

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE



“SELECCIÓN DE CARACTERÍSTICAS: UNA PROPUESTA DE NSGA-II CON NUEVAS ESTRATEGIAS DE INICIALIZACIÓN”

NICOLÁS PATRICIO PAZ TRALMA

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Elizabeth Montero
Profesor Correferente: María Cristina Riff

Julio - 2025



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título; ☐ Tesis de Postgrado;

Título del trabajo: SELECCIÓN DE CARACTERÍSTICAS: UNA PROPUESTA DE NSGA-II CON NUEVAS ESTRATEGIAS DE INICIALIZACIÓN

Nombre del candidato(a): Nicolás Patricio Paz Tralma

Carrera / Grado: Ingeniería Civil Informática

Campus: Casa Central Valparaíso ; **Departamento:** Informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Elizabeth Montero, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

☒ El trabajo **NO contiene información que amerite confidencialidad** y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por:

☐ 6 meses; ☐ 12 meses; ☐ 2 años; ☐ 3 años; ☐ 5 años; ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 29/07/2025

Firma:

Estudiante o Candidato(a):

Fecha: 29/07/2025 ; **Firma:**

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

DEDICATORIA

Este trabajo se lo dedico a mi familia y amigos. Gracias al apoyo de todos es que soy capaz de titularme en el tiempo correspondiente. Todos me apoyaron en los buenos y malos momentos y quiero que cada uno pueda apreciar que el esfuerzo de mi trabajo valió la pena. Le agradezco enormemente a cada uno de ellos, pero principalmente a mi abuela, ya que sin ella yo no estaría aquí.

RESUMEN

Resumen— Esta memoria aborda el problema de selección de características como una tarea de optimización multi-objetivo, motivada por el aumento en el volumen de características de los conjuntos de datos utilizados en aplicaciones modernas de aprendizaje automático. El objetivo es identificar un subconjunto reducido de características que mejore la precisión en la clasificación. Se propone el uso de la prueba estadística chi-cuadrado para inicializar el algoritmo evolutivo NSGA-II, aprovechando su naturaleza basada en población para encontrar un conjunto de soluciones no dominadas y de alta calidad. Se proponen y evalúan dos estrategias de inicialización. Se probaron seis conjuntos de datos de distintas áreas, con un rango de entre 16 y 12,600 características. Se observaron mejoras en la precisión y la generalización, especialmente al inicializar con un 10 % de las características. Además, las estrategias propuestas no aumentaron significativamente el tiempo de cómputo y obtuvieron resultados competitivos frente al estado del arte en términos de hipervolumen.

Palabras Clave— Selección de características; NSGA-II; Chi-cuadrado; Optimización multi-objetivo; Algoritmos evolutivos.

ABSTRACT

Abstract— This thesis addresses the feature selection problem as a multi-objective optimization task, motivated by the increasing number of features in datasets used in modern machine learning applications. The goal is to identify a reduced subset of features that improves classification accuracy. We propose using the chi-square statistical test to initialize the evolutionary algorithm NSGA-II, taking advantage of its population-based nature to find a set of non-dominated, high-quality solutions. Two initialization strategies are proposed and evaluated. Six datasets from different domains were tested, ranging from 16 to 12,600 features. Improvements in accuracy and generalization were observed, especially when initializing with 10 % of the features. Moreover, the proposed strategies did not significantly increase computational time and achieved competitive results compared to the state of art in terms of hypervolume.

Keywords— Feature selection; NSGA-II; Chi-square; Multi-objective optimization; Evolutionary algorithms.

ÍNDICE DE CONTENIDOS

RESUMEN	III
ABSTRACT	III
ÍNDICE DE FIGURAS	VI
ÍNDICE DE TABLAS	VI
INTRODUCCIÓN	1
DEFINICIÓN DEL PROBLEMA	4
1.1 Problema de selección de características	4
1.2 Problema de optimización multi-objetivo	7
ESTADO DEL ARTE	11
2.1 Acercamientos de algoritmos de selección de características	11
2.1.1 Modelos de filtro	11
2.1.2 Modelos de envoltura	13
2.1.3 Modelos integrados	14
2.2 Algoritmos de búsqueda multi-objetivo para el problema de selección de ca- racterísticas	15
PROPUESTA DE SOLUCIÓN	20
3.1 Non-Dominated Sorting Genetic Algorithm II (NSGA-II)	20
3.2 Representación	22
3.3 Funciones Objetivo	22
3.4 Inicialización	23
3.4.1 Inicialización base	23
3.4.2 Inicialización con chi-cuadrado	23
3.4.3 Inicialización híbrida	25
3.5 Operadores de transformación	25
3.5.1 Operador de selección	25
3.5.2 Operador de cruzamiento	26
3.5.3 Operador de mutación	27
VALIDACIÓN DE LA SOLUCIÓN	28
4.1 Conjuntos de datos	28
4.2 Configuraciones experimentales	29
4.3 Algoritmo de predicción	30
RESULTADOS	32
5.1 Resultados con el 10 % de características iniciales	32
5.1.1 Error y cantidad de características	32

5.1.2	Frentes de Pareto	35
5.1.3	Hipervolumen	38
5.2	Resultados con el 50 % de características iniciales	40
5.2.1	Error y cantidad de características	40
5.2.2	Frentes de Pareto	43
5.2.3	Hipervolumen	45
5.3	Comparaciones con el estado del arte	47
5.4	Tiempos obtenidos	48
CONCLUSIONES		50
REFERENCIAS BIBLIOGRÁFICAS		53

ÍNDICE DE FIGURAS

1	Proceso general de selección de características.	6
2	Ejemplo de representación de un frente de Pareto.	8
3	Ejemplo de representación de la medida de hipervolumen.	9
4	Ejemplo de representación de la medida de distancia generacional invertida, donde las soluciones de color azul son parte del conjunto óptimo de Pareto y las de color rojo representan las soluciones no dominadas que se obtuvieron.	10
5	Ejemplo de k-NN con k=2.	15
6	Ejemplo visual del proceso elitista en NSGA-II.	21
7	Ejemplo de solución utilizando la representación.	23
8	Frentes de Pareto del proceso de entrenamiento - 10 % de características iniciales.	36
9	Frentes de Pareto del proceso de prueba - 10 % de características iniciales.	37
10	Frentes de Pareto del proceso de entrenamiento - 50 % de características iniciales.	43
11	Frentes de Pareto del proceso de prueba - 50 % de características iniciales.	45

ÍNDICE DE TABLAS

1	Ejemplo de representación de un conjunto de datos.	5
2	Conjuntos de datos	28
3	Parámetros de NSGA-II.	29
4	Parámetros utilizados en las inicializaciones.	30
5	Información relevante de los conjuntos de datos.	31
6	Resultados del error porcentual en el proceso de entrenamiento - 10 % de características iniciales.	33
7	Resultados del error porcentual en el proceso de prueba - 10 % de características iniciales.	34
8	Resultados de la cantidad de características - 10 % de características iniciales.	34
9	Hipervolumen en el proceso de entrenamiento - 10 % de características iniciales.	38
10	Hipervolumen en el proceso de prueba - 10 % de características iniciales.	39
11	Resultados del error porcentual en el proceso de entrenamiento - 50 % de características iniciales.	40
12	Resultados del error porcentual en el proceso de prueba - 50 % de características iniciales.	41
13	Resultados de la cantidad de características - 50 % de características iniciales.	42
14	Hipervolumen en el proceso de entrenamiento - 50 % de características iniciales.	46
15	Hipervolumen en el proceso de prueba - 50 % de características iniciales.	47
16	Comparación de hipervolumen en el proceso de entrenamiento con CNSGA-II - 50 % de características iniciales.	48

17	Comparación de hipervolumen en el proceso de prueba con CNSGA-II - 50 % de características iniciales.	48
18	Comparación de hipervolumen en el proceso de entrenamiento con CNSGA-II - 10 % de características iniciales.	48
19	Comparación de hipervolumen en el proceso de prueba con CNSGA-II - 10 % de características iniciales.	49
20	Tiempo promedio de ejecución de los experimentos (en segundos) - 10 % de características iniciales.	49
21	Tiempo promedio de ejecución de los experimentos (en segundos) - 50 % de características iniciales.	49

INTRODUCCIÓN

El manejo de grandes volúmenes de datos es un desafío que ha estado creciendo en los últimos años y que se presenta en diversas áreas, como la biomedicina y el análisis de imágenes. Uno de los problemas más relevantes cuando se trabaja con datos de alta dimensionalidad es la selección de características, un proceso muy importante en el preprocesamiento de datos y la mejora de resultados en modelos de aprendizaje automático [Cai *et al.*, 2018].

El problema de selección de características presenta dos objetivos conflictivos: minimizar el número de características seleccionadas y maximizar la precisión del modelo de clasificación. Siguiendo esta lógica, este se puede abordar como un problema de optimización multi-objetivo en el que se buscan soluciones que presenten buenos resultados para ambos casos, lo cual se vuelve desafiante al tener un gran volumen de datos. La selección de características es un problema NP-completo (NP-hard), dado que su complejidad escala exponencialmente con el número de características [Tang *et al.*, 2014]. En base a esto, se han realizado diversas propuestas para abordar este tipo de problemas.

Una de las técnicas más eficaces para abordar problemas de optimización multi-objetivo es el uso de algoritmos evolutivos, los cuales están inspirados en la evolución natural de los seres vivos. Estos algoritmos exploran grandes espacios de soluciones, lo que permite abordar problemas complejos como la selección de características con múltiples objetivos conflictivos. En este trabajo se utiliza Non-Dominated Sorting Genetic Algorithm II (NSGA-II), un algoritmo evolutivo multi-objetivo muy utilizado gracias a su capacidad de encontrar un conjunto de soluciones óptimas no dominadas que representan las mejores soluciones para los objetivos utilizados [Deb *et al.*, 2002]. También se han realizado variantes de NSGA-II para abordar el problema de selección de características con los objetivos de mejorar la eficiencia y la generalización. En este contexto, la generalización se refiere a la capacidad del algoritmo para mantener un buen rendimiento con datos que no se utilizaron para entrenar el algoritmo. Entre estas variantes está [Miyandoab *et al.*, 2023], el cual propone una versión compacta del algoritmo para reducir el consumo de memoria, y [Xue *et al.*, 2023], el cual adapta la inicialización y los operadores basándose en ReliefF [Kononenko, 1994].

NSGA-II es conocido por su capacidad de mantener un buen equilibrio entre la diversidad de soluciones y la convergencia hacia el óptimo, todo esto a través de un proceso iterativo de selección, cruzamiento y mutación. Este algoritmo busca soluciones que no sean dominadas por ninguna otra, esto en base a los objetivos utilizados. Sin embargo, uno de los desafíos de NSGA-II es la inicialización de la población, ya que una mala selección de las características iniciales puede perjudicar la eficiencia del algoritmo. Por esto mismo es que se propone una nueva inicialización a través de la prueba estadística de chi-cuadrado χ^2 , una técnica que permite identificar las características que tienen una mayor dependencia con la variable objetivo [Jin *et al.*, 2006].

El uso de chi-cuadrado en el proceso de inicialización tiene como objetivo reducir la dimen-

sionalidad del problema al seleccionar características más relevantes desde el inicio, generando así una mejora en la precisión del modelo. Además de la inicialización basada en chi-cuadrado, se propone una variante híbrida que le agrega un factor de aleatoriedad, lo que permite explorar más el espacio de soluciones, dado que se incorporan tanto características relevantes como otras que podrían ser útiles si se consideran en conjunto.

El objetivo principal de este trabajo es demostrar la capacidad de la prueba estadística de chi-cuadrado para brindar una solución inicial que permita obtener mejores resultados para el problema de selección de características en el contexto de optimización multi-objetivo. Se espera que las inicializaciones propuestas permitan obtener una mejor precisión y el algoritmo sea capaz de adaptarse a datos con mayor dimensionalidad. Además, se evalúa el impacto de inicializar NSGA-II con un 10 % y un 50 % de características, con el objetivo de analizar cómo influye esta proporción en el rendimiento del algoritmo.

La motivación de este trabajo se encuentra en la necesidad de desarrollar un algoritmo que no solo resuelva el problema de selección de características eficientemente y con un buen desempeño, sino que también sea versátil y pueda aplicarse en diversas áreas y con distintos volúmenes de datos.

Este trabajo busca responder las siguientes preguntas:

- ¿Puede la prueba estadística de chi-cuadrado mejorar la calidad de la población inicial en NSGA-II aplicada al problema de selección de características?
- ¿La implementación híbrida mejora el equilibrio entre exploración y explotación durante la búsqueda evolutiva?
- ¿Existe una diferencia a la hora de inicializar NSGA-II con un 10 % y un 50 % de características?
- ¿Se obtienen mejores resultados de hipervolumen a comparación del estado del arte?
- ¿Qué tanto afectan las nuevas inicializaciones a los tiempos de ejecución del algoritmo?
- ¿Las implementaciones propuestas son capaces de aplicarse en distintas áreas y volúmenes de datos?

Este escrito está estructurado de la siguiente forma:

1. Definición del problema, donde abordan teóricamente los problemas de selección de características y optimización multi-objetivo.
2. Estado del arte, donde se detallan los enfoques de selección de características más conocidos y algunos de los algoritmos evolutivos multi-objetivo aplicados a este problema.

3. Propuesta de solución, donde se presenta el algoritmo NSGA-II y las variantes de inicialización propuestas. También se detalla la representación, funciones objetivo y operadores de transformación utilizados.
4. Validación de la solución, donde se muestran los conjuntos de datos, los parámetros y el algoritmo de aprendizaje automático utilizado en los experimentos.
5. Resultados, donde se muestran los resultados obtenidos de los experimentos, se presentan gráficos y tablas comparativas, y se realizan diversos análisis al respecto.
6. Conclusiones, donde se sintetizan los análisis realizados y se comenta sobre el trabajo futuro.

DEFINICIÓN DEL PROBLEMA

En este capítulo se presenta el problema a resolver. El problema considera dos grandes áreas. Entiéndase selección de características y el problema de optimización multi-objetivo.

1.1. Problema de selección de características

El problema de selección de características surge cuando se trabaja con máquinas de aprendizaje que utilizan grandes volúmenes de datos y bases de datos que consideran miles de características. De esta gran cantidad de información, es común detectar características que sean irrelevantes o redundantes para la clasificación, incrementando la complejidad del proceso sin mostrar ninguna ganancia en los resultados. Esto define el desafío de reducir la dimensionalidad de los datos sin comprometer el desempeño del modelo para mejorar la eficiencia y eficacia de los acercamientos de aprendizaje de máquinas en aplicaciones de la vida real [Tang *et al.*, 2014].

El problema de selección de características es un problema complejo de la ingeniería que apunta a identificar un subconjunto óptimo de características relevantes para el proceso, ayudando a reducir la complejidad cuando se manejan los datos. Este problema presenta naturalmente dos objetivos conflictivos: minimizar el número de características seleccionadas y maximizar la precisión del modelo de clasificación. Debido a estos objetivos, el problema se puede abordar como un problema de optimización mono-objetivo o multi-objetivo, dependiendo del enfoque que se utilice. La selección de características no transforma los datos originales, sino que selecciona un subconjunto de las características existentes [Cai *et al.*, 2018].

Las características se consideran relevantes cuando aportan información útil y no se encuentran incluidas implícitamente en otras características. Se consideran irrelevantes si no están directamente asociadas con el concepto objetivo y afectan al proceso de aprendizaje. Por último, se consideran redundantes si no aportan nueva información con respecto a las características relevantes [Tang *et al.*, 2014].

El parámetro necesario para cualquier variante del problema es un conjunto de datos. El conjunto de datos puede representarse como se muestra en la tabla 5, donde cada fila corresponde a una muestra del conjunto de datos y cada columna corresponde a una característica. En total, un conjunto X tiene D características y una variable objetivo (c_0) que contiene N clases. El objetivo es utilizar las características para predecir correctamente las clases de c_0 . Las columnas pueden ser binarias, como la columna c_1 y c_2 , y pueden ser multiclase, como la columna c_3 y c_4 . Además, existen columnas que pueden estar desbalanceadas, como es el caso de la columna c_5 .

Dado que un conjunto de datos con D características posee 2^D posibles subconjuntos, el es-

c0	c1	c2	c3	c4	c5	...	c_D
0	0	0	0	0	0		2
2	1	1	1	1	0		1
3	1	0	2	2	0		0
1	0	1	2	2	1		0
1	1	0	0	0	0		1
1	1	0	1	1	2		2

Tabla 1: Ejemplo de representación de un conjunto de datos.

pacio de búsqueda crece exponencialmente a medida que aumenta la dimensionalidad. Esto hace que el problema de selección de características se considere NP-completo (NP-hard), ya que no es factible evaluar todos los posibles subconjuntos con una gran cantidad de características. Por ello se suele abordar este problema a través de heurísticas o metaheurísticas como hill-climbing y algoritmos genéticos [Tang *et al.*, 2014].

Una vez que se escoge el conjunto de datos que se busca optimizar, se realiza el proceso que se observa en la figura 1. Primero se obtiene un subconjunto de características del conjunto original, luego se entrena un modelo de clasificación utilizando el subconjunto obtenido, se evalúa el desempeño del modelo a través de medidas como la precisión y se decide si se utiliza ese subconjunto o se vuelve a probar con uno nuevo. Este último paso depende del acercamiento que se utilice para abordar el problema. Los acercamientos de filtro no realizan este paso debido a que son independientes del modelo de clasificación y solo se enfocan en evaluar el modelo a través de otras métricas matemáticas para obtener el mejor subconjunto de características, entrenando al final un modelo con ese subconjunto para evaluar su precisión si es que se desea. En cambio, los modelos de envoltura y los integrados si toman en cuenta la precisión obtenida para decidir el subconjunto de características que se entrega al final del algoritmo [Tang *et al.*, 2014].

Dado que existen métodos de selección de características que no están diseñados para trabajar directamente con variables continuas, es común aplicar una etapa previa de discretización sobre los datos antes de iniciar el proceso de selección.

La discretización es una técnica de preprocesamiento utilizada en aprendizaje automático para transformar variables con valores continuos en valores discretos o nominales. Esta técnica es importante y se aplica en casos donde los algoritmos solo aceptan datos discretos o mejoran su eficiencia al utilizarlos [Tran *et al.*, 2018].

Además de facilitar el uso de algunos algoritmos, la discretización permite que se ignoren pequeñas fluctuaciones o posible ruido en los datos. Por último, los datos discretos son más compactos que los datos continuos, lo que permite que se requiera menor memoria y se

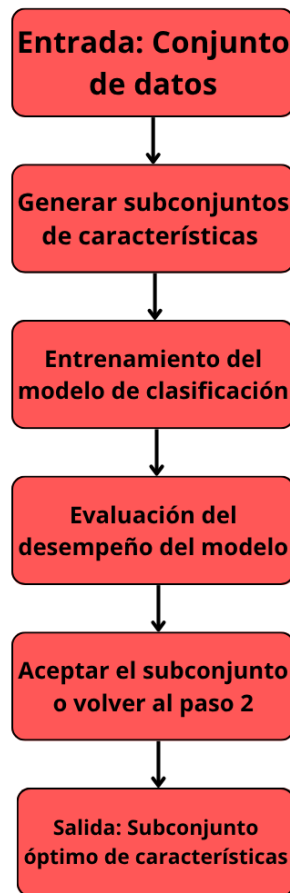


Figura 1: Proceso general de selección de características.
Fuente: Elaboración propia.

mejore la eficiencia de los algoritmos de aprendizaje automático.

Los tipos de discretización se suelen clasificar de la siguiente forma:

- **Supervisada:** utiliza los valores de la variable objetivo para crear los intervalos.
- **No supervisada:** no utiliza los valores de la variable objetivo, sino que se basa en la distribución de los valores del resto de variables.
- **Univariada:** discretiza cada variable por separado, asumiendo que no existen interacciones con otras características.
- **Multivariada:** discretiza considerando múltiples variables a la vez para conservar posibles interacciones entre características relevantes.

Muchos métodos de selección de características requieren datos discretos, por lo que comúnmente se realiza previamente. Sin embargo, una discretización univariada podría perju-

dicar el desempeño de la etapa de selección, dado que se podría perder información relevante con respecto a la interacción entre características [Tran *et al.*, 2018].

Resolver este problema no solo permite mejorar la precisión y la eficiencia en términos computacionales, sino que también facilita una mejor comprensión del modelo de aprendizaje y del comportamiento de los datos [Cai *et al.*, 2018].

1.2. Problema de optimización multi-objetivo

Los problemas multi-objetivo corresponden a la optimización de dos o más funciones objetivo a la vez. Estos se pueden abordar transformando el problema a uno mono-objetivo (por ejemplo, aplicándole pesos a cada objetivo) u obteniendo un conjunto de soluciones que cumplan con los requerimientos y las restricciones establecidas.

La dificultad de estos problemas radica en que generalmente no existe una solución que incluya el óptimo de todas las funciones objetivo. Esto se debe a que los objetivos suelen ser conflictivos entre sí. Un ejemplo muy claro es el problema de selección de características, debido a que se busca aumentar la precisión obtenida a través de un modelo, pero para lograrlo, la idea sería entregar más información para ser más específico, provocando que se aumente la cantidad de características utilizadas.

Matemáticamente hablando, la forma de definir las soluciones de un problema multi-objetivo es a través de vectores como $x^* = [x_1^*, x_2^*, \dots, x_n^*]^T$ que satisfacen las restricciones establecidas y optimizan cada función objetivo, las cuales se representan de la siguiente forma:

$$f(x) = [f_1(x), f_2(x), \dots, f_m(x)]$$

donde $x^* \in S$, un conjunto que representa el espacio de todas las soluciones que satisfacen las restricciones.

Para abordar el conflicto existente entre los objetivos, se utiliza la dominancia de Pareto. Se comparan las soluciones del conjunto S , determinando la dominancia que tenga una sobre otra en base a las funciones objetivo. Matemáticamente hablando, la solución $x^* = [x_1^*, x_2^*, \dots, x_n^*]^T$ domina a la solución $y^* = [y_1^*, y_2^*, \dots, y_n^*]^T$ si se cumple lo siguiente:

1. $\forall i \in \{1, \dots, k\}, f_i(x^*) \leq f_i(y^*)$
2. $\exists i \in \{1, \dots, k\}, f_i(x^*) < f_i(y^*)$

Al cumplirse ambas condiciones, se puede asegurar que $x^* \prec y^*$, es decir, x^* domina a y^* .

De esta manera es que se puede obtener un conjunto que contenga todas las soluciones no dominadas. Estas soluciones se llaman Pareto óptimas y pasan a ser parte del frente de

Pareto. Este se define como $P = \{x \in S, \nexists y \in S : x \prec y\}$ y es donde se presentan los objetivos. Cada una de las soluciones que se encuentran en este frente son parte del conjunto de Pareto. En la figura 2 se muestra una forma de representar las soluciones no dominadas obtenidas en un problema de minimización con dos objetivos a través de un frente de Pareto.

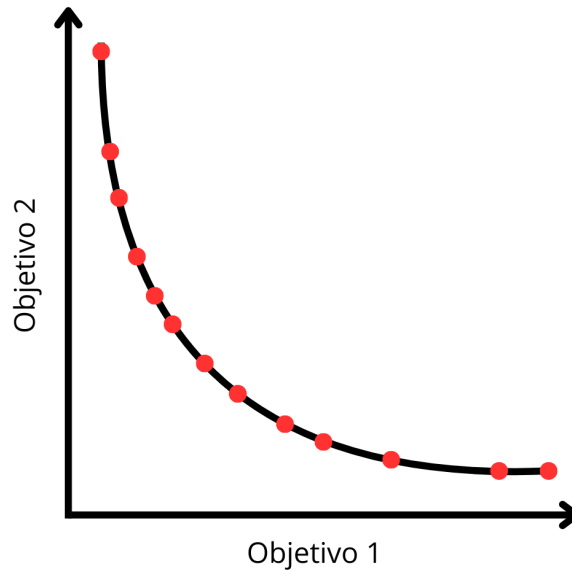


Figura 2: Ejemplo de representación de un frente de Pareto.

Fuente: Elaboración propia.

Lo que se busca a fin de cuentas es obtener un conjunto que represente al frente de Pareto de forma eficiente, obteniendo buenos resultados en un tiempo reducido [Xu *et al.*, 2020].

Los algoritmos evolutivos han sido muy utilizados en la resolución de problemas multi-objetivo. Estos corresponden a técnicas de búsqueda que están inspiradas en la evolución biológica de los seres vivos. La idea central es que los individuos de una población compiten por una cantidad limitada de recursos, y solo los más aptos son aquellos que sobreviven generación tras generación. Este concepto puede aplicarse en problemas de optimización en donde se busque minimizar o maximizar una función objetivo [Perera *et al.*, 2023]. En el contexto de optimización multi-objetivo, los algoritmos evolutivos son útiles gracias a su capacidad para explorar más de una solución a la vez. El proceso de selección que realizan determina qué individuos son más aptos para pasar a la siguiente generación, utilizando indicadores de calidad o dominancia de Pareto. La transformación de estas soluciones se lleva a cabo a través de operadores de mutación y cruzamiento, los cuales permiten explorar el espacio de búsqueda y encontrar mejores soluciones. Para poder ajustar e integrar esta información a la nueva población, se pueden aplicar diversas técnicas. Entre estas técnicas existe el elitismo [Deb *et al.*, 2002], el cual permite mantener una convergencia hacia el frente de Pareto [Zhou *et al.*, 2011].

Para asegurar que se obtengan buenos resultados, existen diversos indicadores de calidad multi-objetivo. Entre ellos se encuentra el hipervolumen [Coello *et al.*, 2007]. Esta medida corresponde a la superficie (o al volumen en más de dos objetivos) de la región dominada por las soluciones del frente de Pareto, en donde se considera un punto de referencia para realizar el cálculo de la medida. Mientras mayor sea la superficie cubierta, mejor será el resultado. Para esto se calcula la medida de Lebesgue S , la cual se aprecia en la ecuación (1).

$$\mathcal{S}(A, y_{\text{ref}}) = \Lambda \left(\bigcup_{y \in A} y' \mid y \prec y' \prec y_{\text{ref}} \right), \quad A \subseteq \mathbb{R}^m \quad (1)$$

donde A corresponde al subconjunto del espacio de objetivos, y_{ref} al punto de referencia dominado por todas las soluciones y Λ a la medida de Lebesgue. En la figura 3 se observa un ejemplo simple de hipervolumen.

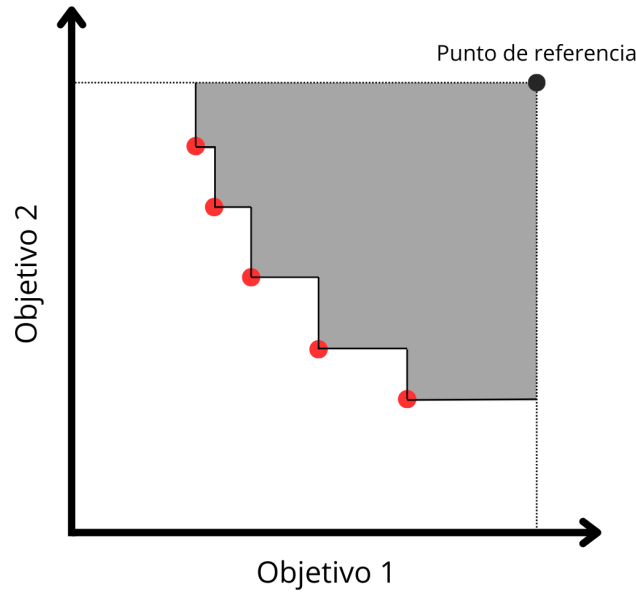


Figura 3: Ejemplo de representación de la medida de hipervolumen.

Fuente: Elaboración propia.

Otro indicador utilizado para este problema es la distancia generacional invertida [Coello *et al.*, 2007], la cual mide la distancia entre las soluciones no dominadas encontradas y las del conjunto óptimo de Pareto. La forma de calcular esta métrica se observa en la ecuación (2).

$$IGD_p = \frac{(\sum_{i=1}^n d_i^p)^{1/p}}{n} \quad (2)$$

Donde n representa el número de vectores en el conjunto óptimo de Pareto, d_i la distancia

Euclidiana entre cada uno de los individuos del conjunto óptimo y la solución más cercana del conjunto de soluciones no dominadas encontrado a cada uno, y p es la norma que se utiliza para calcular la distancia. En la figura 4 se observa un ejemplo simple de la distancia generacional invertida.

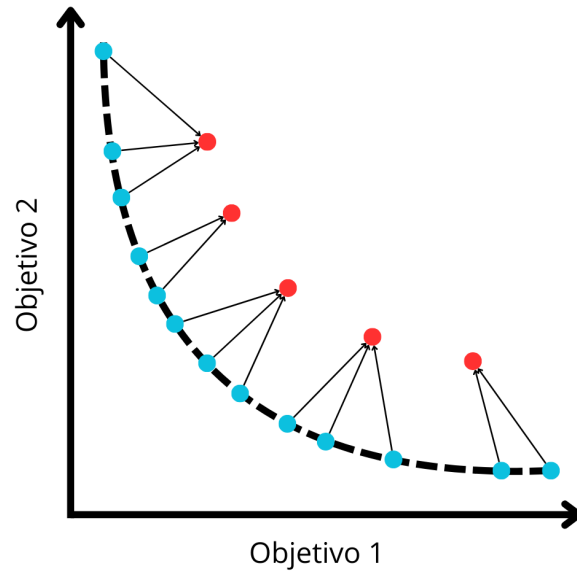


Figura 4: Ejemplo de representación de la medida de distancia generacional invertida, donde las soluciones de color azul son parte del conjunto óptimo de Pareto y las de color rojo representan las soluciones no dominadas que se obtuvieron.

Fuente: Elaboración propia.

ESTADO DEL ARTE

En este capítulo se detallan los enfoques utilizados en la literatura para abordar el problema de selección de características, los cuales se dividen en modelos de filtro, modelos de envoltura y modelos integrados. Por otro lado, se presentan en detalle los principales algoritmos de optimización multi-objetivo que han sido aplicados al problema de este trabajo de título.

2.1. Acercamientos de algoritmos de selección de características

Los algoritmos de selección de característica de la literatura se dividen en tres grandes categorías [Tang *et al.*, 2014]: modelos de filtro, modelos de envoltura y modelos integrados. A continuación se describe cada clasificación de modelos y algunos trabajos relevantes en el área.

2.1.1. Modelos de filtro

Los modelos de filtro evalúan las características sin utilizar ningún algoritmo de clasificación. Estos algoritmos suelen constar de dos pasos. Primero se ordenan las características en función de un criterio específico (como alguna medida estadística) y el segundo paso consiste en elegir las características mejor clasificadas para inducir modelos de clasificación [Tang *et al.*, 2014]. Una de las variantes frecuentemente utilizadas para abordar el problema de clasificación de características es el filtro rápido basado en correlación [Yu y Liu, 2003]. En este se introduce el concepto de correlación predominante y se propone un método eficaz para detectar redundancia entre características sin la necesidad de realizar un análisis completo por pares. El algoritmo calcula los valores de incertidumbre simétrica para cada característica, seleccionando las más relevantes según un umbral predefinido y ordenándolas en orden descendente según su correlación con la clase. Luego de eso, procesa la lista ordenada y elimina las características redundantes, manteniendo solo aquellas cuya correlación con la clase es predominante. Las pruebas mostraron un enfoque no solo más eficiente computacionalmente, sino que también logra reducir la dimensionalidad y mejorar la precisión en comparación con los algoritmos ReliefF [Kononenko, 1994], una variación de un método basado en correlación (CorrSF) [Hall, 2000] y una variación de un método basado en consistencia (ConsSF) [Dash *et al.*, 2000].

Otro enfoque común dentro de los modelos de filtro es la prueba estadística de chi-cuadrado (χ^2), una prueba no paramétrica para analizar las diferencias entre variables categóricas en una población. Al ser no paramétrica, requiere partir de una suposición conocida como hipótesis nula. Esta suposición dice que las características que se están comparando son independientes (no están relacionadas) [Yang y Pedersen, 1997].

El objetivo que tiene es comparar las frecuencias observadas (O), que corresponden a los datos reales, con las frecuencias esperadas (E), que corresponden a los valores que uno esperaría obtener siguiendo una hipótesis de independencia entre las variables. Por esta razón, si los valores observados son similares a los esperados, no se rechaza la hipótesis, y si estos difieren entre sí, entonces se rechaza la hipótesis, lo que indica que puede existir una relación significativa.

En el contexto de aprendizaje automático, es una herramienta utilizada en problemas de clasificación para medir la dependencia entre cada característica y la clase objetivo [Jin *et al.*, 2006]. Se aplica en casos donde los datos sean categóricos o estén discretizados, y permite que las características que presenten una relación significativa con la variable objetivo.

Matemáticamente se calcula como aparece en la fórmula 3:

$$\chi^2 = \sum \frac{(O - E)^2}{E} \quad (3)$$

Para poder obtener los valores de O y E se construye una tabla de contingencia para ver las frecuencias observadas entre las dos variables. El tamaño de la tabla es el múltiplo entre la cantidad de clases de las variables y también corresponde a la cantidad de sumas involucradas en la sumatoria de la fórmula 3. Para poder calcular el valor de las frecuencias esperadas, se utiliza la fórmula 4:

$$E_{ij} = \frac{T_i \cdot T_j}{N} \quad (4)$$

donde E_{ij} corresponde a la frecuencia esperada para la i -ésima fila y la j -ésima columna, T_i al total de frecuencias en la fila i , T_j al total de frecuencias en la columna j y N al total general. La fórmula 3 se calcula para cada una de las características con respecto a la clase objetivo. Es por esto que la complejidad del algoritmo es $O(c_0 \cdot D)$ [Jin *et al.*, 2006].

Estudios como [Yang y Pedersen, 1997] y [Jin *et al.*, 2006] han demostrado que chi-cuadrado es altamente efectivo en el contexto de selección de características, ya que es uno de los métodos más robustos y con mejores resultados en términos de precisión en la clasificación.

Uno de los algoritmos más novedosos de este enfoque se presenta en [Gong *et al.*, 2024], donde se propone CONMI_FS. Este es un algoritmo de selección de características que se basa en la evaluación simultánea de correlaciones lineales y no lineales entre variables, utilizando el coeficiente de correlación de Pearson [Kirch, 2008]. y la información mutua normalizada [Vergara y Estévez, 2014, Hastie *et al.*, 2009]. El objetivo es seleccionar subconjuntos de características altamente relevantes con la variable de clase y con baja redundancia entre ellas. Dado que CONMI_FS representa un enfoque puramente de filtro, prioriza la eficiencia computacional y la capacidad de generalización. Se emplea una estrategia de búsqueda

secuencial hacia adelante, donde en cada iteración se selecciona la característica que maximiza una función de evaluación híbrida, definida como una combinación ponderada del coeficiente de correlación de Pearson y la información mutua normalizada.

El hiperparámetro λ regula la importancia relativa de cada componente de la función, y su valor óptimo, según los experimentos del artículo, fue 0.9, lo que indica una mayor influencia de la información mutua normalizada en el proceso de selección.

El algoritmo fue evaluado en veinte conjuntos de datos obtenidos de los repositorios UCI machine learning repository [Kelly *et al.*, 2023] y KEEL [Alcalá-Fdez *et al.*, 2009]. Estos tienen entre 4 y 85 columnas, entre 80 y 10,992 filas, y entre 2 y 11 clases.

Se comparó con métodos clásicos como selección de características basada en correlación [Hall, 2000], búsqueda basada en consistencia [Dash y Liu, 2003] e información mutua [Vergara y Estévez, 2014], y se utilizó *k-nearest neighbors* [Cunningham y Delany, 2021] [HDS, 2021], *support vector machine* [Cortes y Vapnik, 1995] y árboles de decisión [Quinlan, 1986a] como clasificadores para validar los subconjuntos seleccionados.

Los resultados experimentales mostraron que CONMI_FS obtuvo la mayor tasa de reducción dimensional (80.04 %) y logró la mejor precisión de clasificación en los clasificadores *k-nearest neighbors* (88.82 %) y *support vector machine* (88.98 %). Aunque su rendimiento en árboles de decisión fue ligeramente inferior, los autores lo atribuyen a que este tipo de clasificador se puede beneficiar al utilizar una mayor número de características. No se presenta información relativa a los tiempos de cómputo del algoritmo. Finalmente, se realizaron análisis de estabilidad y pruebas estadísticas (test de Friedman), demostrando que CONMI_FS ofrece una combinación eficaz entre precisión y simplicidad del modelo, siendo especialmente competitivo en escenarios con alta dimensionalidad. Esto último solo lo consideraron en conjuntos de datos con una gran cantidad de muestras, pero con poca cantidad de características.

2.1.2. Modelos de envoltura

Los modelos de filtro seleccionan características independientemente del clasificador utilizado. El problema es que este enfoque ignora por completo el rendimiento del subconjunto de características en el algoritmo de inducción. Por este motivo, los modelos de envoltura utilizan un clasificador específico para evaluar la calidad de las características seleccionadas, ofreciendo así una forma sencilla y potente de abordar el problema. Estos modelos consideran tres pasos. En el primer paso se determina el conjunto de características, luego se evalúa el subconjunto seleccionado y, por último, se repiten los pasos anteriores hasta obtener la calidad deseada. Esta es la razón por la que el modelo de envoltura tiene un mayor coste computacional en comparación con el modelo de filtro [Tang *et al.*, 2014].

Una comparación más detallada entre los modelos de filtro y envoltura se presenta en [Inza *et al.*, 2004], donde el modelo de envoltura se implementó utilizando la técnica secuencial hacia adelante [Kittler, 1978] para identificar los subconjuntos óptimos de características relevantes en el contexto de la bioinformática e iterativamente añadir aquellos que maximizan la precisión del clasificador supervisado. El enfoque de envoltura superó al método de filtro en precisión y disminución de la dimensionalidad. Se utilizaron como clasificadores IB1 [Aha *et al.*, 1991], NB [Langley *et al.*, 1992], C4.5 [Salzberg, 1994] y CN2 [Clark y Niblett, 1989]. Sin embargo, el coste computacional era muy elevado en comparación con el método de filtro, esto considerando que los autores mencionan solamente los tiempos obtenidos por los métodos de envoltura y mencionan que las necesidades computacionales de los métodos de filtro se pueden considerar insignificantes en comparación.

Otra opción que puede utilizarse es el modelo k-nearest neighbors (k-NN), el cual es un algoritmo de aprendizaje supervisado utilizado generalmente para realizar trabajos de clasificación [Cunningham y Delany, 2021, HDS, 2021].

Su funcionamiento se basa en identificar los k puntos más cercanos (llamados vecinos) a un nuevo punto de entrada, los cuales se encuentran en un conjunto de entrenamiento. Para esto, k-NN calcula la distancia entre el nuevo punto y todos los puntos conocidos. Luego, selecciona los k puntos con menor distancia y le asigna al nuevo punto la clase predominante entre los k vecinos. El algoritmo asume que los puntos más cercanos tienen la misma clase. [Cunningham y Delany, 2021].

Para comprender mejor el funcionamiento de k-NN, se presenta un ejemplo básico en la figura 5. Tal y como se puede observar, en el caso de establecer un k igual a 2, los puntos más cercanos al nuevo punto son parte de la clase 2, por lo que a ese nuevo punto se le asigna esta clase.

2.1.3. Modelos integrados

Los modelos integrados abordan la selección de características mediante la construcción de clasificadores. Existen tres tipos de métodos integrados. Los métodos de poda utilizan todas las características para entrenar un modelo y luego intentan eliminar algunas características llevando a cero los coeficientes correspondientes para mantener el rendimiento del modelo [Tang *et al.*, 2014]. En el segundo tipo se encuentran los modelos con un mecanismo incorporado para la selección de características, como ID3 [Quinlan, 1986b] y C4.5 [Salzberg, 1994] que construyen árboles de decisión seleccionando en cada nodo el atributo que maximiza la ganancia de información. Por último, los modelos de regularización pretenden minimizar los errores de ajuste y obligan a que los coeficientes sean lo más pequeños posibles. Las características que llegan a cero se eliminan [Ma y Huang, 2008].

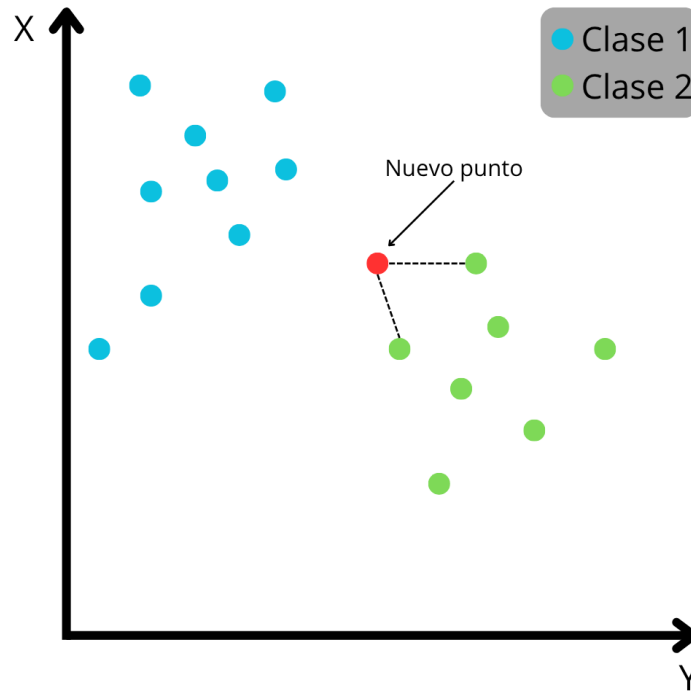


Figura 5: Ejemplo de k-NN con $k=2$.
Fuente: Elaboración propia.

2.2. Algoritmos de búsqueda multi-objetivo para el problema de selección de características

El problema de selección de características es un problema inherentemente multi-objetivo que considera como objetivos clásicos la minimización de la cantidad de características que maximizan la precisión de las predicciones. A continuación se presentan algunos acercamientos basados en búsqueda evolutiva multi-objetivo para el problema.

En [Miyandoab *et al.*, 2023] se propone Compact NSGA-II (CNSGA-II), el cual corresponde a una versión modificada de NSGA-II [Deb *et al.*, 2002] para resolver el problema de selección de características.

Este algoritmo, al igual que NSGA-II, obtiene como resultado final un frente de Pareto. También utiliza ordenamiento no dominado para clasificar las soluciones en niveles según su dominancia y utiliza la misma representación binaria de soluciones de la versión clásica de NSGA-II.

CNSGA-II busca disminuir el consumo de memoria a través del uso de vectores de probabilidad. Los vectores de probabilidad son vectores de valores reales entre 0 y 1 de tamaño D que representan la probabilidad de escoger cada característica. Los vectores de probabilidad se utilizan en cada iteración para generar nuevos individuos. La disminución del consumo de

memoria se logra debido a que los vectores de probabilidad están asociados a un conjunto de soluciones denominadas líderes que representa a los mejores individuos. De esta forma, se reduce la cantidad de soluciones almacenadas en memoria y se guarda un pequeño conjunto de vectores de probabilidad que resume de forma compacta cómo deberían ser los nuevos individuos. Por esta misma razón es que, a diferencia de NSGA-II, CNSGA-II no utiliza los operadores de transformación clásicos de mutación y cruzamiento.

El acercamiento propuesto considera dos funciones objetivo. La ecuación (5) corresponde al error de clasificación y la ecuación (6) corresponde al porcentaje de características seleccionadas. Estas representan los dos objetivos conflictivos de la sección 1.1.

$$\text{Error de Clasificación} = 1 - \frac{\text{Predicciones Correctas}}{\text{Total de Predicciones}} \quad (5)$$

$$\text{Porcentaje de Características Seleccionadas} = \frac{\text{Cantidad de 1s}}{\text{Total de características}} \quad (6)$$

Las principales diferencias entre CNSGA-II y el algoritmo original se listan a continuación:

1. Inicialización: Se inicializan λ vectores de tamaño D con valores 0.5 como los vectores de probabilidad. Con esto, la probabilidad de seleccionar cada característica inicialmente es del 50 %. Se genera una población aleatoria de tamaño λ , se calculan las funciones objetivo para cada individuo y se les considera líderes. Por último, se le aplica ordenamiento no dominado para obtener el primer frente de Pareto.
2. Actualizar vectores de probabilidad: Para actualizar los vectores de probabilidad se realiza una conversión a binario, donde las características son 1 si es que el valor del vector en esa característica es mayor a 0.5 ó 0 en caso contrario. Luego realiza el cálculo de la distancia entre los vectores de probabilidad binarios y los líderes actuales, utilizando para esto la distancia de Hamming. La distancia de Hamming calcula la cantidad de posiciones en las que dos vectores difieren. Se asigna cada vector de probabilidad al líder más cercano y se actualizan estos vectores en base al líder asignado, sumando un porcentaje a la característica y restándose ese valor en caso que el líder no haya sido asignado. Por último, se limitan los vectores de probabilidad dentro de un rango predefinido para asegurar que sus valores permanezcan en el rango 0 y 1.
3. Generación de nuevas soluciones candidatas: Se genera un nuevo individuo a partir de cada vector de probabilidad, se evalúa y se añade a la población.
4. Selección: Se utiliza ordenamiento no dominado para determinar el frente de Pareto. Se calcula la distancia de aglomeración y se identifican los λ mejores individuos no dominados, conservándose en la población. Por último, se revisa si el tamaño de la población no ha superado el tamaño máximo permitido. En dicho caso, se conservan

los mejores individuos. No se especifica cómo determina los mejores individuos, por lo que se asume que lo realiza a través de la distancia de aglomeración.

Se utilizaron cinco conjuntos de datos de gran escala en los campos de microarrays y reconocimiento de imágenes o rostros. Estos tienen entre 2400 y 11340 columnas, 100 y 210 filas, y entre 3 y 10 clases. Los conjuntos de datos son warpAR10P, warpPIE10P, TOX-171, pixraw10P y CLL-SUB-111. Estos se obtuvieron desde [Zhao *et al.*, 2010a].

Para evaluar el desempeño del algoritmo se realiza una división de los datos, donde el 80 % se utiliza para entrenamiento y el resto para pruebas. Cada algoritmo se ejecuta 10 veces, donde en cada ejecución varían los conjuntos de entrenamiento/prueba.

Por último, se emplea *k*-nearest neighbors y facebook ai similarity search [Johnson *et al.*, 2017] para obtener el error de clasificación de cada individuo. Utilizan un valor de $k = 5$ para todos los conjuntos de datos excepto para el último, donde $k = 4$. El trabajo no explica la razón del valor de k utilizado para cada conjunto de datos. Se utiliza el indicador de hipervolumen para comparar los algoritmos de optimización. Se realizan comparaciones con respecto a NSGA-II clásico de representación binaria. Se demostró que el algoritmo propuesto superó a NSGA-II en términos de hipervolumen y de porcentaje de características en todos los conjuntos de datos, mientras que en el error de clasificación solo superó a NSGA-II en tres de cinco conjuntos de datos.

Otro acercamiento que utiliza algoritmos de búsqueda multi-objetivo es el propuesto en [Xue *et al.*, 2023]. En este estudio se propone MOFS-RFGA, el cual corresponde a un NSGA-II al que se le incorpora la técnica ReliefF. Este enfoque híbrido busca mostrar las ventajas tanto de los métodos de filtro como los de envoltura, aprovechando la rapidez del primero y la precisión del segundo.

Al igual que NSGA-II, el algoritmo entrega un conjunto de soluciones no dominadas que forman el frente de Pareto. También tiene una estructura similar, utilizando la misma representación binaria, el ordenamiento no dominado, la estrategia elitista y la distancia de aglomeración.

MOFS-RFGA incorpora la información de pesos asociados a la características obtenidos por ReliefF tanto en la inicialización como en el cruzamiento y la mutación. La población inicial se genera a través de un torneo de selección binario donde se utilizan los pesos asignados por ReliefF para tener más probabilidad de asignar las características con mayor peso, guiando así la búsqueda desde un comienzo. Se propone un cruzamiento 3-a-1, donde se identifican características comunes y se genera un solo hijo utilizando los pesos asignados como guía para la herencia. Por último, el operador de mutación propuesto utiliza los pesos para decidir si eliminar o agregar una característica al individuo.

ReliefF computa los pesos calculando la diferencia de valores entre una muestra seleccionada aleatoriamente y sus vecinos. Si hay diferencia entre vecinos que poseen la misma clase, entonces se reduce el peso, y si hay diferencias con vecinos que poseen otras cla-

ses, entonces el peso de la característica aumenta. Para los detalles matemáticos, revisar en [Xue *et al.*, 2023].

El algoritmo propuesto considera dos funciones objetivo. Estas representan los dos objetivos conflictivos de la sección 1.1. La ecuación (7) corresponde al error de clasificación. En esta, X es la solución, K el número de particiones en la validación cruzada, y N_{Error}^l y N_{All}^l la cantidad de particiones mal predichas y el número de particiones en el conjunto de datos l .

$$\min f_1(X) = \left(\frac{1}{K} \sum_{l=1}^K \frac{N_{Error}^l}{N_{All}^l} \right) \times 100 \% \quad (7)$$

La función (8) corresponde a la minimización de la cantidad de características seleccionadas, donde cada x_i representa

$$\min f_2(X) = \sum_{i=1}^D x_i \quad (8)$$

El algoritmo se evaluó utilizando veinte conjuntos de datos. Estos tienen entre 30 y 2,000 columnas, entre 32 y 6,435 filas, y entre 2 y 26 clases. Los conjuntos fueron extraídos del *UCI Machine Learning Repository* [Kelly *et al.*, 2023].

Se emplea k -nearest neighbors para clasificar los subconjuntos de características y k -fold cross-validation para generar particiones que contengan todas las categorías. Se utiliza un $k = 3$ para todos los conjuntos de datos. Se comparó contra siete algoritmos de optimización multi-objetivo, los cuales son NSGA-II, NSPSOFS, NSGA-II/SDR, SparseEA, CMDPSOFS y MOEA/D. MOFS-RFGA demostró una mejora significativa medida a través de las medidas de distancia generacional invertida e hipervolumen, logrando mejores resultados a usando distancia generacional invertida en 13 y 11 conjuntos para los subconjuntos de entrenamiento y pruebas, y a través del indicador de hipervolumen en 17 y 15 conjuntos.

Hasta ahora se han mencionado dos algoritmos que utilizan una versión modificada de NSGA-II para abordar el problema de selección de características. Sin embargo, existen otros métodos que también se pueden implementar para este problema.

En [Amoozegar y Minaei-Bidgoli, 2018] se presenta un acercamiento basado en Particle swarm optimization [Kennedy y Eberhart, 1995]. Particle swarm optimization es una técnica metaheurística que está inspirada en el comportamiento que tienen los enjambres de aves y los cardúmenes de peces al moverse. En el algoritmo cada partícula del enjambre corresponde a una solución que se desplaza por el espacio de búsqueda. Para encontrar la solución óptima, cada partícula utiliza la información tanto de la mejor posición explorada por ella misma como la de sus vecinas para desplazarse. El rendimiento de cada solución se mide en base a una función de aptitud predefinida. En contextos de optimización multi-

objetivo, esta técnica ha ganado más atención en los últimos años gracias a que se puede ajustar con pocos parámetros, converge rápidamente a la solución óptima y maneja bien el costo computacional [Amoozegar y Minaei-Bidgoli, 2018].

En [Amoozegar y Minaei-Bidgoli, 2018] se propone una variante de PSO llamada RFP SOFS (Ranked Feature PSO Feature Selection) para abordar el problema de selección de características. Este método le asigna niveles a las características en base a la cantidad de veces que se encuentra cada una en el conjunto de soluciones no dominado, utilizando esta información para mejorar la calidad de este conjunto y guiar el movimiento para encontrar mejores soluciones. Se utiliza *k-nearest neighbors* con $k=5$ y validación cruzada con 10 pliegues. El algoritmo propuesto se comparó con NSGA-II [Deb *et al.*, 2002] y tres variantes multi-objetivo de PSO, las cuales son CMDPSOFS [Xue *et al.*, 2013], HMP SOFS [Zhang *et al.*, 2015] y MOPSO [Cagnina *et al.*, 2005] utilizando 9 conjuntos de datos obtenidos de la UCI machine learning repository [Kelly *et al.*, 2023]. Se utilizaron varios indicadores de desempeño, entre ellos el hipervolumen. Los resultados obtenidos mostraron que RFP SOFS es superior al resto de algoritmos en conjuntos de datos de alta dimensión, mientras que en baja dimensión obtenía resultados similares o ligeramente superiores.

Differential evolution es un acercamiento heurístico enfocado en minimizar las funciones no lineales y no diferenciables [Storn y Price, 1997]. Funciona mediante una población de vectores (soluciones) que evoluciona a través de las generaciones mediante operadores de mutación, cruzamiento y selección.

En [Mlakar *et al.*, 2017] se utiliza su variante multi-objetivo (DEMO) para el reconocimiento de expresiones faciales. En esta, DE optimiza tanto la precisión del clasificador como el número de características seleccionadas. El procedimiento consiste en extraer características, realizar la selección a través de DEMO y evaluar el subconjunto utilizando support vector machine. Estos últimos dos pasos se repiten hasta alcanzar la cantidad de generaciones establecida. Se utilizaron tres conjuntos de datos, a los cuales se les aplicaron tres particiones aleatorias independientes. Cada partición se divide en 70 % de las muestras para el entrenamiento y 30 % para las pruebas. Para cada partición, se ejecutó 10 veces el algoritmo propuesto. Se comparan los valores obtenidos a través de las funciones objetivo (cantidad de características seleccionadas y tasa de reconocimiento) y se utilizan medidas estadísticas como la media y la desviación estándar para evaluar el desempeño frente al algoritmo base. Además, se realizan comparaciones con 16 enfoques del estado del arte. Los resultados muestran como la propuesta obtuvo un desempeño competitivo y en algunos casos superior en comparación con los métodos existentes.

PROPUESTA DE SOLUCIÓN

En este capítulo se describe el algoritmo evolutivo multi-objetivo NSGA-II, el cual se utiliza como base para abordar el problema de selección de características. Además, se proponen dos variantes que modifican el proceso de inicialización, utilizando la prueba estadística chi-cuadrado como estrategia para obtener mejores resultados iniciales.

Primero se presenta la descripción general de NSGA-II. Luego de eso, se detallan los elementos específicos de la propuesta, los cuales corresponden a la representación, las funciones objetivo, las inicializaciones y los operadores de transformación.

3.1. Non-Dominated Sorting Genetic Algorithm II (NSGA-II)

Non-Dominated Sorting Genetic Algorithm II es un algoritmo evolutivo multi-objetivo propuesto en [Deb *et al.*, 2002]. Está basado en una población de individuos (soluciones) que se ordenan por frentes de dominancia, para así lograr acercarse al frente de Pareto sin la necesidad de transformar el problema multi-objetivo a uno mono-objetivo, como ocurre en enfoques clásicos.

El algoritmo 1 muestra la estructura principal de NSGA-II. El algoritmo requiere establecer el tamaño de la población (λ), la probabilidad de cruzamiento p_c , la probabilidad de mutación p_m y la cantidad de generaciones $MaxGen$.

Algorithm 1 NSGA-II

Require: $\lambda, p_c, p_m, MaxGen$

Ensure: frente de Pareto F

```

1:  $g = 0$ 
2:  $P_g = \text{Generar población inicial}(\lambda)$ 
3:  $Q_g = \text{Generar descendencia usando selección, cruzamiento y mutación}(\lambda, p_c, p_m)$ 
4: while  $g < MaxGen$  do
5:    $R_g = \text{Combinar } P_g \cup Q_g$ 
6:    $F = \text{Clasificar } R_g \text{ en frentes no dominados } (F_1, F_2, \dots)$ 
7:    $P_{g+1} = \text{Obtener nueva población}(\lambda)$ 
8:    $Q_{g+1} = \text{Generar descendencia usando selección, cruzamiento y mutación}(\lambda, p_c, p_m)$ 
9:    $g = g + 1$ 
10: end while
11:  $R_g = \text{Combinar } P_g \cup Q_g$ 
12:  $F = \text{Clasificar } R_g \text{ en frentes no dominados } (F_1, F_2, \dots)$ 
13: return  $F_1$ 

```

En el pseudocódigo, P_g representa la población de padres y Q_g la población de descendientes

en la generación g , mientras que R_g es la población combinada. $F = (F_1, F_2, \dots)$ agrupa los frentes no dominados obtenidos a través de la clasificación de R_g , donde F_1 es el que contiene las soluciones que no son dominadas por ninguna otra.

NSGA-II comienza con la inicialización de la población en la línea 2. Luego, los operadores de selección genética, cruce y mutación son aplicados para generar las soluciones descendientes en la línea 3. Entre las líneas 4 y 10 se realiza una elección elitista para crear la población de la siguiente generación. La selección se realiza considerando la población de padres e hijos. Este procedimiento es implementado entre las líneas 5 y 7.

Al final de cada generación, se genera nueva descendencia y el proceso se repite. El primer mejor frente no dominado obtenido de la última población de padres e hijos se devuelve en la línea 13 como salida del procedimiento.

En NSGA-II, la naturaleza multi-objetivo de los problemas es gestionada mediante un ordenamiento elitista no dominado. Esto se consigue a través de la selección/remoción iterativa del frente de Pareto no dominado y la distancia de hacinamiento.

La figura 6 ilustra el proceso. En la primera etapa, todas las soluciones de la población combinada $R_g = P_g \cup Q_g$ se clasifican en frentes no dominados, donde el frente F_1 corresponde a aquel que contiene las soluciones que no son dominadas por ninguna otra, el frente F_2 siendo el que contiene las soluciones que solo son dominadas por las de F_1 , y así sucesivamente. Esta evaluación permite establecer una jerarquía basada en la dominancia de Pareto. Luego, se construye la nueva población P_{g+1} , incorporando de forma secuencial las soluciones de los frentes menos dominados hasta completar el tamaño de la población.

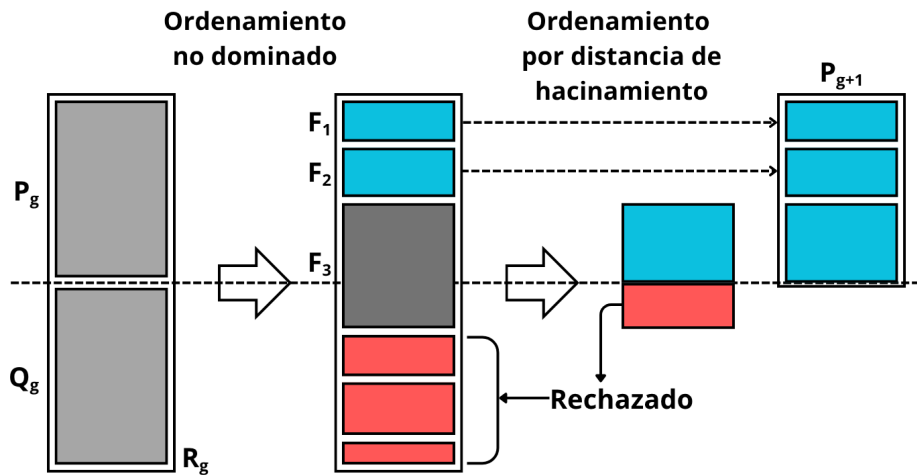


Figura 6: Ejemplo visual del proceso elitista en NSGA-II.

Fuente: Elaboración propia.

Si es que existe un $F_i \in F$ que no es posible incluirlo por completo a la nueva población, se utiliza la distancia de hacinamiento. Esta es una medida de densidad que mide qué tan cerca están las soluciones vecinas en el espacio de objetivos. La fórmula para calcular esta

distancia se aprecia en la ecuación (9), donde DH_i representa la distancia de hacinamiento del individuo i , M la cantidad de funciones objetivo, $f_{i+1}^{(m)}$ y $f_{i-1}^{(m)}$ el valor del objetivo m para los vecinos adyacentes en el frente no dominado, y tanto $f_{\max}^{(m)}$ como $f_{\min}^{(m)}$ los valores máximos y mínimos del objetivo m entre todos los individuos del frente.

$$DH_i = \sum_{m=1}^M \frac{f_{i+1}^{(m)} - f_{i-1}^{(m)}}{f_{\max}^{(m)} - f_{\min}^{(m)}} \quad (9)$$

Se incluyen en la nueva población soluciones que tienen una mayor distancia de hacinamiento, ya que permiten preservar la diversidad al encontrarse más lejos de otras soluciones. Además, para asegurar la extensión del frente de Pareto, se incluyen siempre los individuos que están en los extremos.

El proceso de ordenamiento no dominado es implementado de manera eficiente mediante un sistema de rangos (estrategia elitista) para representar la dominancia. Esto significa que se calcula el número de soluciones que domina cada solución y se identifica el conjunto de soluciones que domina una solución dada. Las soluciones no dominadas son extraídas del primer frente de Pareto y se reduce la clasificación de las soluciones dominadas por las soluciones extraídas en el paso actual. El proceso se repite hasta que se seleccionen todas las nuevas soluciones necesarias. En el caso de un empate, se utiliza la distancia de hacinamiento para preferir las soluciones ubicadas en áreas más aisladas del frente de Pareto. La clasificación de las soluciones en frentes no dominados tiene una complejidad computacional de $O(M\lambda^2)$.

La infactibilidad es incorporada en el análisis de no dominancia. En este sentido, dadas dos soluciones i y j , i es preferida si (1) la solución i es factible y la solución j no lo es, (2) ambas soluciones son infactibles, pero la solución i tiene menor nivel de violación de restricciones que j , y (3) ambas soluciones son factibles e i domina a j en base a las funciones objetivo.

3.2. Representación

Se representarán las soluciones utilizando vectores binarios de tamaño D , siendo D el número total de características. El valor 0 corresponde a una característica no seleccionada y el 1 a una seleccionada. En la figura 7 se aprecia visualmente un ejemplo de solución donde «Car.» significa «Característica».

3.3. Funciones Objetivo

Las ecuaciones (10) y (11) presentan las dos funciones objetivo implementadas. En la ecuación (10), C corresponde al número total de clases, TP_i representa los valores que son ver-

1	0	1	1	...	1
Car. 1	Car. 2	Car. 3	Car. 4		Car. D

Figura 7: Ejemplo de solución utilizando la representación.

Fuente: Elaboración propia.

daderos positivos en la clase C , y S_i representa todos los valores en la clase C . Por otro lado, en la ecuación (11), $\#características_seleccionadas$ corresponde al número de características utilizadas.

$$F_1 = \frac{\sum_{i=1}^C TP_i}{\sum_{i=1}^C S_i} \quad (10)$$

$$F_2 = \#características_seleccionadas \quad (11)$$

3.4. Inicialización

En la propuesta se presentan tres métodos de inicialización. La inicialización que considera la generación de soluciones iniciales aleatorias, la inicialización con chi-cuadrado y la inicialización que mezcla ambas. Los procesos se explican en detalle a continuación.

3.4.1. Inicialización base

La inicialización del código base de NSGA-II genera soluciones binarias aleatorias. En este caso, para cada solución, se establece un $h\%$ de probabilidad de seleccionar una característica. Cabe resaltar que el algoritmo original de NSGA-II utiliza un 50 % de probabilidad [Deb et al., 2002].

3.4.2. Inicialización con chi-cuadrado

Con el fin de mejorar la inicialización de la población en el algoritmo NSGA-II, se implementa una estrategia basada en la prueba estadística chi-cuadrado (χ^2) [Jin et al., 2006]. Esta técnica permite identificar las características que tienen una mayor dependencia con la variable objetivo, lo que resulta útil para reducir la cantidad de características sin afectar la precisión del modelo. Esta prueba se presenta en más detalle en la sección 2.1.1. En el algoritmo 2 se presenta el proceso de inicialización, donde n corresponde a la cantidad de intervalos en

los que se discretiza la data, j el porcentaje de características que se seleccionarán, $prob1$ la probabilidad de mutar un bit, $percent$ el porcentaje de características con mejores valores de chi-cuadrado y P la población inicial.

Para comenzar, en la línea 1 el algoritmo discretiza las características X en n intervalos. Esto convierte las características continuas en valores discretos, lo que permite asegurar el funcionamiento de chi-cuadrado. Luego, en la línea 2, se calculan los puntajes de la prueba de χ^2 para cada característica, los cuales son normalizados para facilitar la selección. En la línea 3 se selecciona el $j\%$ de características con mayores puntajes de χ^2 , eligiendo así las características más relevantes. En la línea 4 se crea un individuo que contiene las características seleccionadas anteriormente y en la línea 5 se genera un vector con los índices de las características ordenadas de mayor a menor valor de χ^2 .

En la línea 6 se inicializa el primer individuo de la población con M_0 . Posteriormente, en las líneas 7 y 8, se calcula hasta qué número de características se pueden realizar las mutaciones en M_0 y la cantidad de veces que se mutará un bit en cada individuo. Los valores obtenidos en las dos líneas anteriores se utilizan en el ciclo que se presenta entre la línea 9 y la 12, donde la idea es generar un bucle para tener individuos distintos en la población. Para esto, lo primero que se hace es copiar el primer individuo a otro en la línea 10. Posteriormente, en la línea 11, se varían los individuos mutando un total de var bits, los cuales se eligen aleatoriamente, cumpliendo con que no se repitan las características mutadas y con limitar las posibles características seleccionadas del vector M en base a la variable $limite$. Una vez que se termina el ciclo, se retorna la población inicial en la línea 13.

Algorithm 2 Inicialización basada en Chi-cuadrado

Require: $X, c0, n, j, D, \lambda, prob, percent$

Ensure: Población inicial P

```

1:  $X_{Disc} = \text{Discretizar la data}(X, n)$ 
2:  $Chi = \text{Calcular y normalizar los puntajes } \chi^2(X_{Disc}, y)$ 
3:  $Chi_j = \text{Seleccionar las características con mayor puntaje}(Chi, j)$ 
4:  $M_0 = \text{Crear vector binario con las características seleccionadas}(Chi_j)$ 
5:  $M = \text{Crear un vector con los índices de las características de la mejor a la peor}(Chi_j)$ 
6:  $P_0 = M_0$ 
7:  $limite = D * percent$ 
8:  $var = \text{Calcular la cantidad de veces que se mutará un bit}(limite, prob)$ 
9: for  $i = 1$  to  $\lambda - 1$  do
10:    $P_i = P_0$ 
11:    $P_i = \text{Mutar utilizando posiciones aleatorias y distintas}(var, limite, M)$ 
12: end for
13: return  $P$ 

```

3.4.3. Inicialización híbrida

Con el fin de probar una mezcla entre las dos inicializaciones mencionadas anteriormente, se propone un enfoque híbrido. Este es similar a la inicialización de chi-cuadrado al inicio, con la diferencia de que considera únicamente un $j/2\%$ de características. El otro $j/2\%$ se genera aleatoriamente para cada individuo de la población. Esto permite utilizar las características más relevantes junto a otras que podrían aportar aún más valor al utilizarlas en conjunto.

En el algoritmo 3 se encuentra el pseudocódigo de la inicialización híbrida. En este, como se mencionó previamente, de la línea 1 a la línea 5 se realiza el mismo procedimiento que en el algoritmo 2. De la línea 6 a la línea 9 se realiza un ciclo donde se asigna a cada individuo el vector binario M_0 , en la línea 7, y luego, en la línea 8, se eligen aleatoriamente $j/2\%$ de las características, sin mutar ninguna de las características seleccionadas por chi-cuadrado.

Algorithm 3 Inicialización híbrida

Require: $X, c0, n, j, D, \lambda$

Ensure: Población inicial P

```
1:  $X_{Disc}$  = Discretizar la data( $X, n$ )
2:  $Chi$  = Calcular y normalizar los puntajes  $\chi^2(X_{Disc}, y)$ 
3:  $Chi_j$  = Seleccionar las características con mayor puntaje( $Chi, j/2$ )
4:  $M_0$  = Crear vector binario con las características seleccionadas( $Chi_j$ )
5:  $M$  = Crear un vector con los índices de las características de la mejor a la peor( $Chi_j$ )
6: for  $i = 0$  to  $\lambda$  do
7:    $P_i = M_0$ 
8:    $P_i$  = Seleccionar nuevas características que no hayan sido seleccionadas( $j/2$ )
9: end for
10: return  $P$ 
```

3.5. Operadores de transformación

3.5.1. Operador de selección

Para la selección se utiliza torneo binario empleando como criterio el operador de comparación por hacinamiento, tal como en [Deb *et al.*, 2002]. Este operador compara dos individuos en base a su rango de no dominancia, priorizando aquellos con menor rango. En caso de un empate, se selecciona el individuo que posea una mayor distancia de hacinamiento. Si también empatan en esta medida, se selecciona uno de los individuos al azar. Como se puede observar en el algoritmo 4, se realizan dos torneos en cada paso. Esta estrategia busca aumentar la diversidad en la selección de los padres y evitar redundancias en los emparejamientos, reduciendo así la probabilidad de escoger a los mismos individuos en una misma generación y mejorando la exploración en el espacio de búsqueda. Luego de seleccionar a

dos individuos en cada torneo, se aplica el operador de cruzamiento las parejas para generar dos descendientes. La aplicación del proceso de selección de padres y cruzamiento requiere siempre poblaciones de tamaño múltiplo de cuatro.

Algorithm 4 Selección por torneo binario con cruzamiento

Require: Población actual P de tamaño λ

Ensure: Nueva población Q de tamaño λ

- 1: Generar dos permutaciones aleatorias A_1 y A_2 de $[0, \lambda - 1]$
 - 2: **for** $i = 0$ **to** $\lambda - 1$ **step** 4 **do**
 - 3: Seleccionar p_1, p_2 mediante Torneo sobre $P[A_1[i:i+3]]$
 - 4: Generar $Q[i], Q[i + 1]$ mediante Cruzamiento(p_1, p_2)
 - 5: Seleccionar p_3, p_4 mediante Torneo sobre $P[A_2[i:i+3]]$
 - 6: Generar $Q[i + 2], Q[i + 3]$ mediante Cruzamiento(p_3, p_4)
 - 7: **end for**
-

3.5.2. Operador de cruzamiento

Para el cruzamiento se implementa un cruce de dos puntos, en el cual se seleccionan dos posiciones aleatorias del individuo y se intercambia el segmento intermedio que forman los puntos entre ambos padres. De esta forma, se genera una mayor diversidad en los hijos y se explora más el espacio de búsqueda a diferencia del operador de cruzamiento en un punto, que es el cruzamiento propuesto para representaciones binarias en [Deb *et al.*, 2002]. Este es un operador clásico para representación binaria.

En el algoritmo 5 se muestra la implementación, donde p_1 y p_2 corresponden a los padres, c_1 y c_2 corresponden a los hijos, y tanto $site_1$ como $site_2$ corresponden a las posiciones de corte aleatorias.

Algorithm 5 Cruzamiento binario de dos puntos

Require: p_1, p_2, p_c, D

Ensure: c_1, c_2

- 1: **if** random(0,1) $\leq p_c$ **then**
 - 2: Seleccionar dos posiciones aleatorias $site_1, site_2$ en $[0, D-1]$, con $site_1 < site_2$
 - 3: Copiar genes desde $[0, site_1)$ y $[site_2, D)$ de los padres a los hijos
 - 4: Intercambiar los genes del segmento $[site_1, site_2)$ entre los padres
 - 5: **else**
 - 6: Copiar todos los genes de p_1 y p_2 a c_1 y c_2 sin cambios
 - 7: **end if**
-

3.5.3. Operador de mutación

Se implementa una mutación bit-flip, tal como en [Deb *et al.*, 2002], donde cada bit corresponde a una característica. Este operador recorre cada una de las características de un individuo e invierte su valor con una probabilidad de p_m . Esta mutación se realiza de forma independiente a cada posición del vector binario, lo que permite modificar más de una característica por individuo. Este operador permite variar la población y evitar que el algoritmo se estanque muy rápidamente en óptimos locales [Adoptante *et al.*, 2023].

En el algoritmo 6 se muestra la implementación, donde *ind* corresponde al individuo. Este procedimiento se aplica para cada solución de la población.

Algorithm 6 Mutación bit-flip

Require: *ind*, p_m

```
1: for  $j = 0$  to  $D - 1$  do
2:   if  $\text{random}(0,1) \leq p_m$  then
3:     Invertir  $\text{ind}[j]$ :  $1 \leftrightarrow 0$ 
4:   end if
5: end for
```

VALIDACIÓN DE LA SOLUCIÓN

En este capítulo se presentan los conjuntos de datos utilizados para la validación de la propuesta. Se detalla el algoritmo de predicción utilizado y se muestran los resultados obtenidos. Por último, se realizan comparaciones tanto entre el algoritmo propuesto y el algoritmo base, como con el algoritmo CNSGA-II [Miyandoab *et al.*, 2023], para así realizar un análisis en profundidad de los resultados.

4.1. Conjuntos de datos

Se utilizaron seis conjuntos de datos para los experimentos, los cuales se listan en la tabla 2. Para cada conjunto se presenta el nombre, cantidad de características, cantidad de muestras y cantidad de clases.

Tabla 2: Conjuntos de datos

Conjunto de datos	Características (D)	Muestras	Clases (N)
DLBCL	5,469	77	2
Lung Cancer	12,600	203	5
warpAR10P	2,400	130	10
warpPIE10P	2,420	210	10
Penbased	16	10,992	10
Texture	40	5,500	11

Los primeros dos conjuntos provienen del área de bioinformática médica, principalmente del diagnóstico de cáncer a través perfiles de expresión génica obtenidos mediante microarrays. Los datos corresponden a muestras humanas, donde el primero corresponde a un tipo de cáncer de la sangre y el segundo a cáncer de pulmón. Estos conjuntos fueron utilizados en [Statnikov *et al.*, 2004] y se encuentran disponibles en <https://github.com/primekangkang/Genedata>.

Los conjuntos *warpAR10P* provienen del área de reconocimiento de imágenes. Estos conjuntos de datos fueron utilizados en [Miyandoab *et al.*, 2023] y se encuentran disponibles en [Zhao *et al.*, 2010b].

Los últimos dos conjuntos de datos abarcan dos áreas diferentes. El conjunto *penbased* está relacionado con el reconocimiento de escritura a mano, dado que contiene las coordenadas de un lápiz digital para reconocer dígitos manuscritos. Por otro lado, el conjunto *texture* se puede clasificar en el área de visión artificial, ya que realiza distinciones entre tipos de texturas a través de características obtenidas por patrones visuales. Ambos fueron utilizados en [Gong *et al.*, 2024] y se encuentran disponibles en el *KEEL Dataset Repository*.

Estos conjuntos de datos incluyen clases binarias y multi-clase. El número de clases va desde las 2 hasta las 11, el número de muestras va desde las 77 hasta las 10,992 y el número de características va desde las 16 hasta las 12,600.

Existen dos razones para utilizar estos conjuntos de datos. La primera es verificar que el algoritmo propuesto es capaz de abordar el problema de selección de características en diversos contextos de datos, demostrando así la capacidad de generalización que posee. La segunda razón consiste en demostrar que el algoritmo funciona tanto en contextos con una gran cantidad de características y pocas muestras como en casos con pocas características y muchas muestras.

4.2. Configuraciones experimentales

Para evaluar el desempeño de los algoritmos, lo primero es dividir los conjuntos de datos en dos partes. La primera corresponde al subconjunto de entrenamiento, al cual se le asigna el 80 % de los datos y se utiliza en el proceso del algoritmo, mientras que el 20 % restante se emplea como conjunto de prueba para evaluar los resultados obtenidos. Estos subconjuntos se generan de manera aleatoria en cada ejecución del algoritmo, asegurando que exista variedad en los resultados. En la tabla 3 se presentan los valores utilizados para los parámetros de NSGA-II, los cuales incluyen las mismas probabilidades de cruzamiento y mutación que en [Deb *et al.*, 2002]. Cada algoritmo se ejecuta 10 veces por conjunto de datos y 10 veces por porcentaje de características seleccionadas, lo que equivale a un total de 380 ejecuciones entre las tres inicializaciones. A través de experimentos preliminares, se eligió utilizar un tamaño de población de 20 individuos para reducir los costos computacionales por iteración y se usaron 500 generaciones para realizar un esfuerzo total equivalente a lo presentado en la literatura [Miyandoab *et al.*, 2023].

Parámetro	Valor
Tamaño de la población (λ)	20
Probabilidad de cruzamiento (p_c)	0.9
Probabilidad de mutación (p_m)	$1/D$
Número de generaciones (MaxGen)	500

Tabla 3: Parámetros de NSGA-II.

En la tabla 4 se presentan los valores utilizados para los parámetros de la inicialización aleatoria y las inicializaciones basadas en chi-cuadrado. Se utiliza dos valores de h y j para ver qué ocurre cuando se comienza con un 10 % y un 50 % de características utilizando cada una de las inicializaciones. El valor de n corresponde al valor por defecto de la función *KBinsDiscretizer* [Scikit-learn, 2024]. El valor *prob* tiene dos valores, dado que se escoge la cantidad de características que se van a mutar según la cantidad total que tenga cada conjunto de datos. Si es que un conjunto tiene menos de 1000 características, se utiliza una probabilidad del 10 %, y en caso contrario se utiliza un 0.1 %. Esto se realizó para mantener una cantidad inicial pareja entre las tres inicializaciones. Por último, el valor de *select*

siempre es del 50 % para mantener el mismo subconjunto de características relevantes en la inicialización de chi-cuadrado tanto con el 10 % como el 50 % de características seleccionadas.

Parámetro	Valor
Porcentaje de características (h)	0.1 y 0.5
Porcentaje de características (j)	0.1 y 0.5
Cantidad de intervalos (n)	5
Probabilidad de mutar un bit ($prob$)	0.1 o 0.001
Porcentaje de características con mejores valores de χ^2 ($percent$)	0.5

Tabla 4: Parámetros utilizados en las inicializaciones.

Para comparar el desempeño de los algoritmos multi-objetivo se utiliza el indicador de hipervolumen como métrica de evaluación, el cual fue descrito en la sección 1.2. Se utilizó el código presentado e implementado en [Fonseca *et al.*, 2006, Beume *et al.*, 2009]. Además, se presentan frentes de Pareto para cada inicialización, donde las soluciones fueron obtenidas al generar un nuevo frente no dominado entre las soluciones obtenidas en las 10 ejecuciones por conjunto de datos. Este frente se denomina frente agregado.

Para hacer más sencillo el cálculo, se adaptará la segunda función objetivo para que corresponda al porcentaje de características seleccionadas. De esta forma, es posible utilizar el punto de referencia (1, 1) para todos los conjuntos de datos. Este punto se estableció para realizar la comparación tanto de la media como la desviación estándar del hipervolumen obtenido en [Miyandoab *et al.*, 2023] para el conjunto *warpAR10P* y *warpPIE10P*, asumiendo que se utilizó el mismo punto de referencia ya que no se menciona en el paper y utilizando los resultados obtenidos con el 50 % de características iniciales.

4.3. Algoritmo de predicción

Para evaluar el acercamiento propuesto, se utiliza el algoritmo k-NN, el cual está detallado en la sección 2.1.2 , junto con Stratified K-Fold Cross-Validation (SKCV) [Prusty *et al.*, 2022]. El procedimiento comienza al recibir el conjunto de datos, donde se utiliza SKCV para dividir los datos en K pliegues que contienen la misma proporción de clases y luego se comienzan a dividir los subconjuntos en entrenamiento y prueba. En cada iteración , uno de los pliegues se utiliza como conjunto de prueba, mientras que los $K - 1$ restantes se utilizan para el entrenamiento. Se calcula la distancia euclidiana entre los nuevos puntos de los pliegues de prueba con los k vecinos más cercanos. Esta distancia se calcula tal como se observa en la ecuación (12):

$$d(x, y) = \sqrt{\sum_{i=1}^M (x_i - y_i)^2} \quad (12)$$

donde x representa los valores de una muestra del subconjunto de prueba, y los valores de una muestra del subconjunto de entrenamiento y cada x_i e y_i corresponde al valor de la característica i en cada muestra. Cabe resaltar que el valor de k asociado a k-NN es distinto al valor de K asociado a SKCV. Para k-NN se utiliza $k = 3$ al igual que en [Xue *et al.*, 2023].

Se realizaron unas variaciones al algoritmo de KNN debido a problemas presentados en experimentos preliminares utilizando los conjuntos de datos. La tabla 5 presenta información más detallada sobre cada conjunto de datos. El porcentaje de clase más pequeña se calcula teniendo en cuenta el número de muestras en las que aparece la clase menos representada en los conjuntos de datos. Debido a esto, al realizar la división del conjunto en datos en un subconjunto de entrenamiento y uno de pruebas, pueden haber problemas a la hora de realizar las evaluaciones. Estos problemas se deben a que, al utilizar Stratified K-Fold Cross-Validation, se limita el número de pliegues posibles en base a la clase con menor cantidad de muestras. Si una clase tiene solo una muestra, no es posible generar más de un pliegue para dividir entre conjuntos de entrenamiento y prueba, lo que impide realizar correctamente la evaluación.

Tabla 5: Información relevante de los conjuntos de datos.

Conjunto de datos	%Clase más pequeña	%Clase más grande	Errores en Pruebas
DLBCL	24.68	75.32	1
Lung Cancer	2.96	68.47	5
warpAR10P	10.00	10.00	8
warpPIE10P	10.00	10.00	6
Penbased	9.60	10.41	0
Texture	9.09	9.09	0

Para corroborar este problema y revisar cuántas veces se presenta la situación durante los experimentos se realizaron 10 ejecuciones por cada conjunto de datos utilizando el código base de NSGA-II. Los resultados obtenidos en los subconjuntos de pruebas se muestran en la última columna de la tabla 5, donde se puede apreciar que 4 de los 6 conjuntos de datos presentaban problemas.

Por esta razón es que se modifican los conjuntos de datos eliminando las clases que contengan solo una muestra. De esta manera se asegura que 5 de los 6 conjuntos de datos no presenten problemas durante el funcionamiento del algoritmo. El conjunto restante corresponde a *DLBCL*. Este presentó problemas en el subconjunto de pruebas debido a que en una de cada diez ejecuciones, se generaban conjuntos que contenían una sola clase luego de la eliminación de las clases con una sola muestra. Esto mismo provoca que solo se genere un pliegue y, por ende, no se pueda realizar el entrenamiento, dado que el único pliegue se asigna como subconjunto de pruebas. Considerando el hecho de que es muy poco probable que ocurra este problema, se ha decidido realizar los experimentos y, en caso de que una de las ejecuciones presente un error, simplemente se repetirá el experimento para ese caso en particular.

RESULTADOS

En este capítulo se presentan los resultados obtenidos al utilizar NSGA-II con las cada una de las inicializaciones y se realizan tanto comparaciones como análisis entre estas. Se presentan primero los resultados considerando 10 % de características iniciales y posteriormente los resultados considerando 50 % de características iniciales. Para cada caso se revisan los errores y la cantidad de características obtenidas por cada versión del algoritmo, los frentes de Pareto obtenidos y los hipervolumenes. Además, se realiza una comparación con el algoritmo CNSGA-II.

5.1. Resultados con el 10 % de características iniciales

5.1.1. Error y cantidad de características

En las tablas 6, 7 y 8 se presentan los resultados obtenidos utilizando un h y j de 0.1.

En la tabla 6 se encuentra el error porcentual en el proceso de entrenamiento.

Al observar los valores mínimos de error, se aprecia que la inicialización basada en chi-cuadrado logra los menores errores individuales en 4 de las 6 instancias, lo que respalda su capacidad para generar soluciones de alta calidad desde un comienzo. Eso sí, al observar la implementación aleatoria e híbrida, se ve que obtienen buenos mínimos, pero en *warpAR10P* les va bastante peor. Además, si bien se observan buenos mínimos, se puede considerar que estos no son tan relevantes, ya que se observa que existe una gran diferencia, en muchas instancias, entre el mínimo y la media.

Ahora bien, al fijarse en la media del error, se puede observar que la inicialización de chi-cuadrado obtuvo un menor promedio de error en 4 de los 6 casos. Esto se atribuye nuevamente a su elitismo, ya que permite asegurar buenos resultados desde un comienzo utilizando características relevantes. Al observar la columna de los valores mínimos se evidencia que la inicialización de chi-cuadrado tiende a alcanzar los menores errores individuales en varias instancias, como en *warpAR10P* y *warpPIE10P*. Esto refuerza aún más carácter elitista. En el caso de la implementación aleatoria, se puede observar que en general obtiene excelentes mínimos, a excepción del caso de *warpAR10P*, donde se puede ver que obtuvo los peores resultados. Esto se atribuye a que el conjunto debe tener muchas características irrelevantes y redundantes, lo que provoca que sea poco probable encontrar buenos resultados con esta inicialización. Esto se consolida al observar los resultados en las otras dos implementaciones, donde la que obtuvo mejores resultados fue la única que no aplica aleatoriedad en su proceso. Por último, se puede apreciar que los resultados de las últimas dos instancias son muy parejos para las tres inicializaciones, lo que tiene relación con el hecho de ser los conjuntos con poca cantidad de características y muchas muestras, ya que son

conjuntos menos desafiantes.

Es importante observar es que la inicialización híbrida que se propuso obtuvo una mayor desviación estándar en la mitad de las instancias, siendo estas las que se encuentran en negrita. Esto quiere decir que, para esos casos, la implementación explora mejor el espacio de soluciones con respecto al error. Sin embargo, el hecho de que se tenga un promedio de error muy alto junto con una gran desviación estándar, como es el caso de *warpAR10P* en la inicialización aleatoria, indica que se encuentran muchas soluciones con mucho error. El hecho de que Chi-cuadrado solamente haya logrado una mejor desviación estándar en 1 de los 6 casos es debido a que es muy elitista, lo que provoca que sea más difícil encontrar aquellas características que brinden nuevas soluciones en conjunto. Por esta misma razón es que tiene sentido que la implementación híbrida tenga mejores resultados, ya que permite que NSGA-II realice una mejor exploración al tener un porcentaje de aleatoriedad.

Iniciación Instancia	Aleatoria		Chi-cuadrado		Híbrida	
	Mínimo	Media $\pm$ DE	Mínimo	Media $\pm$ DE	Mínimo	Media $\pm$ DE
DLBCL	0.00	1.76 $\pm$ 0.89	0.00	2.86 $\pm$ 1.59	0.00	0.46 $\pm$ 0.87
Lung_Cancer	0.00	2.84 $\pm$ 2.41	3.68	5.11 $\pm$ 0.91	3.72	4.92 $\pm$ 0.81
warpAR10P	22.10	32.67 $\pm$ 4.86	5.00	14.32 $\pm$ 3.50	12.39	18.87 $\pm$ 3.45
warpPIE10P	1.30	2.89 $\pm$ 1.02	0.00	2.23 $\pm$ 0.82	1.09	2.69 $\pm$ 1.06
penbased	0.70	13.56 $\pm$ 19.55	0.60	13.22 $\pm$ 19.00	0.61	14.42 $\pm$ 20.07
texture	0.67	9.72 $\pm$ 15.82	0.76	9.72 $\pm$ 15.53	0.80	11.66 $\pm$ 18.12

Tabla 6: Resultados del error porcentual en el proceso de entrenamiento - 10 % de características iniciales.

En la tabla 7 se muestra el error porcentual en el proceso de prueba. En cuanto a los valores mínimos alcanzados, se observa que la estrategia de chi-cuadrado logra los menores mínimos en 3 de las 6 instancias. Este comportamiento refuerza el carácter elitista de esta estrategia. Por otro lado, la inicialización aleatoria, si bien también consigue excelentes mínimos en varias instancias, vuelve a fallar en *warpAR10P*, posiblemente debido a la presencia de muchas características irrelevantes que dificultan la búsqueda eficiente. Se vuelve a observar que los mínimos obtenidos están bastante alejados de la media en general.

Con respecto a la media del error se observa lo mismo que en la tabla 6, donde chi-cuadrado obtuvo un mejor promedio en 5 de 6 instancias. Al observar la columna de mínimos, se puede notar que la implementación aleatoria obtuvo excelentes mínimos, a excepción de la instancia *warpAR10P*. Se atribuye la misma lógica mencionada en la tabla 6, junto con el hecho de que chi-cuadrado llega a obtener los menores mínimos en 4 de 6 instancias y que se llegan a resultados similares en los últimos dos conjuntos de datos.

Al igual que en la tabla 6, se puede observar es que existen diferencias notables en la desviación estándar y la media del error entre las distintas estrategias de inicialización. En este caso, la estrategia aleatoria obtiene la mayor desviación estándar en 4 de las 6 instancias, lo

que sugiere que en estos conjuntos existe una mayor dispersión de resultados al aplicar esta estrategia, posiblemente debido a una exploración más amplia del espacio de búsqueda, aunque no siempre con buenos resultados. Por otro lado, la inicialización híbrida presenta la mayor desviación en dos instancias, mostrando que también es capaz de cubrir una mayor variedad de soluciones en el espacio de prueba. Un punto interesante es que, al igual que lo que se pudo apreciar en el entrenamiento, la estrategia chi-cuadrado no destaca en tener los mejores valores de desviación estándar, de hecho no obtuvo mayor desviación estándar en ninguna instancia.

Iniciación	Aleatoria		Chi-cuadrado		Híbrida	
	Mínimo	Media $\pm$ DE	Mínimo	Media $\pm$ DE	Mínimo	Media $\pm$ DE
DLBCL	0.00	10.13 $\pm$ 6.54	6.25	11.07 $\pm$ 4.79	14.29	17.08 $\pm$ 4.39
Lung Cancer	2.56	15.14 $\pm$ 9.69	7.26	11.05 $\pm$ 3.46	5.41	16.21 $\pm$ 7.53
warpAR10P	62.14	71.19 $\pm$ 4.96	37.54	50.22 $\pm$ 7.68	39.58	58.88 $\pm$ 9.63
warpPIE10P	39.39	65.54 $\pm$ 11.28	37.07	53.73 $\pm$ 8.59	36.62	53.75 $\pm$ 7.29
Penbased	1.48	15.97 $\pm$ 19.81	1.46	15.73 $\pm$ 19.44	1.55	16.12 $\pm$ 19.54
Texture	2.49	11.60 $\pm$ 14.26	2.36	11.36 $\pm$ 14.59	2.49	12.63 $\pm$ 16.15

Tabla 7: Resultados del error porcentual en el proceso de prueba - 10 % de características iniciales.

En la tabla 8 se presenta la cantidad de características seleccionadas en las soluciones obtenidas. Esta métrica refleja directamente qué tan efectiva fue la búsqueda multi-objetivo en reducir la dimensionalidad del problema.

Iniciación	Aleatoria		Chi-cuadrado		Híbrida	
	Mínimo	Media $\pm$ DE	Mínimo	Media $\pm$ DE	Mínimo	Media $\pm$ DE
DLBCL	270	295.83 $\pm$ 12.85	329	333.92 $\pm$ 5.37	265	288.03 $\pm$ 16.54
Lung_Cancer	932	1051.47 $\pm$ 120.06	1027	1032.25 $\pm$ 3.73	930	955.36 $\pm$ 16.47
warpAR10P	87	111.88 $\pm$ 15.43	115	134.10 $\pm$ 25.90	112	128.02 $\pm$ 8.94
warpPIE10P	68	93.34 $\pm$ 11.47	73	91.47 $\pm$ 9.59	58	76.52 $\pm$ 6.83
penbased	1	6.45 $\pm$ 3.56	1	6.79 $\pm$ 3.89	1	6.58 $\pm$ 3.93
texture	1	7.47 $\pm$ 4.82	1	6.96 $\pm$ 4.37	1	6.49 $\pm$ 4.18

Tabla 8: Resultados de la cantidad de características - 10 % de características iniciales.

Lo más notorio con respecto a los mínimos es que los valores en *penbased* y *texture* para todas las inicializaciones son de 1. Esto da a entender que, para casos con pocas características, los tres algoritmos permiten que NSGA-II logre llegar a los resultados con menores características. Con respecto al resto de resultados, se puede observar que chi-cuadrado es quien obtuvo los mayores mínimos, lo cual refuerza la idea de que no es una buena implementación para explorar zonas del espacio de búsqueda, limitando así su capacidad de generar

nuevas soluciones. A diferencia de esta última, la inicialización híbrida alcanzó mejores mínimos en 3 de las primeras 4 instancias, lo cual se puede atribuir al agregado de la aleatoriedad junto con la selección de características relevantes con chi-cuadrado.

Siguiendo con la media de las características, se puede observar que en 4 de las 6 instancias se obtuvo un menor promedio con la implementación híbrida. Esto da a entender que esta realiza un mejor trabajo a la hora de reducir la cantidad de características, sobre todo considerando que tanto la inicialización aleatoria como la de chi-cuadrado obtuvieron la peor media en tres instancias cada uno. Se puede atribuir esto al hecho de que encuentra mejores soluciones gracias a las características relevantes que fueron seleccionadas por chi-cuadrado y logra explorar más gracias a la parte aleatoria que tiene. También se observa que, pese a que algunas implementaciones obtuvieron mejores resultados que otras, se mantienen resultados bastante parejos en las últimas dos instancias.

Por otro lado se puede apreciar que la implementación aleatoria tiene mayor desviación estándar en la mitad de los conjuntos de datos. Incluso, se puede observar que en *Lung_Cancer* hay una diferencia muy grande entre la desviación de la implementación aleatoria con respecto a las otras dos. Esto se puede atribuir al hecho de que la inicialización aleatoria explora más el espacio de búsqueda, además de que esa instancia es la que tiene mayor cantidad de características. Lo otro interesante es que se obtuvo una diferencia bastante mayor en *warpAR10P*, siendo chi-cuadrado el mayor. Esto se puede justificar debido a que, al quedarse atrapado con mayor frecuencia en óptimos locales, las nuevas soluciones que encuentra y que forman parte del frente de Pareto tienen una mayor cantidad de características, lo que tiene sentido al observar el promedio de características en ese caso.

5.1.2. Frentes de Pareto

En las figuras 8 y 9 se presentan los frentes de Pareto obtenidos en el proceso de entrenamiento y el proceso de prueba, donde cada figura representa una instancia.

En la figura 8(a) se observa que la estrategia híbrida logra los mejores resultados, con un punto que es claramente dominante al tener la menor cantidad de características con un error igual a cero. Las soluciones generadas con la inicialización de chi-cuadrado están muy por encima en cantidad de características, mientras que las aleatorias se concentran cerca de las 270-280. Esto refuerza la idea de que la inicialización híbrida logra frentes más compactos sin sacrificar rendimiento.

En la figura 8(b) la implementación aleatoria es la que obtiene la mayor cantidad de puntos del frente de Pareto, siendo la única que llega a un 0 % de error. Con respecto a la inicialización híbrida, solamente una solución evita ser dominada por el frente aleatorio. En este caso es particular, la aleatoriedad parece haber sido suficiente para encontrar buenos subconjuntos, posiblemente porque este conjunto de datos requiere de una mayor exploración para

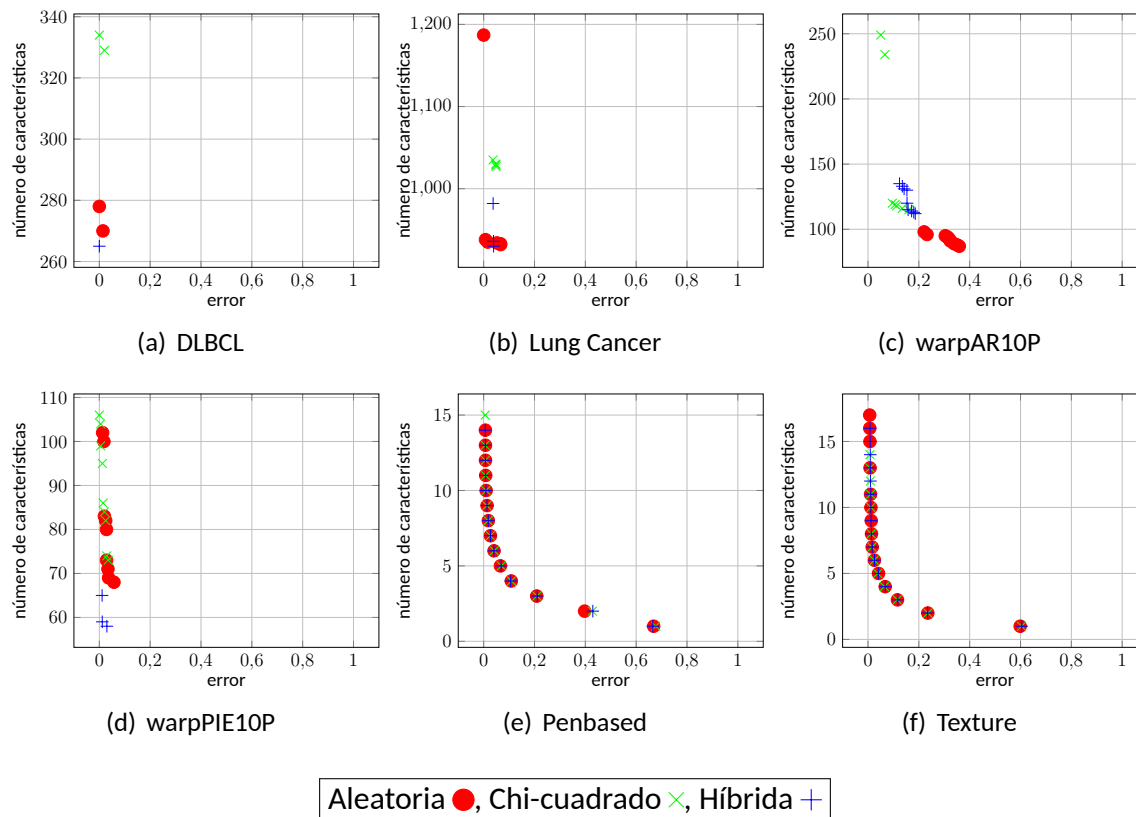


Figura 8: Frentes de Pareto del proceso de entrenamiento - 10 % de características iniciales.

encontrar mejores resultados al ser el que posee mayor cantidad de características. Esto último también se justifica al observar que la implementación de chi-cuadrado está dominada por completo.

En la subfigura 8(c) se presenta un caso donde ninguna de las inicializaciones es claramente mejor, ya que se puede observar que todas poseen puntos que forman parte de un frente de Pareto en conjunto. Si hablamos en términos del error, la implementación de chi-cuadrado obtiene los mejores resultados. Pero si hablamos con respecto a la cantidad de características, podemos observar que claramente la inicialización aleatoria es la mejor. Esto es consistente con lo señalado en las tablas 6, 7 y 8. Ahora bien, si hablamos sobre una buena proporción, se puede considerar que la implementación híbrida fue la mejor. Esto se debe a que fue capaz de obtener bajos errores teniendo una baja cantidad de características, lo que no quita el hecho de que la mayoría de sus puntos se ven opacados por la inicialización de chi-cuadrado.

En la figura 8(d), la estrategia híbrida domina el frente, alcanzando soluciones con menor cantidad de características y con bajo error. Aunque la implementación aleatoria y la de chi-cuadrado también logran buenos resultados y se encuentran más repartidas en el frente, sus soluciones se ven dominadas por la híbrida. Esto valida lo bueno de elegir un porcentaje

de características relevantes y otro aleatorio. Existen unos puntos de la inicialización de chi-cuadrado que no se ven dominados, pero la diferencia entre cantidad de características a comparación del error es bastante mayor.

Con respecto a las figuras 8(e) y 8(f), podemos observar que todas las inicializaciones llegan a frentes de Pareto casi iguales. Esto demuestra que las dos inicializaciones propuestas en este trabajo son capaces de identificar los frentes en conjuntos con poca cantidad de características. Cabe aclarar que todos los frentes tienen la misma cantidad de soluciones en *Penbased*, pero no en *Texture*. En el frente aleatorio se obtienen 15 soluciones no dominadas, en el de chi-cuadrado se obtienen 13 y en el de la implementación híbrida se obtienen 15.

Ahora bien, con respecto a la figura 9, se puede observar que los últimos tres gráficos presentan casos similares a los últimos tres de la figura 8. Sin embargo, las primeras tres presentan diferencias notorias.

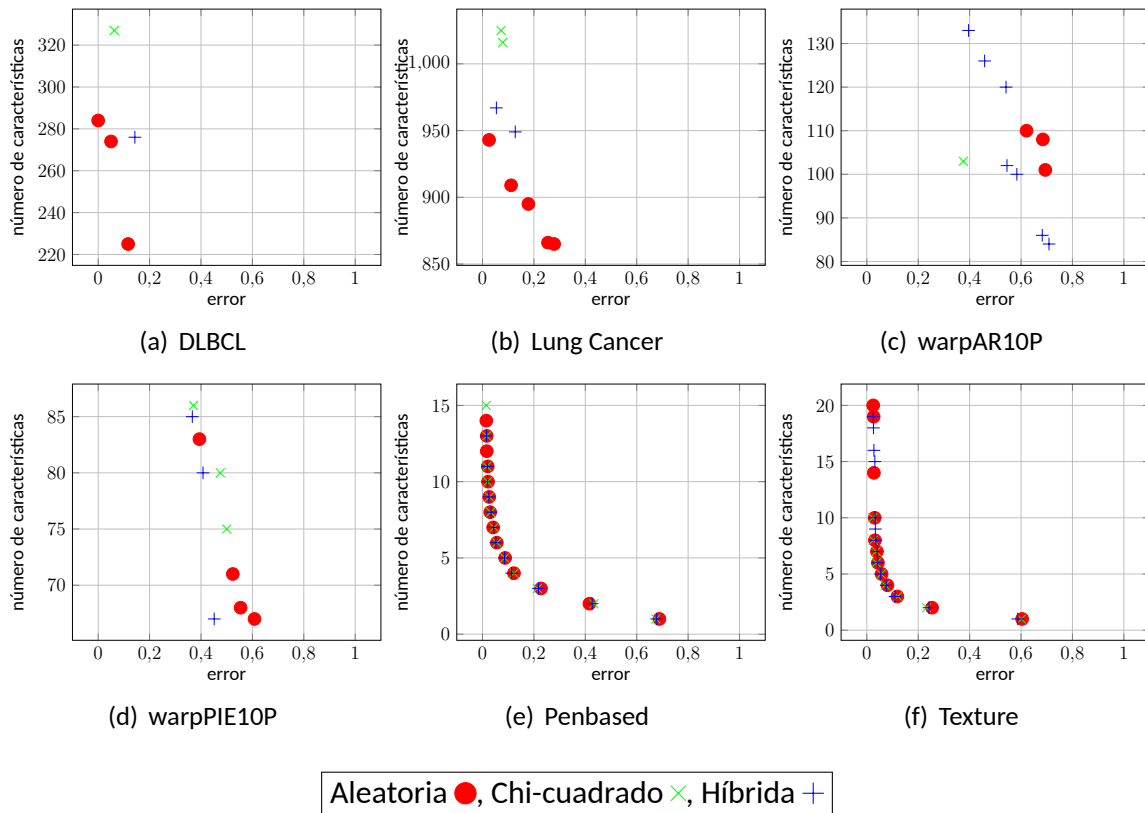


Figura 9: Frentes de Pareto del proceso de prueba - 10 % de características iniciales.

Tanto en la subfigura 9(a) como en 9(b) la inicialización aleatoria dominó por completo a las otras dos implementaciones. Esto da a entender que, para estos casos, la estrategia aleatoria tiene una mayor capacidad de generalización en comparación con el resto. Además, en la figura 9(c) se puede observar que ahora la inicialización aleatoria es completamente dominada, mientras que la híbrida entrega un frente bastante disperso. Esto se puede de-

ber a que el conjunto debe presentar una mayor cantidad de características irrelevantes y redundantes, lo que hace que la inclusión de chi-cuadrado permita encontrar mejores resultados. Al ver que la inicialización de chi-cuadrado obtuvo un punto excelente en términos de proporción, se refuerza esto. Cabe aclarar que todos los frentes tienen la misma cantidad de soluciones en *Penbased*, pero no en *Texture*, al igual que en las subfiguras 8(e) y 8(f). La diferencia es que en el frente aleatorio de 9(f) se obtienen 12 soluciones no dominadas, en el de chi-cuadrado se obtienen 9 y en el de la inicialización híbrida se obtienen 13.

En resumen, los frentes de Pareto reflejan claramente cómo la elección de la estrategia de inicialización influye tanto en el rendimiento como en la capacidad de generalización de las soluciones. La estrategia aleatoria demuestra buen comportamiento en conjuntos más complejos, mientras que la híbrida ofrece un mejor balance general. Chi-cuadrado, aunque útil en algunos casos, tiende a generar soluciones menos compactas si no se complementa con exploración.

5.1.3. Hipervolumen

En las tablas 9 y 10 se encuentran los resultados de hipervolumen obtenidos, donde en negrita están marcados los promedios más altos para cada instancia.

Con respecto a la tabla 9, se puede observar que en 4 de los 6 conjuntos de datos la inicialización aleatoria obtuvo un mejor promedio en comparación con el resto. En el caso de las dos primeras instancias es más esperable, ya que en las tablas y frentes de Pareto anteriores se pudo apreciar que esta implementación, en general, obtenía mejores resultados para estos dos conjuntos. Lo mismo aplica para *warpAR10P*, donde se ve que los resultados son mucho peores con la inicialización aleatoria. Ahora bien, en base a las otras dos implementaciones, se puede apreciar que ambas estuvieron bastante parejas entre si, obteniendo el mejor resultado en 1 de los 6 conjuntos. Como tal esto indica que, aunque fueron mejores en algunos casos, la inicialización aleatoria se mantiene como una mejor opción en general para seleccionar características durante el proceso de entrenamiento.

Inicialización	Aleatoria		Chi-cuadrado		Híbrida	
Instancia	Máximo	Promedio $\pm$ DE	Máximo	Promedio $\pm$ DE	Máximo	Promedio $\pm$ DE
DLBCL	0.7717	0.7159 $\pm$ 0.0406	0.6518	0.6011 $\pm$ 0.0253	0.5882	0.5521 $\pm$ 0.0203
Lung Cancer	0.9258	0.9203 $\pm$ 0.0051	0.8841	0.8707 $\pm$ 0.0096	0.8901	0.8798 $\pm$ 0.0097
warpAR10P	0.7478	0.6638 $\pm$ 0.0480	0.8619	0.8288 $\pm$ 0.0287	0.8311	0.7817 $\pm$ 0.0329
warpPIE10P	0.9472	0.9375 $\pm$ 0.0091	0.9591	0.9444 $\pm$ 0.0074	0.9648	0.9494 $\pm$ 0.0103
Penbased	0.8360	0.8343 $\pm$ 0.0010	0.8338	0.8331 $\pm$ 0.0005	0.8344	0.8329 $\pm$ 0.0008
Texture	0.9402	0.9386 $\pm$ 0.0010	0.9401	0.9384 $\pm$ 0.0010	0.9396	0.9384 $\pm$ 0.0006

Tabla 9: Hipervolumen en el proceso de entrenamiento - 10 % de características iniciales.

Al fijarse en los máximos obtenidos, se puede ver que, aunque existan algunos casos con excelentes mínimos, no significa que siempre se lleguen a obtener esos valores. Por ejemplo, en *warpAR10P*, donde los máximos en los tres casos se alejan bastante a comparación de los promedios. Esto mismo se refleja en la desviación estándar, donde se puede apreciar que, en general, es la instancia donde se encuentran los mayores promedios. Esto último indica que los máximos fueron obtenidos más por la dispersión de los resultados que por consistencia. A diferencia de este, conjuntos como *Lung_Cancer*, *Penbased* y *Texture* demostraron ser mucho más consistentes en sus resultados, obteniendo máximos bastante cercanos al promedio que se relacionan con la baja desviación estándar obtenida.

A diferencia de la tabla 9, se puede ver que en la tabla 10 se obtienen mejores promedios en 5 de las 6 instancias utilizando la inicialización de chi-cuadrado. Esto mismo indica que, si bien para el entrenamiento se obtienen mejores frentes con una implementación aleatoria, a la hora de evaluar esto con un conjunto externo se muestra que chi-cuadrado tiene una mayor capacidad de generalización. Por otra parte, se puede observar que la implementación híbrida se mantuvo bastante bien con sus resultados para todas las instancias, a diferencia de la aleatoria, donde obtuvo muy buen promedio en todas las instancias, menos en *warpAR10P* y *warpPIE10P*.

Inicialización	Aleatoria		Chi-cuadrado		Híbrida	
Instancia	Máximo	Promedio $\pm$ DE	Máximo	Promedio $\pm$ DE	Máximo	Promedio $\pm$ DE
DLBCL	0.9481	0.8512 $\pm$ 0.0658	0.8814	0.8399 $\pm$ 0.0466	0.8138	0.7627 $\pm$ 0.0516
Lung Cancer	0.9015	0.7679 $\pm$ 0.0748	0.8520	0.7988 $\pm$ 0.0451	0.8733	0.7551 $\pm$ 0.0575
warpAR10P	0.3613	0.2853 $\pm$ 0.0565	0.5978	0.4826 $\pm$ 0.0937	0.5707	0.3954 $\pm$ 0.0984
warpPIE10P	0.5853	0.3468 $\pm$ 0.1325	0.6070	0.4918 $\pm$ 0.0863	0.6127	0.4660 $\pm$ 0.0857
Penbased	0.8204	0.8180 $\pm$ 0.0020	0.8207	0.8186 $\pm$ 0.0015	0.8238	0.8184 $\pm$ 0.0033
Texture	0.9218	0.9145 $\pm$ 0.0053	0.9245	0.9174 $\pm$ 0.0054	0.9218	0.9147 $\pm$ 0.0044

Tabla 10: Hipervolumen en el proceso de prueba - 10 % de características iniciales.

Al observar los valores máximos, se puede ver que, a excepción de las últimas dos instancias, los valores están bastante más alejados del promedio a comparación de los resultados obtenidos en la tabla 9. Esto es bastante coherente con lo que ocurrió en las tablas 6 y 7, donde se observó una mayor dispersión con los conjuntos de prueba. La principal razón detrás de este fenómeno se relaciona con la capacidad de generalización de las soluciones. Mientras que en el entrenamiento algunas estrategias logran obtener frentes de Pareto con alta dominancia, esas mismas soluciones no siempre se mantienen al ser aplicadas a datos no vistos. Esto se refleja en una mayor desviación estándar en los valores de hipervolumen en el proceso de prueba, especialmente para las estrategias aleatoria e híbrida en las primeras cuatro instancias. A causa de esto, aunque estas estrategias pueden alcanzar valores máximos altos (incluso superiores a los obtenidos en chi-cuadrado), su rendimiento promedio es menos estable, lo que puede representar un comportamiento menos robusto a la hora de obtener buenos frentes de Pareto. Por el contrario, chi-cuadrado muestra menores desviaciones es-

tándar en general, lo que refuerza su carácter elitista y su capacidad de producir soluciones iniciales con bajo porcentaje de error.

5.2. Resultados con el 50 % de características iniciales

5.2.1. Error y cantidad de características

En las tablas 11, 12 y 13 se presentan los resultados obtenidos utilizando un h y j de 0.5.

Lo primero que se puede apreciar en la tabla 11 es que se obtuvieron peores resultados en general para todas las instancias a comparación de los resultados vistos en 6 en base al mínimo y la media. Esto tiene bastante sentido, dado que, al utilizar una mayor cantidad de características en la inicialización, se incluyen más características irrelevantes y redundantes, lo que provoca que los errores obtenidos sean peores. Ahora bien, con respecto a la desviación estándar, se puede observar que ahora es chi-cuadrado el que tiene mayores desviaciones estándar. Esto puede ser debido a que, al elegirse una mayor cantidad de características iniciales, se explora mejor el espacio de búsqueda a través del procedimiento de NSGA-II.

Inicialización	Aleatoria		Chi-cuadrado		Híbrida	
Instancia	Mínimo	Media $\pm$ DE	Mínimo	Media $\pm$ DE	Mínimo	Media $\pm$ DE
DLBCL	3.10	4.52 $\pm$ 0.52	1.43	3.70 $\pm$ 0.89	6.00	8.17 $\pm$ 0.77
Lung Cancer	1.96	4.01 $\pm$ 1.17	3.62	5.01 $\pm$ 1.02	4.29	6.02 $\pm$ 1.20
warpAR10P	39.65	41.83 $\pm$ 1.60	27.91	31.07 $\pm$ 1.40	21.05	24.64 $\pm$ 1.31
warpPIE10P	1.50	4.23 $\pm$ 0.97	1.82	3.19 $\pm$ 1.40	2.68	4.00 $\pm$ 0.73
penbased	0.60	13.34 $\pm$ 18.91	0.63	14.59 $\pm$ 20.46	0.58	14.22 $\pm$ 20.12
texture	0.73	9.19 $\pm$ 15.07	0.73	9.91 $\pm$ 15.98	0.75	10.06 $\pm$ 16.32

Tabla 11: Resultados del error porcentual en el proceso de entrenamiento - 50 % de características iniciales.

Con respecto al promedio, se puede observar que la inicialización aleatoria es la que ahora tiene las menores medias de error en 3 de 6 instancias. Esto también puede deberse a la cantidad de características iniciales, ya que las otras inicializaciones están agregando una mayor cantidad de características con chi-cuadrado que puede que no sean las más adecuadas para minimizar el error. Sin embargo, la implementación aleatoria sigue demostrando pésimos resultados en *warpAR10P* con un 41.83 % de error en promedio. Por último, con respecto a los mínimos, se puede observar que estos se encuentran más cerca del promedio a comparación de los resultados en la tabla 6, lo cual también se refleja en las menores desviaciones estándar observadas en general. Esto sugiere que, al utilizar una mayor cantidad de características, se reduce la capacidad encontrar soluciones con alta diversidad en

el frente de Pareto. Por último, se observa que los últimos dos conjuntos tienen resultados similares entre las tres inicializaciones, lo que permite pensar que, al igual que los resultados con 10 % de características, todas son apropiadas para incluir en NSGA-II para conjuntos con poca cantidad de características.

Al observar la tabla 12, se puede observar que se obtuvieron peores resultados en términos de mínimos y de media de error a comparación de la tabla 7. Esto nuevamente se atribuye a la cantidad de características seleccionadas, ya que existe una mayor probabilidad de seleccionar características irrelevantes y redundantes durante el proceso de NSGA-II. Además, también se puede ver que la inicialización aleatoria presenta la mayor desviación estándar en 3 de los 6 conjuntos, lo cual se atribuye a su gran capacidad de exploración.

Instancia	Aleatoria		Chi-cuadrado		Híbrida	
	Mínimo	Media $\pm$ DE	Mínimo	Media $\pm$ DE	Mínimo	Media $\pm$ DE
DLBCL	16.67	21.76 $\pm$ 4.92	17.00	23.56 $\pm$ 2.87	12.96	16.14 $\pm$ 2.07
Lung Cancer	2.50	13.70 $\pm$ 6.57	6.07	12.59 $\pm$ 4.58	5.59	18.15 $\pm$ 6.85
warpAR10P	60.61	72.44 $\pm$ 7.39	45.77	68.24 $\pm$ 12.06	49.21	63.81 $\pm$ 10.68
warpPIE10P	53.44	67.29 $\pm$ 11.06	53.40	66.75 $\pm$ 7.95	49.50	68.21 $\pm$ 7.34
penbased	1.27	15.66 $\pm$ 19.64	1.48	15.16 $\pm$ 19.19	1.60	15.35 $\pm$ 19.47
texture	2.37	11.43 $\pm$ 14.38	2.52	12.04 $\pm$ 15.37	2.52	12.49 $\pm$ 16.04

Tabla 12: Resultados del error porcentual en el proceso de prueba - 50 % de características iniciales.

Con respecto a los mínimos, no se observan patrones claros ni diferencias que permitan sacar conclusiones, ya que estos valores se encuentran alejados del promedio en la mayoría de los casos, lo que demuestra que existe cierta inestabilidad. Por otro lado, respecto a los promedios, se puede observar que están muy parejos en las últimas tres instancias, mientras que en el resto la solución híbrida es la que tiene mejores promedios. Esto se puede deber a su capacidad de encontrar buenas características desde un inicio junto a la adición de nuevos factores que pueden promover mejores resultados en general.

En la tabla 13 se presenta la cantidad de características seleccionadas en las soluciones obtenidas cuando se utilizó un 50 % de características iniciales. A diferencia de los resultados en la tabla 8, se observa que ahora las medidas son considerablemente mayores, lo que es esperable dado que se comienza con una mayor cantidad de características, donde se puede añadir más información irrelevante y redundante. Esto hace que reducir significativamente la dimensionalidad sea más complejo.

Lo más notorio con respecto a los mínimos es que, al igual que en la tabla 8, los valores en *penbased* y *texture* para todas las inicializaciones son de 1. Esto da a entender que, sin importar la cantidad de características que se utilicen en la inicialización, los tres algoritmos permiten que NSGA-II alcance soluciones con la menor cantidad de características posibles. Con respecto al resto de mínimos, se puede observar que todos se encuentran bastante

Iniciación Instancia	Aleatoria		Chi-cuadrado		Híbrida	
	Mínimo	Media $\pm$ DE	Mínimo	Media $\pm$ DE	Mínimo	Media $\pm$ DE
DLBCL	1892	1912.87 $\pm$ 18.19	2003	2041.77 $\pm$ 19.59	1939	1967.08 $\pm$ 26.33
Lung Cancer	5277	5377.11 $\pm$ 43.65	5531	5557.71 $\pm$ 16.78	5362	5416.06 $\pm$ 46.47
warpAR10P	574	592.81 $\pm$ 12.77	618	648.06 $\pm$ 22.62	551	584.92 $\pm$ 16.23
warpPIE10P	534	580.53 $\pm$ 32.18	586	609.25 $\pm$ 11.16	555	589.76 $\pm$ 20.90
penbased	1	6.70 $\pm$ 3.91	1	6.27 $\pm$ 3.52	1	6.67 $\pm$ 3.93
texture	1	7.60 $\pm$ 5.08	1	7.70 $\pm$ 5.25	1	7.37 $\pm$ 4.73

Tabla 13: Resultados de la cantidad de características - 50 % de características iniciales.

cerca de las medias. Sin embargo nuevamente es chi-cuadrado quien obtuvo los mayores mínimos, reforzando aún más la idea de que no es buena implementación para explorar zonas del espacio de búsqueda. A diferencia de chi-cuadrado, la inicialización aleatoria fue la que obtuvo menores mínimos en 3 de las primeras 4 instancias (menos en *warpAR10P*), lo cual resulta consistente con las medias obtenidas, confirmando que esta estrategia es más efectiva al momento de reducir la dimensionalidad en estas instancias al comenzar con 50 % de características.

Siguiendo con la media de las características, se puede observar que es la inicialización aleatoria la que obtiene la menor cantidad promedio de características seleccionadas en más instancias, siendo estas *DLBCL*, *Lung_Cancer* y *warpPIE10P*. Con esto podemos apreciar que es esta implementación la que logra reducir más la dimensionalidad al utilizar un 50 % de características iniciales. La inicialización híbrida demuestra realizar una buena reducción de igual manera, al tener menor promedio en *warpAR10P* y *texture*. Ahora bien, con respecto a la implementación de chi-cuadrado, se puede observar que, aunque haya obtenido el menor promedio en *penbased*, tiene las mayores medias en 5 de los 6 conjuntos de datos, lo que deja aún más en claro el problema que tiene esta inicialización para permitir que NSGA-II explore y reduzca la dimensionalidad de las soluciones. También se observa que las últimas dos instancias se obtuvieron resultados bastante parejos.

Ahora bien, al observar la desviación estándar, se nota que la inicialización híbrida presenta mayor dispersión en 3 de las 6 instancias. Esto sugiere que su comportamiento exploratorio fue más amplio al combinar información relevante con aleatoriedad, generando soluciones con mayor variabilidad en la cantidad de características. También se observa que chi-cuadrado en el conjunto *warpAR10P* tuvo mayor desviación estándar. Esto puede deberse, al igual que en 8, a la mayor cantidad de características que tienen las nuevas soluciones que encuentra NSGA-II con esta inicialización, lo cual está alineado con el hecho de haber obtenido la mayor media en esa instancia.

5.2.2. Frentes de Pareto

En las figuras 10 y 11 se observan los frentes de Pareto obtenidos en el proceso de entrenamiento y el proceso de prueba, donde cada figura representa una instancia.

En la figura 10(a) se observa como NSGA-II con la inicialización aleatoria obtuvo los puntos con menor número de características y menor error, dominando por completo a la implementación híbrida. Aunque chi-cuadrado haya logrado obtener un mejor error en uno de los puntos a comparación de la aleatoria, sigue siendo mejor esta última dado que la diferencia entre los errores es mínima y la diferencia entre la cantidad de características es bastante considerable.

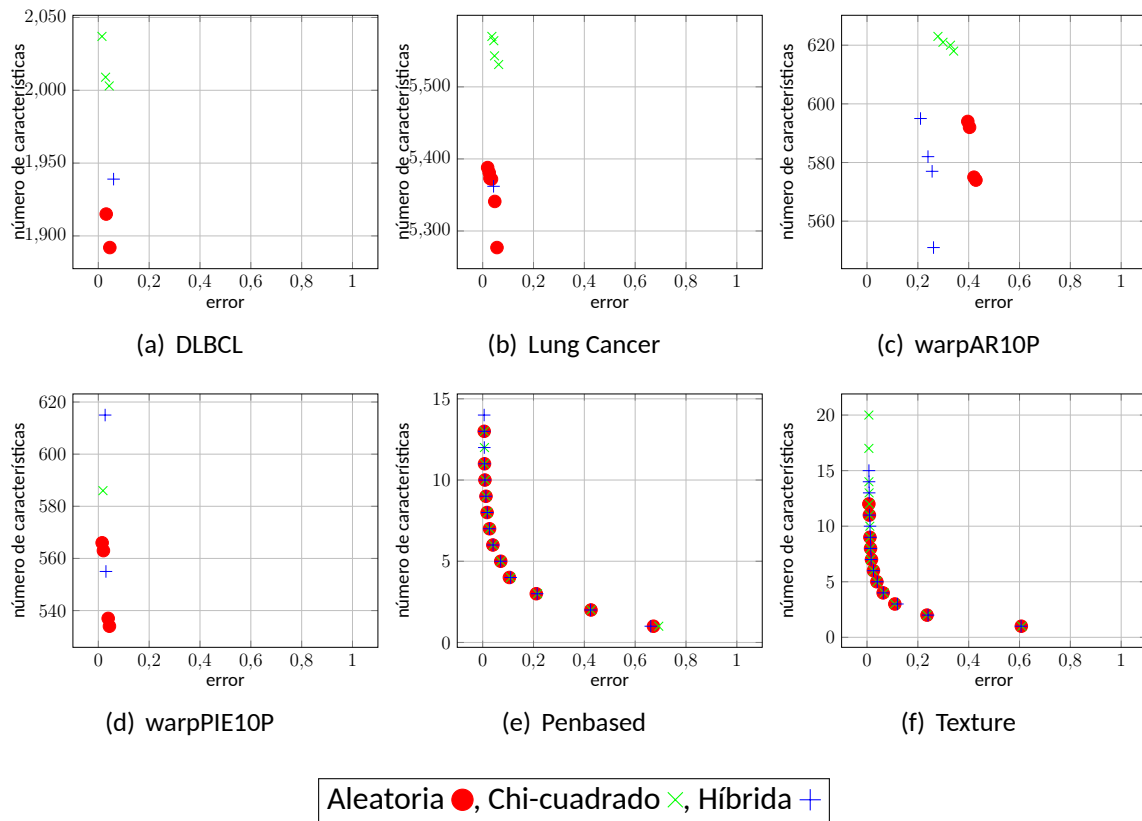


Figura 10: Frentes de Pareto del proceso de entrenamiento - 50 % de características iniciales.

Con respecto a la figura 10(b), se puede apreciar algo similar a lo anterior, solo que ahora el único punto de la implementación híbrida no está dominado por la aleatoria. Si bien esto puede dar a entender que la inicialización híbrida entrega un buen frente de Pareto para esta instancia, sigue siendo mejor la aleatoria al entregar una mayor diversidad de puntos no dominados en el frente.

En la figura 10(c) se observa claramente que es la inicialización híbrida la que genera el mejor frente de Pareto, dominando por completo a las otras dos. Esto se complementa tanto con

la tabla 11 como con la tabla 13, donde se observa claramente que obtuvo tanto el menor promedio de error como el menor promedio de características seleccionadas.

En la figura 10(d) se observa que nuevamente es la inicialización aleatoria la que obtiene un mejor frente de Pareto a comparación de los otros dos, a pesar de que uno de los puntos de la híbrida no está dominado.

En las figuras 10(e) y 10(f) podemos observar como todas las inicializaciones permiten que NSGA-II logre llegar al mismo frente de Pareto, demostrando así que todas son buenas opciones para instancias con poca cantidad de características y muchas muestras. Cabe aclarar que los frentes tienen distinta cantidad de soluciones en ambos casos. Para *Penbased* se obtuvieron 12 soluciones en el frente aleatorio, 13 en el de chi-cuadrado y 14 en el de la inicialización híbrida. Por otra parte, para *Texture* se obtuvieron 11 soluciones en el frente aleatorio, 14 en el de chi-cuadrado y 16 en el de la implementación híbrida.

En general, se puede considerar que la mejor opción para el proceso de entrenamiento con 50 % de características iniciales es la inicialización aleatoria, ya que, a diferencia de la instancia *warpAR10P*, logra los mejores frentes en todas las instancias.

Con respecto a los frentes de Pareto en la figura 11, podemos observar como ocurre lo mismo que en la figura 10 para las figuras 11(e) y 11(f). Sin embargo, se obtienen resultados distintos en el resto de los casos.

Para empezar, en la figura 11(a) se aprecia que ahora hay una menor cantidad de puntos no dominados y la implementación de chi-cuadrado está dominada por la híbrida, la cual forma un frente de Pareto en conjunto con los resultados de la inicialización aleatoria. En este caso en particular ambos frentes muestran que el rendimiento del modelo es bueno. Para considerar cuál es mejor, depende de lo que uno busque, ya que tampoco es tanta la diversidad de las soluciones en la implementación aleatoria como para afirmar que entrega un mejor frente de Pareto.

En la figura 11(b) se observa que la inicialización aleatoria dominó por completo a las inicializaciones propuestas, demostrando que para este conjunto en particular NSGA-II funciona mejor con una implementación aleatoria al inicio.

En la figura 11(c) se puede apreciar que se forma un frente de Pareto conjunto entre las tres inicializaciones, mostrando que la aleatoria es la más diversa, chi-cuadrado la que tiene menor error y la híbrida siendo la que tiene mejor proporción entre cantidad de características y error. Para este caso se vuelve a repetir lo mismo que en 11(a), donde el mejor frente depende únicamente de lo que uno busque.

En la figura 11(d) se observa que la implementación aleatoria presenta el mejor frente de Pareto, ya que entrega soluciones diversas y con un error cercano entre cada una. Si bien existe una solución del frente de la inicialización híbrida que es parte del frente de Pareto conjunto, el resto de puntos está dominado por la aleatoria.

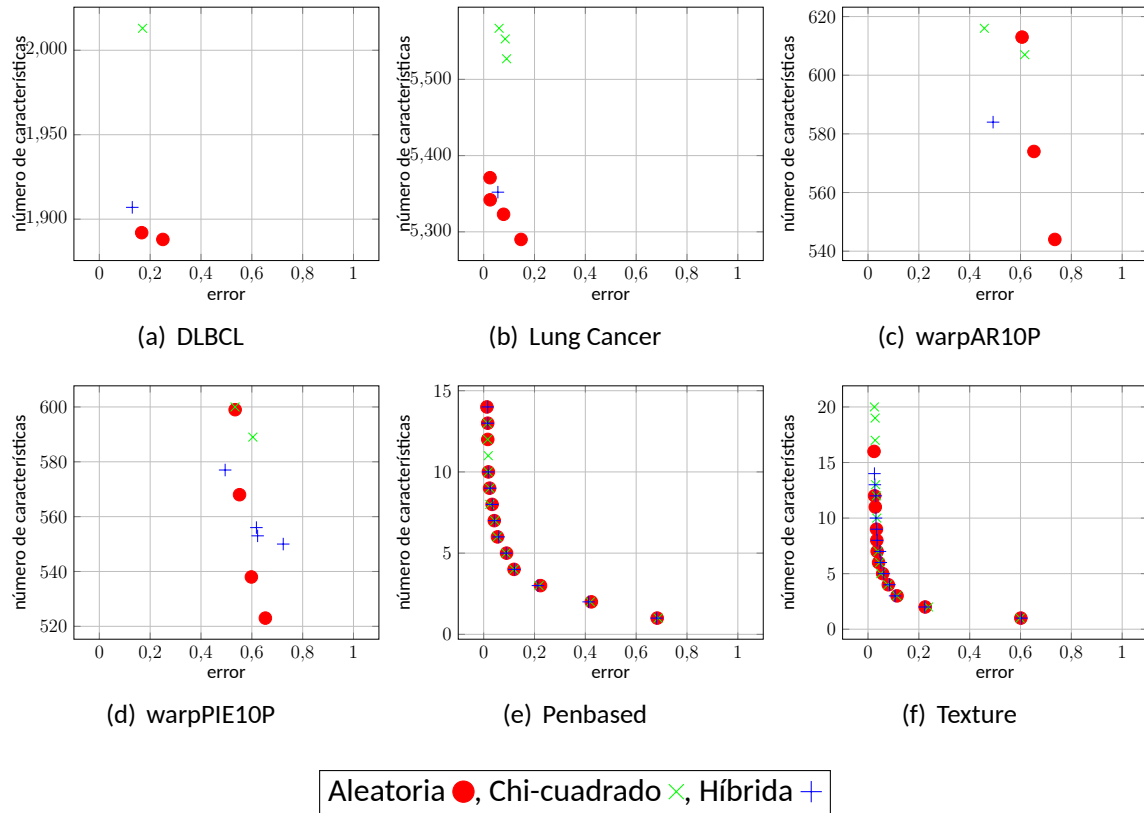


Figura 11: Frentes de Pareto del proceso de prueba - 50% de características iniciales.

Por último, cabe aclarar que no todos los frentes tienen la misma cantidad de soluciones en las subfiguras 11(e) y 11(f). Para *Penbased* se obtuvieron 13 soluciones en el frente aleatorio y el híbrido, mientras que se obtuvieron 12 en el de chi-cuadrado. Por otra parte, para *Texture* se obtuvieron 12 soluciones en el frente aleatorio, mientras que en chi-cuadrado y en la implementación híbrida se obtuvieron 13.

En general, al igual que en la figura 10, se puede considerar que la mejor opción es la inicialización aleatoria, ya que logra mantener los mejores resultados en general con conjuntos de datos que no fueron utilizados en el entrenamiento.

5.2.3. Hipervolumen

En las tablas 14 y 15 se encuentran los resultados de hipervolumen obtenidos en los conjuntos de entrenamiento y prueba. En negrita están marcados los promedios más altos para cada instancia.

Con respecto a la tabla 14, se puede observar que la inicialización aleatoria vuelve a destacar como la mejor estrategia en términos generales de hipervolumen, obteniendo el mejor pro-

medio en 4 de los 6 conjuntos de datos. Esta tendencia es coherente con lo observado previamente en la tabla 9, lo que sugiere que esta implementación tiene una mayor capacidad para generar frentes de Pareto con buen equilibrio entre error y número de características seleccionadas durante el proceso de entrenamiento, incluso cuando se inicia de una cantidad mayor de características. Por otro lado, las inicializaciones híbrida y de chi-cuadrado lograron solamente un mejor promedio en una instancia cada uno. Si bien la diferencia de promedios en *warpAR10P* es bastante mayor, en el caso de *Penbased* es muy poca, por lo que se puede observar claramente que chi-cuadrado obtuvo el peor rendimiento en términos de hipervolumen.

Inicialización	Aleatoria		Chi-cuadrado		Híbrida	
Instancia	Máximo	Promedio $\pm$ DE	Máximo	Promedio $\pm$ DE	Máximo	Promedio $\pm$ DE
DLBCL	0.6297	0.6210 $\pm$ 0.0046	0.6185	0.6048 $\pm$ 0.0081	0.6068	0.5895 $\pm$ 0.0089
Lung Cancer	0.5612	0.5524 $\pm$ 0.0060	0.5377	0.5290 $\pm$ 0.0051	0.5498	0.5365 $\pm$ 0.0080
warpAR10P	0.4546	0.4334 $\pm$ 0.0094	0.5344	0.5075 $\pm$ 0.0145	0.5938	0.5707 $\pm$ 0.0101
warpPIE10P	0.7559	0.7321 $\pm$ 0.0148	0.7441	0.7255 $\pm$ 0.0141	0.7476	0.7262 $\pm$ 0.0119
Penbased	0.8346	0.8333 $\pm$ 0.0007	0.8337	0.8337 $\pm$ 0.0001	0.8348	0.8329 $\pm$ 0.0011
Texture	0.9407	0.9387 $\pm$ 0.0013	0.9405	0.9385 $\pm$ 0.0013	0.9399	0.9383 $\pm$ 0.0008

Tabla 14: Hipervolumen en el proceso de entrenamiento - 50 % de características iniciales.

Al fijarse en los máximos obtenidos, se puede ver que, a diferencia del hipervolumen obtenido en la tabla 9, los máximos se mantuvieron más cercanos al promedio, lo cual es consistente con lo que se pudo ver en los frentes de Pareto en la figura 10. Esto mismo se refleja con los bajos valores de desviación estándar, mostrando que existe una mayor posibilidad de llegar a estos resultados. El caso más llamativo de esto ocurre en la instancia *Penbased* utilizando la inicialización de chi-cuadrado, ya que el máximo obtenido y el promedio son exactamente el mismo valor. Esto quiere decir que al utilizar esta implementación para este conjunto de datos, siempre se obtendrán resultados prácticamente iguales entre las ejecuciones, sobre todo considerando que la desviación estándar es de apenas 0.0001. La justificación de este fenómeno se puede dar, nuevamente, gracias al hecho de que es una implementación muy elitista.

Con respecto a la tabla 15, se puede observar que la inicialización híbrida obtiene el mejor promedio en 3 de las 6 instancias, siendo la que mejor generaliza al equilibrar exploración y explotación desde un inicio. A diferencia de lo que se observó en la tabla 10, donde chi-cuadrado fue el que mejor generalizó en casi todas las instancias, aquí solamente generalizó mejor en 2 de las 6 instancias, demostrando ser levemente inferior que la implementación híbrida al iniciar NSGA-II con 50 % de características. Por último, pese a que obtuvo mejores resultados en el proceso de entrenamiento, la inicialización aleatoria demostró tener problemas a la hora de generalizar, ya que solamente obtuvo un mejor promedio en *Lung_Cancer*.

Al observar los valores máximos y las desviaciones estándar, se nota una mayor dispersión en las primeras 4 instancias, donde los máximos tienden a alejarse considerablemente del

Inicialización	Aleatoria		Chi-cuadrado		Híbrida	
Instancia	Máximo	Promedio $\pm$ DE	Máximo	Promedio $\pm$ DE	Máximo	Promedio $\pm$ DE
DLBCL	0.5451	0.4957 $\pm$ 0.0333	0.5245	0.4827 $\pm$ 0.0225	0.5669	0.5392 $\pm$ 0.0165
Lung Cancer	0.5613	0.5032 $\pm$ 0.0476	0.5243	0.4871 $\pm$ 0.0321	0.5430	0.4812 $\pm$ 0.0414
warpAR10P	0.2933	0.1992 $\pm$ 0.0649	0.4031	0.2470 $\pm$ 0.0951	0.3843	0.2624 $\pm$ 0.0860
warpPIE10P	0.3504	0.2499 $\pm$ 0.0828	0.3505	0.2607 $\pm$ 0.0596	0.3846	0.2486 $\pm$ 0.0631
Penbased	0.8211	0.8183 $\pm$ 0.0022	0.8223	0.8190 $\pm$ 0.0022	0.8211	0.8184 $\pm$ 0.0024
Texture	0.9230	0.9163 $\pm$ 0.0050	0.9198	0.9160 $\pm$ 0.0036	0.9227	0.9165 $\pm$ 0.0046

Tabla 15: Hipervolumen en el proceso de prueba - 50 % de características iniciales.

promedio (sobre todo en *warpAR10P* y *warpPIE10P*). Esto indica que, aunque hay ejecuciones que logran hipervolumenes altos, la estabilidad y repetibilidad de esos resultados es baja, reflejando menor robustez. Por último, en las últimas dos instancias, las tres estrategias presentan desviaciones estándar muy bajas y resultados casi idénticos, confirmando así que en instancias de baja dimensionalidad todas las inicializaciones permiten alcanzar soluciones estables.

5.3. Comparaciones con el estado del arte

En las tablas 16 y 17 se presentan los promedios de hipervolumen y sus correspondientes desviaciones estándar obtenidos para cada proceso de inicialización en los conjuntos de entrenamiento y prueba. En estas comparaciones se incluyen los promedios de hipervolumen y sus correspondientes desviaciones estándar publicados para CNSGA-II en [Miyandoab *et al.*, 2023]. Se presentan las comparaciones con los escenarios de 50 % de características iniciales primero para hacer una comparación pareja con el estado del arte.

Al observar la tabla 16, se puede observar que para *warpAR10P* la inicialización híbrida fue la que obtuvo el mejor promedio de hipervolumen, siendo la única que superó al estado del arte. Mientras tanto en *warpPIE10P*, si bien la inicialización aleatoria fue la que obtuvo un mejor hipervolumen, los hipervolumenes de las dos inicializaciones propuestas lograron que NSGA-II obtuviera un mayor hipervolumen promedio en sus soluciones a diferencia de CNSGA-II.

Ahora bien, al fijarse en la tabla 17, se puede ver que para ambos casos la implementación del estado del arte logró mejores resultados.

Con esto se puede apreciar que, aunque las inicializaciones propuestas en este trabajo lograron un mejor promedio al utilizar el conjunto de datos de entrenamiento, demostraron tener un peor desempeño a la hora de evaluar los frentes obtenidos, lo que indica que existe una menor capacidad de generalización en comparación con CNSGA-II.

Inicialización	Aleatoria	Chi-cuadrado	Híbrida	CNSGA-II
Instancia	Promedio $\pm$ DE	Promedio $\pm$ DE	Promedio $\pm$ DE	Promedio $\pm$ DE
warpAR10P	0.4334 $\pm$ 0.0094	0.5075 $\pm$ 0.0145	0.5707 $\pm$ 0.0101	0.54 $\pm$ 0.03
warpPIE10P	0.7321 $\pm$ 0.0148	0.7255 $\pm$ 0.0141	0.7262 $\pm$ 0.0119	0.65 $\pm$ 0.01

Tabla 16: Comparación de hipervolumen en el proceso de entrenamiento con CNSGA-II - 50 % de características iniciales.

Inicialización	Aleatoria	Chi-cuadrado	Híbrida	CNSGA-II
Instancia	Promedio $\pm$ DE	Promedio $\pm$ DE	Promedio $\pm$ DE	Promedio $\pm$ DE
warpAR10P	0.1992 $\pm$ 0.0649	0.2470 $\pm$ 0.0951	0.2624 $\pm$ 0.0860	0.33 $\pm$ 0.04
warpPIE10P	0.2499 $\pm$ 0.0828	0.2607 $\pm$ 0.0596	0.2486 $\pm$ 0.0631	0.43 $\pm$ 0.04

Tabla 17: Comparación de hipervolumen en el proceso de prueba con CNSGA-II - 50 % de características iniciales.

Ahora bien, en las tablas 18 y 19 se observan las mismas comparaciones, solo que considerando a NSGA-II inicializado con un 10 % de características. En estas se puede apreciar que claramente son mejores los resultados obtenidos por NSGA-II al utilizar las inicializaciones propuestas, siendo chi-cuadrado el mejor en 3 de las 4 comparaciones. Esto es bastante lógico, ya que se utiliza una menor cantidad de características, por lo que es esperable que se obtenga un mejor resultado de hipervolumen. Incluso sabiendo eso, se destaca el hecho de que CNSGA-II está muy cerca del resultado obtenido en *warpPIE10P* en la tabla 19, ya que este se inicializa con un 40 % más de características.

Inicialización	Aleatoria	Chi-cuadrado	Híbrida	CNSGA-II
Instancia	Promedio $\pm$ DE	Promedio $\pm$ DE	Promedio $\pm$ DE	Promedio $\pm$ DE
warpAR10P	0.6638 $\pm$ 0.0480	0.8288 $\pm$ 0.0287	0.7817 $\pm$ 0.0329	0.54 $\pm$ 0.03
warpPIE10P	0.9375 $\pm$ 0.0091	0.9444 $\pm$ 0.0074	0.9494 $\pm$ 0.0103	0.65 $\pm$ 0.01

Tabla 18: Comparación de hipervolumen en el proceso de entrenamiento con CNSGA-II - 10 % de características iniciales.

5.4. Tiempos obtenidos

En las tablas 20 y 21 se presentan los tiempos medidos por ejecución en segundos por cada inicialización en cada conjunto de datos.

En estas se puede apreciar que los tiempos de ejecución se mantuvieron bastante parejos entre sí. Esto indica que tanto la implementación de chi-cuadrado como la híbrida no presentan grandes consumos de cómputo a comparación de la aleatoria, permitiendo así que

Inicialización	Aleatoria	Chi-cuadrado	Híbrida	CNSGA-II
Instancia	Promedio $\pm$ DE	Promedio $\pm$ DE	Promedio $\pm$ DE	Promedio $\pm$ DE
warpAR10P	0.2853 $\pm$ 0.0565	0.4826 $\pm$ 0.0937	0.3954 $\pm$ 0.0984	0.33 $\pm$ 0.04
warpPIE10P	0.3468 $\pm$ 0.1325	0.4918 $\pm$ 0.0863	0.4660 $\pm$ 0.0857	0.43 $\pm$ 0.04

Tabla 19: Comparación de hipervolumen en el proceso de prueba con CNSGA-II - 10 % de características iniciales.

la comparación entre estrategias sea únicamente basada en rendimiento. Además, cabe resaltar que los tiempos de ejecución con un 10 % de características iniciales fueron menores en comparaciones de los otros. Esto se debe a que el algoritmo va a tardar más tiempo en evaluar cada subconjunto mientras mayor cantidad de información se le entregue, por lo que es claro que es más conveniente utilizar una menor cantidad de características iniciales para hacer que el proceso de NSGA-II sea más eficiente.

Instancia	Aleatoria	Chi cuadrado	Híbrida
DLBCL	3,334	3,347	3,302
Lung Cancer	7,076	7,116	7,124
warpAR10P	3,412	3,413	3,391
warpPIE10P	3,607	3,643	3,602
penbased	8,589	8,555	8,679
texture	6,831	6,944	6,927

Tabla 20: Tiempo promedio de ejecución de los experimentos (en segundos) - 10 % de características iniciales.

Instancia	Aleatoria	Chi cuadrado	Híbrida
DLBCL	5,082	5,353	5,317
Lung Cancer	13,480	11,740	11,766
warpAR10P	4,150	4,222	4,088
warpPIE10P	4,675	4,664	4,597
penbased	8,815	8,793	8,828
texture	6,780	6,601	6,757

Tabla 21: Tiempo promedio de ejecución de los experimentos (en segundos) - 50 % de características iniciales.

CONCLUSIONES

Este trabajo abordó el problema de selección de características desde una perspectiva de optimización multi-objetivo, empleando el algoritmo evolutivo NSGA-II como técnica base. A través del estudio teórico y práctico, se propuso la incorporación de la prueba estadística de chi-cuadrado en el proceso de inicialización del algoritmo, evaluando tanto una estrategia puramente basada en esta métrica como una estrategia híbrida que combina dicha métrica con un componente aleatorio. Además, se compararon los resultados obtenidos al utilizar distintos porcentajes iniciales de características (10 % y 50 %) para explorar el efecto que tiene esta configuración sobre la calidad de las soluciones finales.

En primer lugar, se respondieron las preguntas planteadas en la introducción del trabajo:

- **¿Puede la prueba estadística de chi-cuadrado mejorar la calidad de la población inicial en NSGA-II aplicada al problema de selección de características?**

Sí. Los resultados demostraron que la inicialización basada en chi-cuadrado mejora significativamente el error promedio y mínimo de las soluciones, especialmente cuando se parte con un 10 % de características, lo que sugiere una mayor eficiencia en la generación de soluciones relevantes desde la primera generación.

- **¿La implementación híbrida mejora el equilibrio entre exploración y explotación durante la búsqueda evolutiva?**

Efectivamente. La estrategia híbrida mostró un comportamiento más equilibrado, combinando la explotación de características relevantes con la exploración de nuevas combinaciones. Esto se reflejó en su capacidad para encontrar soluciones con buen rendimiento y baja dimensionalidad, especialmente en instancias complejas como *warpAR10P*.

- **¿Existe una diferencia a la hora de inicializar NSGA-II con un 10 % y un 50 % de características?**

Sí. Se observó que inicializar con un 10 % de características permitió reducir la dimensionalidad, disminuir los errores y mejorar los resultados de hipervolumen. A diferencia de este, al utilizar un 50 % de características aumentó la variabilidad de los resultados, dificultando la convergencia hacia soluciones óptimas y aumentando la presencia de características irrelevantes y redundantes.

- **¿Se obtienen mejores resultados de hipervolumen a comparación del estado del arte?**

En algunos casos, sí. Hablando específicamente de las comparaciones entre CNSGA-II y las propuestas con 10 % de características, se observaron mejores valores de hipervolumen tanto en el proceso de entrenamiento como en el proceso de prueba. Sin embargo, al comparar CNSGA-II y las propuestas con 50 % de características iniciales, se

observó que se obtuvieron peores resultados en el proceso de prueba, lo cual evidencia lo necesario que es trabajar en una propuesta que generalice mejor los conjuntos de datos externos al proceso de entrenamiento.

■ **¿Qué tanto afectan las nuevas inicializaciones a los tiempos de ejecución del algoritmo?**

Muy poco. Los tiempos de ejecución entre las tres estrategias de inicialización fueron similares, lo cual demuestra que las mejoras introducidas en el rendimiento no implican un aumento significativo en el costo computacional.

■ **¿Las implementaciones propuestas son capaces de aplicarse en distintas áreas y volúmenes de datos?**

Sí. Se utilizaron conjuntos de datos provenientes de distintas áreas como bioinformática, reconocimiento de imágenes, visión artificial y escritura a mano. En todos los casos, las inicializaciones propuestas mostraron un buen desempeño general, lo cual demuestra la versatilidad del enfoque.

Conclusiones generales

A partir del análisis de los resultados, se puede concluir que:

- La inicialización de NSGA-II basada en chi-cuadrado presenta una mejora significativa a comparación del resto en términos de error, siendo la propuesta que entregó los menores promedios. Esto es gracias a los buenos resultados que genera desde un comienzo.
- La estrategia híbrida propuesta representa una excelente alternativa al incorporar un balance entre características relevantes y aleatorias, promoviendo una exploración más efectiva del espacio de búsqueda sin perder precisión.
- El uso de un menor porcentaje inicial de características (10 %) fue, en general, más beneficioso para obtener soluciones con menor error y mayor capacidad de generalización, además de ser más eficiente desde el punto de vista computacional.
- La comparación con el estado del arte mostró que las estrategias propuestas pueden ser competitivas y superarlo en algunos contextos, aunque se deben seguir explorando mecanismos que aumenten la estabilidad y la capacidad de generalización.

Trabajo futuro

Este trabajo abre diversas líneas de investigación que pueden ser abordadas en estudios futuros, entre las cuales destacan las siguientes:

- **Diseño de operadores de mutación basados en chi-cuadrado:** Aprovechar los valores estadísticos de chi-cuadrado no solo para la inicialización, sino también como criterio de mutación, promoviendo la selección de características relevantes y agregando características complementarias durante las generaciones.
- **Incorporación de nuevos operadores de cruzamiento:** Investigar combinaciones de operadores de cruzamiento que potencien la explotación del espacio de soluciones, fomentando la convergencia hacia regiones más prometedoras sin comprometer la diversidad.
- **Incorporación de nuevos algoritmos de filtro:** Analizar el impacto de nuevas estrategias de inicialización basadas en filtros distintos al chi-cuadrado, como mutual information, con el fin de comparar su influencia sobre la calidad y generalización de las soluciones.
- **Extensión a otros dominios:** Aplicar las estrategias propuestas a conjuntos de datos de mayor escala, estructuras de alta dimensionalidad o instancias con clases altamente desbalanceadas, evaluando su robustez y adaptabilidad.
- **Comparación con más algoritmos del estado del arte:** Ampliar el análisis comparativo incorporando algoritmos relevantes en la literatura, como MOEA/D o SPEA2, para validar el desempeño de las estrategias de inicialización propuestas en un entorno más competitivo.

En resumen, esta investigación demuestra que la incorporación de pruebas estadísticas en la etapa inicial de NSGA-II puede mejorar notablemente la calidad de las soluciones obtenidas, y abre el camino hacia futuras mejoras en exploración, convergencia y eficiencia para el problema de selección de características.

REFERENCIAS BIBLIOGRÁFICAS

- [Adoptante *et al.*, 2023] Adoptante, A. J., Baes, A., Catilo, J. C., Lucero, P. K., De Ocampo, A. L., S. Alon D.Eng, A., y Dellosa, R. (2023). Spoken-digit classification using artificial neural network. ASEAN Engineering Journal, 13:93–99.
- [Aha *et al.*, 1991] Aha, D. W., Kibler, D., y Albert, M. K. (1991). Instance-based learning algorithms. Machine Learning, 6(1):37–66.
- [Alcalá-Fdez *et al.*, 2009] Alcalá-Fdez, J., Sánchez, L., García, S., del Jesus, M. J., Ventura, S., Garrell, J. M., Otero, J., Romero, C., Bacardit, J., Rivas, V. M., Fernández, J. C., y Herrera, F. (2009). KEEL: A software tool to assess evolutionary algorithms for data mining problems. Soft Computing, 13(3):307–318.
- [Amoozegar y Minaei-Bidgoli, 2018] Amoozegar, M. y Minaei-Bidgoli, B. (2018). Optimizing multi-objective pso based feature selection method using a feature elitism mechanism. Expert Systems with Applications, 113:499–514.
- [Beume *et al.*, 2009] Beume, N., Fonseca, C. M., Lopez-Ibanez, M., Paquete, L., y Vahrenhold, J. (2009). On the complexity of computing the hypervolume indicator. IEEE Transactions on Evolutionary Computation, 13(5):1075–1082.
- [Cagnina *et al.*, 2005] Cagnina, L., Esquivel, S., y Coello, C. A. C. (2005). A particle swarm optimizer for multi-objective optimization. Journal of Computer Science and Technology, 5(04):204–210.
- [Cai *et al.*, 2018] Cai, J., Luo, J., Wang, S., y Yang, S. (2018). Feature selection in machine learning: A new perspective. Neurocomputing, 300:70–79.
- [Clark y Niblett, 1989] Clark, P. y Niblett, T. (1989). The cn2 induction algorithm. Machine Learning, 3(4):261–283.
- [Coello *et al.*, 2007] Coello, C., Lamont, G., y Veldhuizen, D. V. (2007). Evolutionary Algorithms for Solving Multi-Objective Problems. Springer, New York, NY.
- [Cortes y Vapnik, 1995] Cortes, C. y Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3):273–297.
- [Cunningham y Delany, 2021] Cunningham, P. y Delany, S. J. (2021). k-nearest neighbour classifiers – a tutorial. ACM Comput. Surv., 54(6).
- [Dash y Liu, 2003] Dash, M. y Liu, H. (2003). Consistency-based search in feature selection. Artificial Intelligence, 151(1):155–176.
- [Dash *et al.*, 2000] Dash, M., Liu, H., y Motoda, H. (2000). Consistency based feature selection. En Knowledge Discovery and Data Mining. Current Issues and New Applications, pp. 98–109, Berlin, Heidelberg. Springer Berlin Heidelberg.

- [Deb *et al.*, 2002] Deb, K., Pratap, A., Agarwal, S., y Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Transactions on Evolutionary Computation, 6(2):182–197.
- [Fonseca *et al.*, 2006] Fonseca, C., Paquete, L., y Lopez-Ibanez, M. (2006). An improved dimension-sweep algorithm for the hypervolume indicator. En 2006 IEEE International Conference on Evolutionary Computation, pp. 1157–1163.
- [Gong *et al.*, 2024] Gong, H., Li, Y., Zhang, J., Zhang, B., y Wang, X. (2024). A new filter feature selection algorithm for classification task by ensembling pearson correlation coefficient and mutual information. Engineering Applications of Artificial Intelligence, 131:107865.
- [Hall, 2000] Hall, M. A. (2000). Correlation-based feature selection for discrete and numeric class machine learning. En Proceedings of the Seventeenth International Conference on Machine Learning, ICML '00, p. 359–366, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- [Hastie *et al.*, 2009] Hastie, T., Tibshirani, R., y Friedman, J. (2009). The Elements of Statistical Learning. Springer Series in Statistics. Springer New York, NY, 2 edición.
- [HDS, 2021] HDS (2021). K-nearest neighbors algorithm: Classification and regression star.
- [Inza *et al.*, 2004] Inza, I., Larrañaga, P., Blanco, R., y Cerrolaza, A. J. (2004). Filter versus wrapper gene selection approaches in dna microarray domains. Artificial Intelligence in Medicine, 31(2):91–103. Data Mining in Genomics and Proteomics.
- [Jin *et al.*, 2006] Jin, X., Xu, A., Bie, R., y Guo, P. (2006). Machine learning techniques and chi-square feature selection for cancer classification using sage gene expression profiles. En Data Mining for Biomedical Applications, pp. 106–115. Springer.
- [Johnson *et al.*, 2017] Johnson, J., Douze, M., y Jégou, H. (2017). Faiss: A library for efficient similarity search. <https://engineering.fb.com/2017/03/29/data-infrastructure/faiss-a-library-for-efficient-similarity-search/>.
- [Kelly *et al.*, 2023] Kelly, M., Longjohn, R., y Nottingham, K. (2023). The uci machine learning repository.
- [Kennedy y Eberhart, 1995] Kennedy, J. y Eberhart, R. (1995). Particle swarm optimization. En Proceedings of ICNN'95 - International Conference on Neural Networks, volumen 4, pp. 1942–1948 vol.4.
- [Kirch, 2008] Kirch, W. (2008). Encyclopedia of Public Health. Springer Dordrecht, 1 edición.
- [Kittler, 1978] Kittler, J. (1978). Feature set search algorithms. En Pattern Recognition and Signal Processing, pp. 41–60. Sijthoff & Noordhoff, Alphen aan den Rijn, Netherlands.
- [Kononenko, 1994] Kononenko, I. (1994). Estimating attributes: Analysis and extensions of relief. En Machine Learning: ECML-94, pp. 171–182, Berlin, Heidelberg. Springer Berlin Heidelberg.

- [Langley *et al.*, 1992] Langley, P., Iba, W., y Thompson, K. (1992). An analysis of bayesian classifiers. En Proceedings of the Tenth National Conference on Artificial Intelligence, pp. 223–228, San Jose, CA, USA. AAAI Press.
- [Ma y Huang, 2008] Ma, S. y Huang, J. (2008). Penalized feature selection and classification in bioinformatics. Briefings in Bioinformatics, 9(5):392–403.
- [Miyandoab *et al.*, 2023] Miyandoab, S. Z., Rahnamayan, S., y Bidgoli, A. A. (2023). Compact nsga-ii for multi-objective feature selection. En 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE.
- [Mlakar *et al.*, 2017] Mlakar, U., Fister, I., Brest, J., y Potočnik, B. (2017). Multi-objective differential evolution for feature selection in facial expression recognition systems. Expert Systems with Applications, 89:129–137.
- [Perera *et al.*, 2023] Perera, Y. S., Ratnaweera, D., Dasanayaka, C. H., y Abeykoon, C. (2023). The role of artificial intelligence-driven soft sensors in advanced sustainable process industries: A critical review. Engineering Applications of Artificial Intelligence, 121:105988.
- [Prusty *et al.*, 2022] Prusty, S., Patnaik, S., y Dash, S. (2022). Skcv: Stratified k-fold cross-validation on ml classifiers for predicting cervical cancer. Frontiers in Nanotechnology, 4:972421.
- [Quinlan, 1986a] Quinlan, J. R. (1986a). Induction of decision trees. Machine Learning, 1(1):81–106.
- [Quinlan, 1986b] Quinlan, J. R. (1986b). Induction of decision trees. Machine Learning, 1(1):81–106.
- [Salzberg, 1994] Salzberg, S. L. (1994). C4.5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. Machine Learning, 16(3):235–240.
- [Scikit-learn, 2024] Scikit-learn (2024). sklearn.preprocessing.kbinsdiscretizer — scikit-learn documentation.
- [Statnikov *et al.*, 2004] Statnikov, A., Aliferis, C. F., Tsamardinos, I., Hardin, D., y Levy, S. (2004). A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. Bioinformatics, 21(5):631–643.
- [Storn y Price, 1997] Storn, R. y Price, K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization, 11(4):341–359.
- [Tang *et al.*, 2014] Tang, J., Alelyani, S., y Liu, H. (2014). Feature selection for classification: A review, pp. 37–64. CRC Press. Publisher Copyright: © 2015 by Taylor Francis Group, LLC.

- [Tran *et al.*, 2018] Tran, B., Xue, B., y Zhang, M. (2018). A new representation in pso for discretization-based feature selection. IEEE Transactions on Cybernetics, 48(6):1733–1746.
- [Vergara y Estévez, 2014] Vergara, J. R. y Estévez, P. A. (2014). A review of feature selection methods based on mutual information. Neural Computing and Applications, 24(1):175–186.
- [Xu *et al.*, 2020] Xu, Y., Yan, C., Liu, H., Wang, J., Yang, Z., y Jiang, Y. (2020). Smart energy systems: A critical review on design and operation optimization. Sustainable Cities and Society, 62:102369.
- [Xue *et al.*, 2013] Xue, B., Zhang, M., y Browne, W. N. (2013). Particle swarm optimization for feature selection in classification: A multi-objective approach. IEEE Transactions on Cybernetics, 43(6):1656–1671.
- [Xue *et al.*, 2023] Xue, Y., Zhu, H., y Neri, F. (2023). A feature selection approach based on nsga-ii with relieff. Applied Soft Computing, 134:109987.
- [Yang y Pedersen, 1997] Yang, Y. y Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. En Proc. 14th Int. Conf. Machine Learning (ICML), pp. 412–420.
- [Yu y Liu, 2003] Yu, L. y Liu, H. (2003). Feature selection for high-dimensional data: A fast correlation-based filter solution. En Proc. 20th Int. Conf. Machine Learning (ICML), pp. 856–863.
- [Zhang *et al.*, 2015] Zhang, Y., Gong, D.-W., y Cheng, J. (2015). Multi-objective particle swarm optimization approach for cost-based feature selection in classification. IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM, 14.
- [Zhao *et al.*, 2010a] Zhao, Z., Morstatter, F., Sharma, S., Alelyani, S., Anand, A., y Liu, H. (2010a). Advancing feature selection research. ASU Feature Selection Repository Arizona State University, pp. 1–28.
- [Zhao *et al.*, 2010b] Zhao, Z., Morstatter, F., Sharma, S., Alelyani, S., Anand, A., y Liu, H. (2010b). Advancing feature selection research. ASU Feature Selection Repository Arizona State University, pp. 1–28.
- [Zhou *et al.*, 2011] Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N., y Zhang, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. Swarm and Evolutionary Computation, 1(1):32–49.