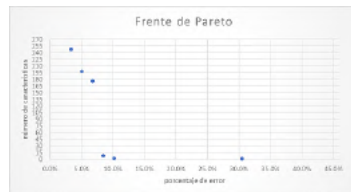
(b) Ejecución 17



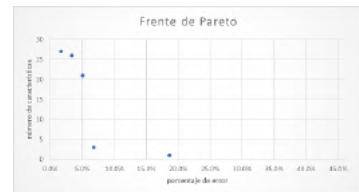
(c) Ejecución 18



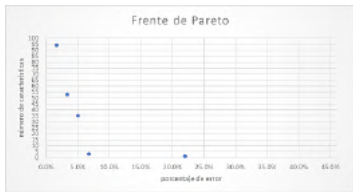
(d) Ejecución 19



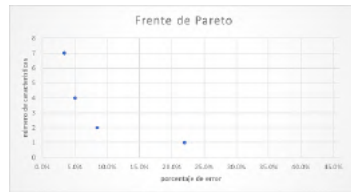
(e) Ejecución 20



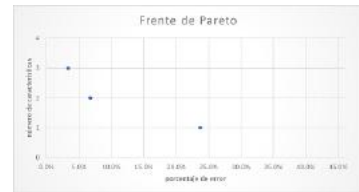
(f) Ejecución 21



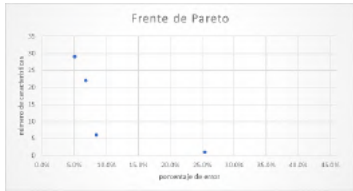
(g) Ejecución 22



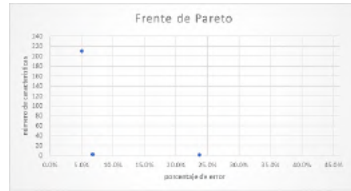
(h) Ejecución 23



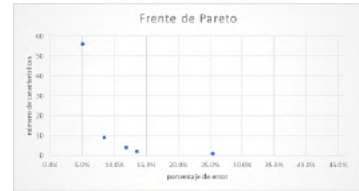
(i) Ejecución 24



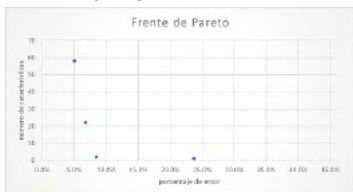
(j) Ejecución 25



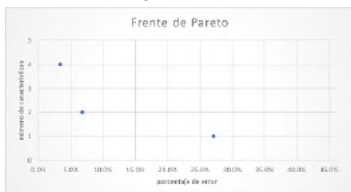
(k) Ejecución 26



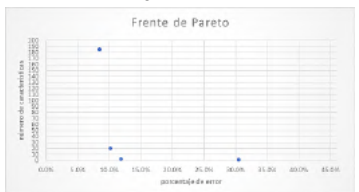
(l) Ejecución 27



(m) Ejecución 28

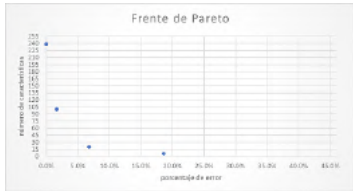


(n) Ejecución 29



(ñ) Ejecución 30

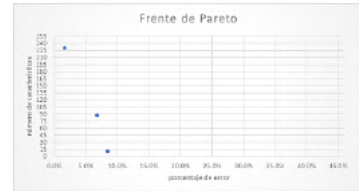
Figura C.18: Frentes de Pareto del Conjunto de datos srbcct con FSGDE3. (2 - 2)



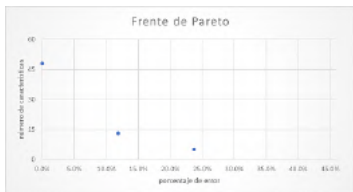
(a) Ejecución 1



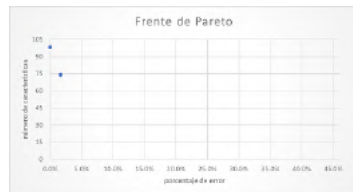
(b) Ejecución 2



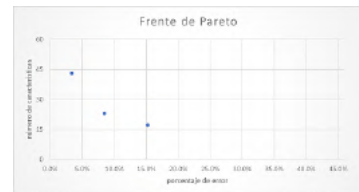
(c) Ejecución 3



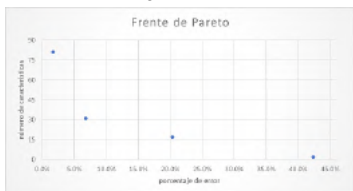
(d) Ejecución 4



(e) Ejecución 5



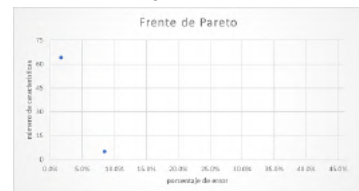
(f) Ejecución 6



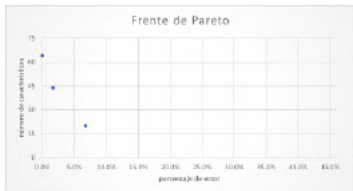
(g) Ejecución 7



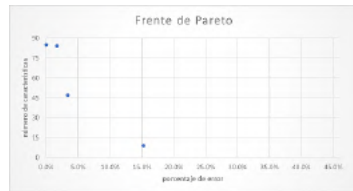
(h) Ejecución 8



(i) Ejecución 9

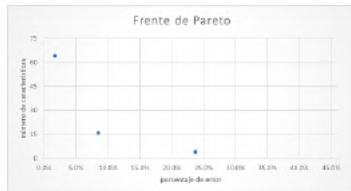


(j) Ejecución 10

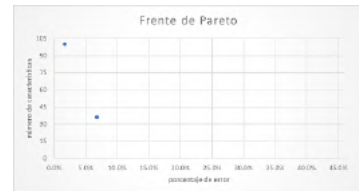




(a) Ejecución 16



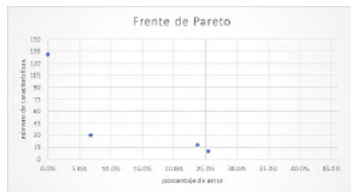
(b) Ejecución 17



(c) Ejecución 18



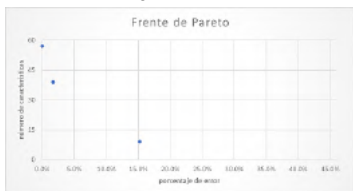
(d) Ejecución 19



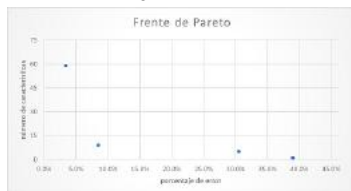
(e) Ejecución 20



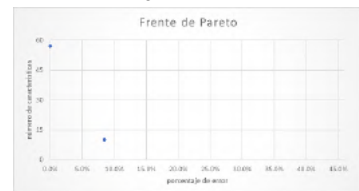
(f) Ejecución 21



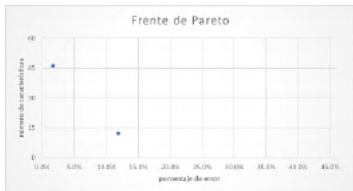
(g) Ejecución 22



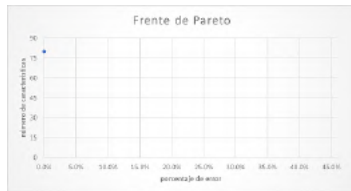
(h) Ejecución 23



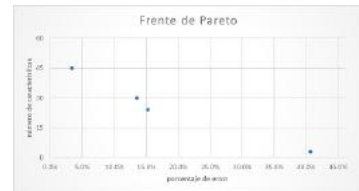
(i) Ejecución 24



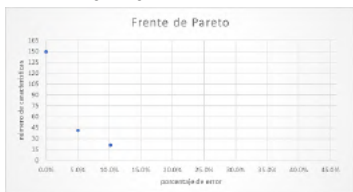
(j) Ejecución 25



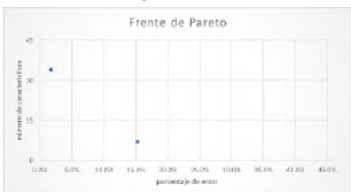
(k) Ejecución 26



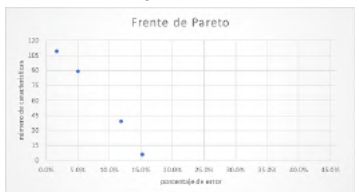
(l) Ejecución 27



(m) Ejecución 28



(n) Ejecución 29



(ñ) Ejecución 30

Figura C.20: Frentes de Pareto del Conjunto de datos srbc con FSNSGA II. (2 - 2)

Bibliografía

- [1] T. O. Ayodele. Types of machine learning algorithms. *New advances in machine learning*, 3:19–48, 2010.
- [2] B. Babu and M. M. L. Jehan. Differential evolution for multi-objective optimization. In *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.*, volume 4, pages 2696–2703. IEEE, 2003.
- [3] M. Birattari, T. Stützle, L. Paquete, K. Varrentrapp, et al. A racing algorithm for configuring metaheuristics. In *Gecco*, volume 2, 2002.
- [4] E. Blanco and H. Sanz. Algoritmos de clustering y aprendizaje automático aplicados a twitter. *Universidad Politécnica de Catalunya*, 56, 2016.
- [5] G. Chandrashekar and F. Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.
- [6] C. A. coello. Introducción a la optimización multiobjetivo, 2021.
- [7] C. A. C. Coello and N. C. Cortés. Solving multiobjective optimization problems using an artificial immune system. *Genetic programming and evolvable machines*, 6(2):163–190, 2005.
- [8] C. A. C. Coello, G. B. Lamont, D. A. Van Veldhuizen, et al. *Evolutionary algorithms for solving multi-objective problems*, volume 5. Springer, 2007.
- [9] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.
- [10] D. Dua and C. Graff. UCI machine learning repository, 2017.
- [11] J. J. Durillo and A. J. Nebro. jmetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, 42(10):760–771, 2011.
- [12] A. E. Eiben and S. K. Smit. Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm and Evolutionary Computation*, 1(1):19–31, 2011.
- [13] A. E. Eiben and J. E. Smith. *Introduction to evolutionary computing*. Springer, 2015.

- [14] C. Feng, C. Qian, and K. Tang. Unsupervised feature selection by pareto optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3534–3541, 2019.
- [15] L. J. Fogel. *Intelligence through simulated evolution: forty years of evolutionary programming*. John Wiley & Sons, Inc., 1999.
- [16] D. E. Goldberg, J. Richardson, et al. Genetic algorithms with sharing for multimodal function optimization. In *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*, pages 41–49. Hillsdale, NJ: Lawrence Erlbaum, 1987.
- [17] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.
- [18] E. Hancer, B. Xue, and M. Zhang. Differential evolution for filter feature selection based on information theory and feature ranking. *Knowledge-Based Systems*, 140:103–119, 2018.
- [19] M. P. Hansen and A. Jaszkievicz. *Evaluating the quality of approximations to the non-dominated set*. Citeseer, 1994.
- [20] G. R. Harik. Finding multimodal solutions using restricted tournament selection. In *ICGA*, pages 24–31, 1995.
- [21] C. Hernández, J. E. R. Rodríguez, et al. Preprocesamiento de datos estructurados. *Revista vínculos*, 4(2):27–48, 2008.
- [22] F. Herrera and J. Cano. Técnicas de reducción de datos en kdd. el uso de algoritmos evolutivos para la selección de instancias. *Actas del I Seminario sobre Sistemas Inteligentes (SSI_06)*, pages 165–181, 2006.
- [23] K. Kira, L. A. Rendell, et al. The feature selection problem: Traditional methods and a new algorithm. In *Aaai*, pages 129–134, 1992.
- [24] S. Kukkonen and J. Lampinen. Comparison of generalized differential evolution algorithm to other multi-objective evolutionary algorithms. In *Proceedings of the 4th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS2004)*, page 445, 2004.
- [25] S. W. Mahfoud. Crowding and preselection revisited. In *PPSN*, volume 2, pages 27–36. Citeseer, 1992.
- [26] O. Maimon and L. Rokach. *Data mining and knowledge discovery handbook*. *Computer science*, 2005.
- [27] S. Maldonado and R. Weber. Modelos de selección de atributos para support vector machines. *Revista Ingeniería de sistemas*, 26:49–70, 2012.

- [28] A. Marcano-Cedeño, J. Quintanilla-Domínguez, M. Cortina-Januchs, and D. Andina. Feature selection using sequential forward selection and classification applying artificial metaplasticity neural network. In *IECON 2010-36th annual conference on IEEE industrial electronics society*, pages 2845–2850. IEEE, 2010.
- [29] T. Marill and D. Green. On the effectiveness of receptors in recognition systems. *IEEE transactions on Information Theory*, 9(1):11–17, 1963.
- [30] O. J. Mengshoel and D. E. Goldberg. The crowding approach to niching in genetic algorithms. *Evolutionary computation*, 16(3):315–354, 2008.
- [31] A. Moreno. *Aprendizaje automático*, 1994.
- [32] H. B. Nguyen, B. Xue, and P. Andreae. Pso with surrogate models for feature selection: static and dynamic clustering-based methods. *Memetic Computing*, 10(3):291–300, 2018.
- [33] A. Osyczka. Multicriteria optimization for engineering design. In *Design optimization*, pages 193–227. Elsevier, 1985.
- [34] A. Pétrowski. A clearing procedure as a niching method for genetic algorithms. In *Proceedings of IEEE international conference on evolutionary computation*, pages 798–803. IEEE, 1996.
- [35] K. Price, R. M. Storn, and J. A. Lampinen. *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media, 2006.
- [36] R. Rivera-López, E. Mezura-Montes, J. Canul-Reich, and M. A. Cruz-Chávez. A permutational-based differential evolution algorithm for feature subset selection. *Pattern Recognition Letters*, 133:86–93, 2020.
- [37] R. S. Rosenberg. Stimulation of genetic populations with biochemical properties: I. the model. *Mathematical Biosciences*, 7(3-4):223–257, 1970.
- [38] G. Rudolph and A. Agapie. Convergence properties of some multi-objective evolutionary algorithms. In *Proceedings of the 2000 congress on evolutionary computation. CEC00 (Cat. No. 00TH8512)*, volume 2, pages 1010–1016. IEEE, 2000.
- [39] Y. Saeys, I. Inza, and P. Larrañaga. A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19):2507–2517, 2007.
- [40] B. Sahu and D. Mishra. A novel feature selection algorithm using particle swarm optimization for cancer microarray data. *Procedia Engineering*, 38:27–31, 2012.
- [41] N. Sánchez Anzola. Máquinas de soporte vectorial y redes neuronales artificiales en la predicción del movimiento usd/cop spot intradiario. *ODEON-Observatorio de Economía y Operaciones Numéricas*, 9:113–172, 2015.

- [42] J. D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the first international conference on genetic algorithms and their applications, 1985*. Lawrence Erlbaum Associates. Inc., Publishers, 1985.
- [43] R. Storn and K. Price. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.
- [44] L. E. Sucar and M. Tonantzintla. Redes bayesianas. *Aprendizaje Automático: conceptos básicos y avanzados*, pages 77–100, 2006.
- [45] C. Tang, X. Liu, X. Zhu, J. Xiong, M. Li, J. Xia, X. Wang, and L. Wang. Feature selective projection with low-rank embedding and dual laplacian regularization. *IEEE Transactions on Knowledge and Data Engineering*, 2019.
- [46] B. Tran, B. Xue, and M. Zhang. Bare-bone particle swarm optimisation for simultaneously discretising and selecting features for high-dimensional classification. In *European Conference on the Applications of Evolutionary Computation*, pages 701–718. Springer, 2016.
- [47] B. Tran, B. Xue, and M. Zhang. A new representation in pso for discretization-based feature selection. *IEEE Transactions on Cybernetics*, 48(6):1733–1746, 2017.
- [48] B. Tran, B. Xue, and M. Zhang. Using feature clustering for gp-based feature construction on high-dimensional data. In *European Conference on Genetic Programming*, pages 210–226. Springer, 2017.
- [49] B. Tran, M. Zhang, and B. Xue. A pso based hybrid feature selection algorithm for high-dimensional classification. In *2016 IEEE congress on evolutionary computation (CEC)*, pages 3801–3808. IEEE, 2016.
- [50] C.-J. Tu, L.-Y. Chuang, J.-Y. Chang, C.-H. Yang, et al. Feature selection using pso-svm. *International Journal of Computer Science*, 2007.
- [51] A. W. Whitney. A direct method of nonparametric measurement selection. *IEEE Transactions on Computers*, 100(9):1100–1103, 1971.
- [52] F. Wilcoxon. Individual comparisons by ranking methods. In *Breakthroughs in statistics*, pages 196–202. Springer, 1992.
- [53] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.
- [54] B. Xue and M. Zhang. Evolutionary computation for feature selection and feature construction. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, pages 1283–1312, 2020.
- [55] B. Xue, M. Zhang, and W. N. Browne. Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE transactions on cybernetics*, 43(6):1656–1671, 2012.

- [56] B. Xue, M. Zhang, W. N. Browne, and X. Yao. A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, 20(4):606–626, 2015.
- [57] Z. Zhu, Y.-S. Ong, and M. Dash. Markov blanket-embedded genetic algorithm for gene selection. *Pattern Recognition*, 40(11):3236–3248, 2007.
- [58] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on evolutionary computation*, 7(2):117–132, 2003.