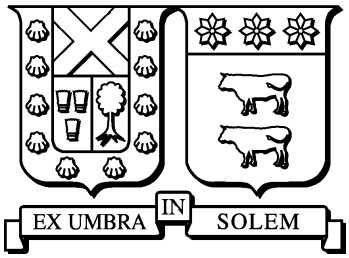



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE INFORMÁTICA

# SELECCIÓN DE CARACTERÍSTICAS: UNA PROPUESTA DE NSGA-II CON NUEVOS OPERADORES

Nicolás Paz Tralma

MAGÍSTER EN CIENCIAS DE LA INGENIERÍA INFORMÁTICA

ENERO 2026

# 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 NUEVOS OPERADORES

**Nombre del candidato(a):** Nicolás Patricio Paz Tralma

**Carrera / Grado:** Magister en Ciencias de la Ingeniería Informática

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

## 2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Elizabeth Del Carmen Montero Ureta, 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

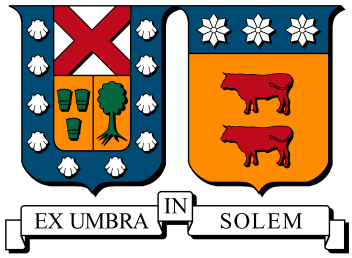
**Profesor(a) guía o director(a) de memoria o tesis:**

**Fecha: 04/01/2026** ; **Firma:** E Montero

**Estudiante o Candidato(a):**

**Fecha:** 04/01/2026 ; **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.*




UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA

# SELECCIÓN DE CARACTERÍSTICAS: UNA PROPUESTA DE NSGA-II CON NUEVOS OPERADORES

Tesis de Grado presentada por
NICOLÁS PAZ TRALMA

como requisito parcial para optar al grado de
MAGÍSTER EN CIENCIAS DE LA INGENIERÍA INFORMÁTICA

PROFESOR GUÍA
ELIZABETH MONTERO

ENERO 2026

TÍTULO DE LA TESIS:
SELECCIÓN DE CARACTERÍSTICAS: UNA PROPUESTA DE NSGA-II CON NUEVOS OPERADORES

AUTOR:
NICOLÁS PAZ TRALMA

Trabajo de Tesis, presentado en cumplimiento parcial de los requisitos para el Grado de Magíster en Ciencias de la Ingeniería Informática de la Universidad Técnica Federico Santa María.

Elizabeth Montero _____
Profesor Guía

María Cristina Riff _____
Profesor Correferente

Ignacio Araya _____
Profesor Externo

Enero 2026.
Santiago, Chile.

# Agradecimientos

Quiero agradecer a todas las personas que han formado parte de este proceso. A mi familia, a mis amigos y a los profesores que se me han ayudado a completar este trabajo. 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. Quiero agradecer especialmente a mi profesora guía, ya que me brindó la oportunidad de presentar mi trabajo en una conferencia internacional que me ayudó a enfocarme en mis objetivos personales para el futuro.

# Resumen

Este trabajo de tesis aborda el problema de selección de características como una tarea de optimización multi-objetivo, motivada por el aumento en la dimensionalidad de los conjuntos de datos utilizados en aplicaciones modernas de aprendizaje automático. El objetivo es identificar subconjuntos reducidos de características que mejoren el desempeño en clasificación. Para ello, se propone una adaptación del algoritmo evolutivo NSGA-II que integra métodos de selección de características basados en filtros, específicamente la prueba estadística de chi-cuadrado, incorporada en la inicialización, la mutación y a través de un proceso de reducción de dimensionalidad. La propuesta fue evaluada utilizando veinte conjuntos de datos de distintas áreas, con un rango de entre 16 y 12,600 características, analizando el impacto de distintos escenarios de proporciones iniciales de características. Los resultados muestran que las estrategias propuestas superan al método base en la mayoría de los conjuntos evaluados en términos de hipervolumen, particularmente al utilizar un 10 % de características iniciales, manteniendo tiempos de cómputo comparables a la versión base de NSGA-II y obteniendo resultados competitivos frente a enfoques del estado del arte, lo que confirma la efectividad de la adaptación propuesta en escenarios de alta dimensionalidad.

# 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 optimizes the classification accuracy. To this end, an adaptation of the evolutionary algorithm NSGA-II is proposed, integrating filter-based feature selection methods, specifically the chi-square statistical test, incorporated into the initialization, mutation, and a dimensionality reduction process. The proposed approach was evaluated using twenty datasets from different domains, with dimensionalities ranging from 16 to 12,600 features, analyzing the impact of different initial feature selection rates. The results show that the proposed strategies outperform the baseline method on the majority of the evaluated datasets in terms of hypervolume, particularly when using 10 % of initial features, while maintaining computational times comparable to the base version of NSGA-II and achieving competitive results with respect to state-of-the-art approaches, thereby confirming the effectiveness of the proposed adaptation in high-dimensional scenarios.

# Índice de Contenidos

# Lista de Tablas

# Lista de Algoritmos

# Lista de Figuras

# Capítulo 1

# 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 [1].

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 [2]. 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 [3]. 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 [4], el cual propone una versión compacta del algoritmo para reducir el consumo de memoria, y [5], el cual adapta la inicialización y los operadores basándose en ReliefF [6].

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. Otro desafío importante corresponde a los operadores de transformación, ya que el desempeño del algoritmo depende de su eficacia a lo largo de las generaciones.

Para abordar estos problemas, se presenta un nuevo enfoque de NSGA-II que incorpora técnicas de filtro para mejorar el tiempo computacional y la adaptabilidad en problemas con conjuntos de datos de diversos tamaños y contextos. Cabe resaltar que, para este trabajo, el método de filtro utilizado corresponde a la prueba estadística chi-cuadrado, una técnica que permite identificar las características que tienen una mayor dependencia con la variable objetivo [7].

El uso de chi-cuadrado en el proceso de inicialización tiene como objetivo reducir la dimensionalidad 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. Se proponen también tres inicializaciones con individuos que poseen distintos porcentajes de características iniciales entre sí. Por último, se propone una reducción de características a través de chi-cuadrado, la cual se aplica previo a la etapa de inicialización del algoritmo, con el objetivo de disminuir la dimensionalidad del problema y facilitar el proceso de búsqueda evolutiva.

Con respecto a los operadores de transformación, se propone una variante del operador de mutación bit-flip que limita la variación de las características a aquellas más relevantes según la prueba estadística de chi-cuadrado, con el fin de reducir la aleatoriedad de la mutación tradicional y orientar la búsqueda hacia regiones más informativas del espacio de soluciones. Dado que este operador depende de información obtenida durante la inicialización, se evalúa únicamente en combinación con la inicialización basada en chi-cuadrado y la inicialización híbrida.

## 1.1. Hipótesis y Objetivos

Para este trabajo se plantea la siguiente hipótesis:

- El uso de nuevas estrategias de inicialización y operadores de mutación guiados que incorporen métodos de selección de características basados en filtros puede mejorar el rendimiento de los conjuntos de datos etiquetados en términos de reducción de dimensionalidad y error de clasificación, sin aumentar significativamente el tiempo de cálculo.

### 1.1.1. Objetivo general

Para poder comprobar la hipótesis planteada, se establece el siguiente objetivo general:

- Adaptar el algoritmo NSGA-II para resolver el problema multi-objetivo de selección de características.

### 1.1.2. Objetivos específicos

Para cumplir con el objetivo general, se establecen los siguientes objetivos específicos:

- Proponer estrategias de inicialización y operadores de mutación guiados que utilicen métodos de selección de características basados en filtros en el algoritmo NSGA-II para resolver el problema de selección de características.
- Evaluar el rendimiento de las adaptaciones propuestas utilizando conjuntos de datos de distintas áreas y con diversas dimensionalidades, y compararlos con una base de NSGA-II.
- Comparar el rendimiento de las adaptaciones propuestas con enfoques del estado del arte.

## 1.2. Contribuciones

Las contribuciones principales de este trabajo son:

- Una estrategia de inicialización para NSGA-II que integra métodos de selección de características basados en filtros bien conocidos de la bibliografía.
- Una propuesta de reducción de dimensionalidad para cada uno de los problemas mediante de métodos de filtro.
- Un proceso de mutación guiado a través de métodos de filtro para equilibrar la exploración y la explotación.
- Una evaluación experimental de los enfoques propuestos junto con el análisis del impacto de distintos porcentajes de selección de características iniciales.

- Una experimentación que tiene en cuenta conjuntos de datos del mundo real de diferentes dominios con diversos números de características y muestras.

## 1.3. Estructura de la tesis

Esta tesis está estructurada en los siguientes capítulos con el detalle que se indica a continuación:

1. Definición del problema: Se abordan teóricamente los problemas de selección de características y optimización multi-objetivo.
2. Estado del arte: 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: Se presentan el algoritmo NSGA-II y las variantes de inicialización propuestas. También se detallan la representación, las funciones objetivo y los operadores de transformación utilizados.
4. Validación de la solución: Se presentan los conjuntos de datos, los valores de parámetros utilizados y el algoritmo de aprendizaje automático empleado en los experimentos.
5. Resultados: Se presentan los resultados obtenidos de los experimentos a través de gráficos de frentes de Pareto e indicadores de calidad como hipervolúmenes y tiempos de cómputo.
6. Conclusiones: Se sintetizan los análisis realizados para extraer las conclusiones más relevantes y se comenta sobre posibles trabajos futuros.

# Capítulo 2

# 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.

## 2.1. 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 [2].

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 [1].

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 [2].

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 2.1, 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 ($c0$) que contiene $N$ clases. El objetivo es utilizar las características para predecir correctamente las clases de $c0$. Las columnas pueden ser binarias, como las columnas $c1$ y $c2$, y pueden ser multiclase, como las columnas $c3$ y $c4$. Además, existen columnas que pueden estar desbalanceadas, como es el caso de la columna $c5$.

Dado que un conjunto de datos con $D$ características posee $2^D$ posibles subconjuntos, el espacio 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 [2].

Una vez que se escoge el conjunto de datos que se busca optimizar, se realiza el proceso que se observa en la figura 2.2. Primero se obtiene un subconjunto de características del conjunto original; luego se entrena un modelo de clasificación utilizando el subconjunto

| 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 |

Figura 2.1: Ejemplo de representación de un conjunto de datos.

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 sí toman en cuenta la precisión obtenida para decidir el subconjunto de características que se entrega al final del algoritmo [2].

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 [8].

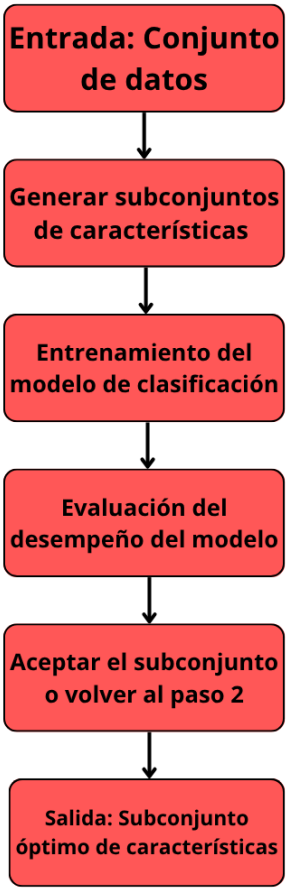



Figura 2.2: Proceso general de selección de características.
Fuente: Elaboración propia.

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 menos memoria y se 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 perjudicar el desempeño de la etapa de selección, dado que se podría perder información relevante con respecto a la interacción entre características [8].

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 [1].

## 2.2. 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 $x^* = [x_1^*, x_2^*, ..., x_n^*]^T$ que satisfacen las restricciones establecidas y optimizan cada función objetivo, las cuales se representan de la siguiente

forma:

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

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 $x^* = [x_1^*, x_2^*, ..., x_n^*]^T$ domina a la solución $y^* = [y_1^*, y_2^*, ..., y_n^*]^T$ si se cumple lo siguiente:

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

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 2.3 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.

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 [9].

Los algoritmos evolutivos han sido muy utilizados en la resolución de problemas multiobjetivo. 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

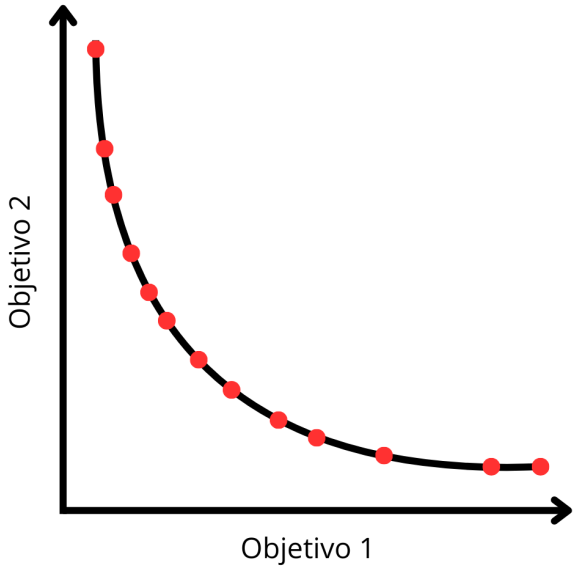



Figura 2.3: Ejemplo de representación de un frente de Pareto.
Fuente: Elaboración propia.

sobreviven generación tras generación. Este concepto puede aplicarse en problemas de optimización en donde se busque minimizar o maximizar una función objetivo [10]. 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 [3], el cual permite mantener una convergencia hacia el frente de Pareto [11].

Para asegurar que se obtengan buenos resultados, existen diversos indicadores de calidad multi-objetivo. Entre ellos se encuentra el hipervolumen [12]. 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 presenta en la ecuación (2.1).

$$\mathcal{S}(A, y_{\text{ref}}) = \Lambda\left(\cup_{y \in A} y' \,\middle|\, y \prec y' \prec y_{\text{ref}}\right), \quad A \subseteq \mathbb{R}^m \tag{2.1}$$

, donde $A$ corresponde al subconjunto del espacio de objetivos, $y_{\text{ref}}$ al punto de referencia dominado por todas las soluciones y $\Lambda$ a la medida de Lebesgue. En la figura 2.4 se presenta un ejemplo simple de hipervolumen. El área de color gris representa el hipervolumen (en dos dimensiones) computado en este ejemplo.

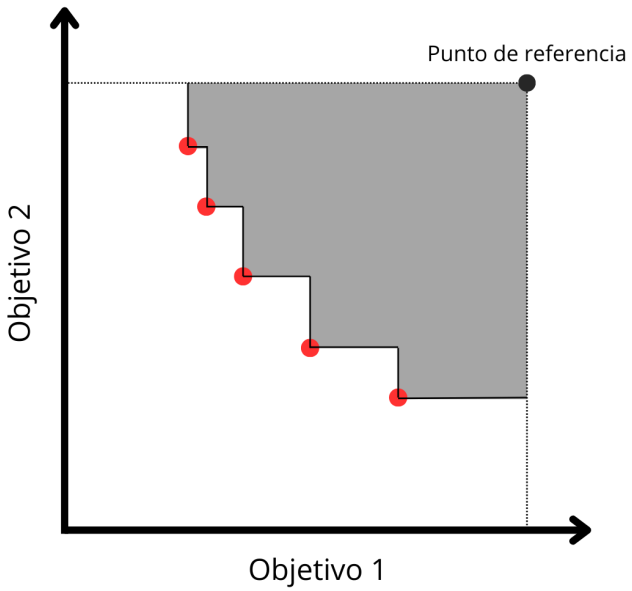



Figura 2.4: Ejemplo de representación de la medida de hipervolumen.
Fuente: Elaboración propia.

Otro indicador utilizado para este problema es la distancia generacional invertida [12], 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 (2.2).

$$IGD_p = \frac{\left(\sum_{i=1}^{n} d_i^p\right)^{1/p}}{n} \tag{2.2}$$

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 2.5 se observa un ejemplo simple de la distancia generacional invertida.

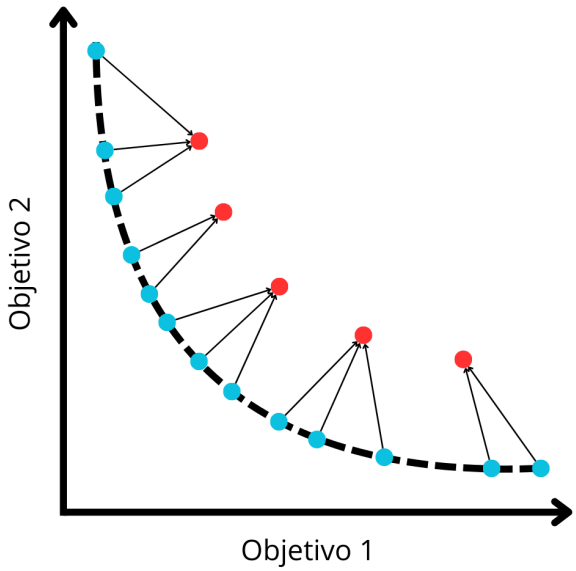



Figura 2.5: Ejemplo de representación de la medida de distancia generacional invertida, donde las soluciones de color azul son parte del conjunto óptimo de Pareto y las de color rojo representan las soluciones no dominadas que se obtuvieron.

Fuente: Elaboración propia.

# Capítulo 3

# 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.

## 3.1. Acercamientos de algoritmos de selección de características

Los algoritmos de selección de característica de la literatura se dividen en tres grandes categorías [2]: modelos de filtro, modelos de envoltura y modelos integrados. A continuación se describe cada clasificación de modelos y algunos trabajos relevantes en el área.

### 3.1.1. 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 [2]. Una de las variantes frecuentemente utilizadas para abordar el problema de clasificación de características es el filtro rápido basado en correlación [13]. 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 [6], una variación de un método basado en correlación (CorrSF) [14] y una variación de un método basado en consistencia (ConsSF) [15].

Otro enfoque común dentro de los modelos de filtro es la prueba estadística de chi-cuadrado ($\chi^2$), 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) [16].

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 si, 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 [7]. 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 3.1:

$$\chi^2 = \sum \frac{(O - E)^2}{E} \tag{3.1}$$

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 3.1. Para poder calcular el valor de las frecuencias esperadas, se utiliza la fórmula 3.2:

$$E_{ij} = \frac{T_i \cdot T_j}{N} \tag{3.2}$$

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 3.1 se calcula para cada una de las características con respecto a la clase objetivo. Es por esto que la complejidad del algoritmo es $O(c_0 \cdot D)$ [7].

Estudios como [16] y [7] 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 [17], 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 [18] y la información mutua normalizada [19, 20]. 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 $\lambda$ 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 [21] y KEEL [22]. 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 [14], búsqueda basada en consistencia [23] e información mutua [19], y se utilizó k-nearest neighbors [24] [25], support vector machine [26] y árboles de decisión [27] 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 informacińn 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.

### 3.1.2. 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 [2].

Una comparación más detallada entre los modelos de filtro y envoltura se presenta en [28], donde el modelo de envoltura se implementó utilizando la técnica secuencial hacia adelante [29] 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 [30], NB [31], C4.5 [32] y CN2 [33]. 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 [24, 25].

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. [24].

Para comprender mejor el funcionamiento de k-NN, se presenta un ejemplo básico en la figura 3.1. 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.

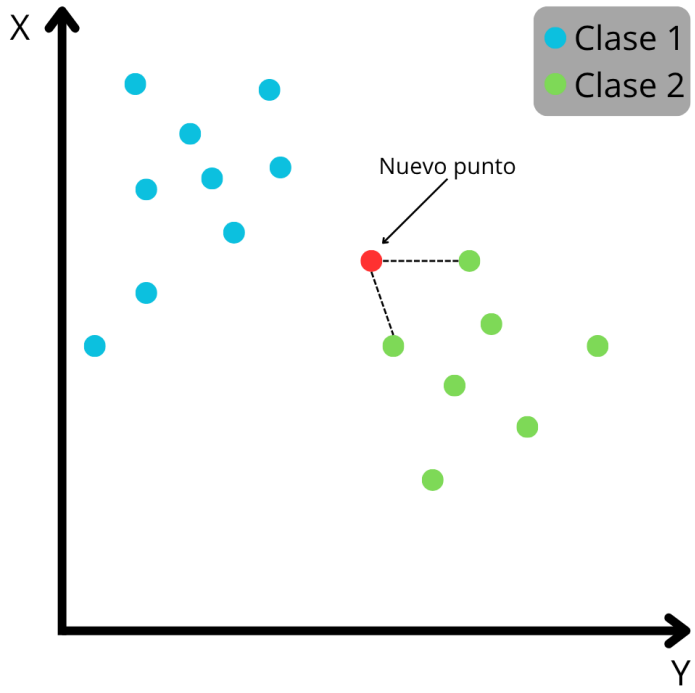



Figura 3.1: Ejemplo de k-NN con k=2.

Fuente: Elaboración propia.

### 3.1.3. 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 [2]. En el segundo tipo se encuentran los modelos con un mecanismo incorporado para la selección de características, como ID3 [34] y C4.5 [32] 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 [35].

## 3.2. 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 [4] se propone Compact NSGA-II (CNSGA-II), el cual corresponde a una versión modificada de NSGA-II [3] 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 $\boldsymbol{D}$ 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 (3.3) corresponde al error de clasificación y la ecuación (3.4) corresponde al porcentaje de características seleccionadas. Estas representan los dos objetivos conflictivos de la sección 2.1.

$$\text{Error de Clasificación} = 1 - \frac{\text{Predicciones Correctas}}{\text{Total de Predicciones}} \tag{3.3}$$

$$\text{Porcentaje de Características Seleccionadas} = \frac{\text{Cantidad de 1s}}{\text{Total de características}} \tag{3.4}$$

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

1. Inicialización: Se inicializan $\lambda$ 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 $\lambda$, 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 $\lambda$ 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 [36].

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 [37] 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 clásiso 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 [5]. 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 clases, entonces el peso de la característica aumenta. Para los detalles matemáticos, revisar en [5].

El algoritmo propuesto considera dos funciones objetivo. Estas representan los dos objetivos conflictivos de la sección 2.1. La ecuación (3.5) corresponde al error de clasificación. En esta, X es la solución, K el número de particiones en la validación cruzada, y $N^l_{Error}$ y $N^l_{All}$ la cantidad de particiones mal predichas y el número de particiones en el conjunto de datos $l$.

$$\text{mín } f_1(X) = \left( \frac{1}{K} \sum_{l=1}^{K} \frac{N^l_{Error}}{N^l_{All}} \right) \times 100\,\% \tag{3.5}$$

La función (3.6) corresponde a la minimización de la cantidad de características seleccionadas, donde cada $x_i$ representa

$$\text{mín } f_2(X) = \sum_{i=1}^{D} x_i \tag{3.6}$$

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 extraidos del UCI Machine Learning Repository [21].

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 [38] se presenta un acercamiento basado en Particle swarm optimization [39]. 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 movilizarse. 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 [38].

En [38] se propone una variante de PSO llamada RFPSOFS (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 [3] y tres variantes multi-objetivo de PSO, las cuales son CMDPSOFS [40], HMPSOFS [41] y MOPSO [42] utilizando 9 conjuntos de datos obtenidos de la UCI machine learning repository [21]. Se utilizaron varios indicadores de desempeño, entre ellos el hipervolumen. Los resultados obtenidos mostraron que RFPSOFS 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 [43]. Funciona mediante una población de vectores (soluciones) que evoluciona a través de las generaciones mediante operadores de mutación, cruzamiento y selección.

A partir del análisis del estado del arte, se observa que, si bien se han realizado avances significativos en el problema de selección de características, aún persisten desafíos relevantes en el diseño de los algoritmos que guían la búsqueda evolutiva. Tanto la inicialización de la población como los operadores de mutación son aspectos que continúan estando abiertos, especialmente en escenarios de alta dimensionalidad. Se han explorado estrategias guiadas por métodos de filtro para influir en estas etapas. Sin embargo, no existe un único enfoque que permita equilibrar de manera efectiva la calidad del frente de Pareto, la diversidad de soluciones y el costo computacional. En este contexto, resulta necesario profundizar en estrategias que integren información proveniente de métodos de filtro simples y robustos, como pruebas estadísticas, con el objetivo de orientar de manera más eficiente el proceso evolutivo desde sus etapas iniciales y durante su evolución. Esta necesidad motiva la propuesta de este trabajo, que busca contribuir al problema de selección de características multi-objetivo mediante nuevos algoritmos de inicialización y mutación basados en la prueba estadística de chi-cuadrado, abordando así un desafío que permanece abierto en la literatura.

# Capítulo 4

# 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.

## 4.1. Non-Dominated Sorting Genetic Algorithm II (NSGA-II)

Non-Dominated Sorting Genetic Algorithm II es un algoritmo evolutivo multi-objetivo propuesto en [3]. 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 ($\lambda$), la probabilidad de cruzamiento $p_c$, la probabilidad

de mutación $p_m$ y la cantidad de generaciones *MaxGen*.

Algoritmo 1 NSGA-II

Input: $\lambda$, $p_c$, $p_m$, *MaxGen*

Output: frente de Pareto $F$

1: $g = 0$
2: $P_g$ = Generar población inicial($\lambda$)
3: $Q_g$ = Generar descendencia usando selección, cruzamiento y mutación($\lambda, p_c, p_m$)
4: while $g <$ *MaxGen* do
5: $R_g$ = Combinar $P_g \cup Q_g$
6: $F$ = Clasificar $R_g$ en frentes no dominados ($F_1, F_2, ...$)
7: $P_{g+1}$ = Obtener nueva población($\lambda$)
8: $Q_{g+1}$ = Generar descendencia usando selección, cruzamiento y mutación($\lambda, p_c, p_m$)
9: $g = g + 1$
10: end while
11: $R_g$ = Combinar $P_g \cup Q_g$
12: $F$ = Clasificar $R_g$ en frentes no dominados ($F_1, F_2, ...$)
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 linea 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 4.1 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.

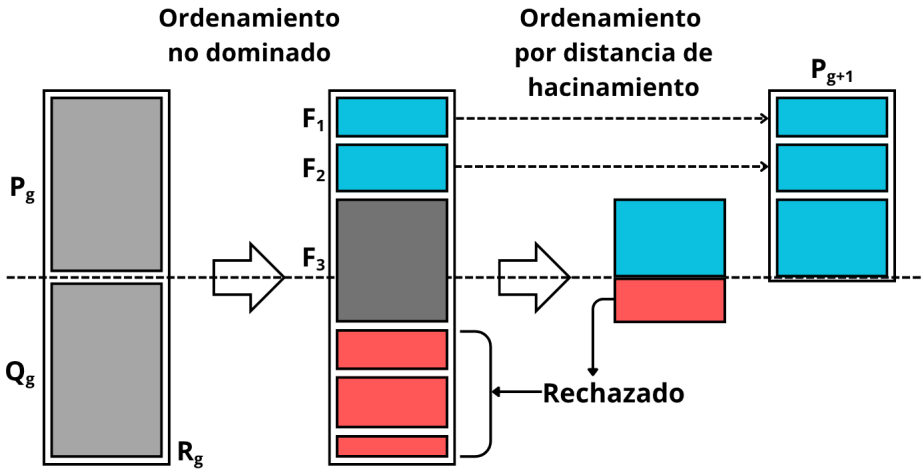



Figura 4.1: Ejemplo visual del proceso elitista en NSGA-II.

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 (4.1), 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.

$$\mathrm{DH}_i = \sum_{m=1}^{M} \frac{f_{i+1}^{(m)} - f_{i-1}^{(m)}}{f_{\text{máx}}^{(m)} - f_{\text{mín}}^{(m)}} \tag{4.1}$$

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.

## 4.2. 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 4.2 se aprecia visualmente un ejemplo

de solución donde «Car.» significa «Característica».

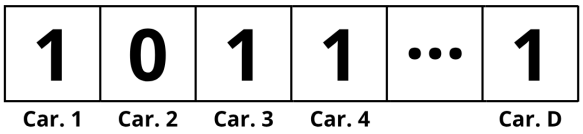



Figura 4.2: Ejemplo de solución utilizando la representación.
Fuente: Elaboración propia.

## 4.3. Funciones Objetivo

Las ecuaciones (4.2) y (4.3) presentan las dos funciones objetivo implementadas. En la ecuación (4.2), $C$ corresponde al número total de clases, $FP_i$ representa los valores que son falsos positivos en la clase $C$, y $S_i$ representa todos los valores en la clase $C$. Por otro lado, en la ecuación (4.3), #características_seleccionadas corresponde al número de características utilizadas.

$$F_1 = \frac{\sum_{i=1}^{C} FP_i}{\sum_{i=1}^{C} S_i} \tag{4.2}$$

$$F_2 = \#\text{características\_seleccionadas} \tag{4.3}$$

## 4.4. Inicialización

Se presentan seis métodos de inicialización. La inicialización que considera la generación de soluciones iniciales aleatorias (original de NSGA-II), la inicialización con chi-cuadrado y la inicialización que mezcla ambas. Las otras tres inicializaciones corresponden a variaciones de cada una de las mencionadas anteriormente, donde explícitamente se determina la cantidad de características iniciales que posee cada individuo. Los procesos se explican en detalle a continuación.

### 4.4.1. 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 [3].

### 4.4.2. Inicialización basada en 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 ($\chi^2$) [7]. 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 3.1.1. 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, *prob* 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 $\chi^2$ 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 $\chi^2$, 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 $\chi^2$.

En la línea 6 se inicializa el primer individuo de la población con $M_O$. 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.

Algoritmo 2 Inicialización basada en chi-cuadrado

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

Output: 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($Chi$, $j$)
4: $M_0$ = Crear vector binario con las características seleccionadas($Chi_j$)
5: $M$ = Crear 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: $var$ = 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: $P_i$ = Mutar utilizando posiciones aleatorias y distintas(*var*,*limite*,$M$)
12: end for
13: return $P$

### 4.4.3. 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.

Algoritmo 3 Inicialización híbrida

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

Output: 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 vector con los índices de las características de la mejor a la peor($Chi_j$)
6: for $i = 0$ to $\lambda$ do
7: $\quad P_i = M_0$
8: $\quad P_i$ = Seleccionar nuevas características que no hayan sido seleccionadas($j/2$)
9: end for
10: return $P$

### 4.4.4. Variante aleatoria basada en diversidad de características

Para explorar los posibles resultados iniciales al generar diversos individuos, se propone una variante de la inicialización aleatoria. Esta propuesta genera individuos con porcentajes que se encuentran dentro de la lista de porcentajes $Ran = [0.1, 0.2, 0.3, ..., 1]$, los cuales controlan la proporción de características seleccionadas en cada individuo. Por esta misma razón, solo se puede utilizar un tamaño de población que sea múltiplo de 10, ya que la cantidad de individuos para cada porcentaje debe ser equitativa. En

el algoritmo 4 se presenta el proceso de inicialización.

En la línea 1 se calcula la cantidad de individuos a la que se les asignará un porcentaje dentro de ***Ran***. Luego, en la línea 2, se inicializa la población con ceros. Entre las líneas 3 y 11 se encuentran tres bucles. El primero busca iterar por cada uno de los porcentajes de la lista ***Ran***, el segundo busca abordar cada uno de los individuos asociados al porcentaje, y el tercero busca iterar por cada individuo. Entre las líneas 6 y 8 se realiza la asignación de características a cada individuo al azar, donde en 6 revisa si cada característica debe ser asignada y en 7 se asigna la característica en caso de haber cumplido con la condicional. Una vez que termina el primer ciclo, se retorna la población inicial en la línea 12.

Algoritmo 4 Variante aleatoria basada en diversidad de características

Input: $D$, $\lambda$, ***Ran***

Output: Población inicial $P$

```
1:  num = Número de individuos por porcentaje(λ/len(Ran))
2:  P = Inicializar población de tamaño λ con ceros en cada individuo
3:  for i = 0 to len(Ran) − 1 do
4:      for j = 0 to num − 1 do
5:          for k = 0 to D − 1 do
6:              if random(0,1) ≤ Ran[i] then
7:                  P_(i×num)+j[k] = 1
8:              end if
9:          end for
10:     end for
11: end for
12: return P
```

### 4.4.5. Variante de chi-cuadrado basada en diversidad de características

Otra de las variantes propuestas es la variante de la inicialización basada en chi-cuadrado. Al igual que la variante anterior, esta utiliza la lista de porcentajes ***Ran***.

La diferencia está en que la selección se realiza mediante los procedimientos vistos en el algoritmo 2.

En el algoritmo 5 se presenta el pseudocódigo de la variante de chi-cuadrado basada en diversidad de características. En este, como se mencionó previamente, se utilizan los procedimientos de la inicialización basada en chi-cuadrado, lo cual se puede apreciar entre las líneas 1 y 2. La diferencia radica en que no se genera la variable $\boldsymbol{M_0}$ que se utilizaba antes, sino que aquí se trabaja directamente desde el vector $\boldsymbol{M}$. Se realiza el mismo procedimiento que la variante anterior para el número de individuos por porcentaje y la inicialización de la población en las líneas 4 y 5. Entre las líneas 6 y 13 se realizan tres bucles, donde la única diferencia con la anterior propuesta es que se calcula la cantidad de características que se van a asignar al individuo en la línea 7 para luego asignar las características más relevantes hasta esa cantidad entre las líneas 9 y 11. Una vez que termina el primer ciclo, se retorna la población inicial en la línea 14.

### 4.4.6. Variante híbrida basada en diversidad de características

La última variante propuesta es la variante de la inicialización híbrida. Al igual que las variantes anteriores, se utiliza la lista de porcentajes $\boldsymbol{Ran}$. La diferencia está en que se asigna la mitad de las características más relevantes y la otra mitad se obtiene de forma aleatoria.

En el algoritmo 6 se encuentra el pseudocódigo de la variante híbrida basada en diversidad de características. En este, como se mostró previamente, de la línea 1 a la línea 5 se realiza el mismo procedimiento que en el algoritmo 5. De la línea 6 a la línea 14 se realizan los mismos tres ciclos. Las únicas diferencias son que en la línea 7 se utiliza la mitad de las características más relevantes y que en la línea 12 se añade la otra mitad a cada individuo de manera aleatoria.

Algoritmo 5 Variante de chi-cuadrado basada en diversidad de características

Input: $X$, $y$, $n$, $D$, $\lambda$, $Ran$

Output: Población inicial $P$

1: $X_{Disc}$ = Discretizar la data($X$, $n$)
2: $Chi$ = Calcular y normalizar los puntajes $\chi^2(X_{Disc}, y)$
3: $M$ = Crear un vector con los índices de las características ordenadas de mayor a menor puntaje($Chi$)
4: $num$ = Número de individuos por porcentaje($\lambda/len(Ran)$)
5: $P$ = Inicializar población de tamaño $\lambda$ con ceros en cada individuo
6: for $i = 0$ to $len(Ran) - 1$ do
7: $\quad rel = Ran[i] \times D$
8: $\quad$ for $j = 0$ to $num - 1$ do
9: $\quad\quad$ for $k = 0$ to $rel - 1$ do
10: $\quad\quad\quad P_{(i*num)+j}[M[k]] = 1$
11: $\quad\quad$ end for
12: $\quad$ end for
13: end for
14: return $P$

## 4.5. Propuesta: Reducción de características a través de chi-cuadrado

Otra de las propuestas desarrolladas en esta memoria corresponde a la reducción de la dimensionalidad del problema mediante el uso de chi-cuadrado, la cual se realiza previo a la etapa de inicialización del algoritmo. Para este propósito, se utilizan los valores normalizados de chi-cuadrado obtenidos para cada característica, los cuales permiten estimar su relevancia respecto a la variable objetivo.

A partir de estos valores, se define un umbral en el intervalo $[0, 1]$ que determina qué características serán consideradas en el proceso, descartando aquellas cuyo valor normalizado de chi-cuadrado se encuentre por debajo del umbral. Una vez realizada esta reducción inicial del conjunto de características, el algoritmo NSGA-II [3] se ejecuta

Algoritmo 6 Variante híbrida basada en diversidad de características

Input: $X$, $y$, $n$, $D$, $\lambda$, $Ran$

Output: Población inicial $P$

1: $X_{Disc}$ = Discretizar la data($X$, $n$)
2: $Chi$ = Calcular y normalizar los puntajes $\chi^2(X_{Disc}, y)$
3: $M$ = Crear un vector con los índices de las características ordenadas de mayor a menor puntaje($Chi$)
4: $num$ = Número de individuos por porcentaje($\lambda/len(Ran)$)
5: $P$ = Inicializar población de tamaño $\lambda$ con ceros en cada individuo
6: for $i = 0$ to $len(Ran) - 1$ do
7: $\quad rel = Ran[i]/2 \times D$
8: $\quad$ for $j = 0$ to $num - 1$ do
9: $\quad\quad$ for $k = 0$ to $rel - 1$ do
10: $\quad\quad\quad P_{(i*num)+j}[M[k]] = 1$
11: $\quad\quad$ end for
12: $\quad\quad P_{(i*num)+j}$ = Seleccionar características no seleccionadas($Ran[i]/2$)
13: $\quad$ end for
14: end for
15: return $P$

sin modificaciones adicionales, utilizando la inicialización y los operadores de transformación estándar durante el resto de la ejecución.

En el algoritmo 7 se encuentra el pseudocódigo de la nueva propuesta. En las líneas 1 y 2 se obtienen los valores de chi-cuadrado normalizados para cada una de las características. Luego de eso, en la línea 3, se crea el vector que contiene todas las características que se encuentran por encima del umbral de chi-cuadrado. Se utiliza este vector en la línea 4 para calcular la nueva cantidad de características. Esta se asigna a la variable $D$ en la línea 5 para mostrar que ahora se estará trabajando con una nuevo número de características, donde cabe resaltar que $D$ se considera como una variable global. Por último, se retorna el vector $M_U$ en la línea 6 ya que se estará utilizando esta lista durante el resto del algoritmo para evitar que los individuos utilicen características que

se encuentren por debajo del umbral establecido.

Algoritmo 7 Reducción de características a través de chi-cuadrado

Input: $X$, $y$, $n$, $D$, $umb$

Output: Población inicial $P$

1: $X_{Disc}$ = Discretizar la data($X$, $n$)
2: $Chi$ = Calcular y normalizar los puntajes $\chi^2(X_{Disc}, y)$
3: $M_U$ = Crear un vector con los índices de las características que cumplen con el umbral ordenadas de mayor a menor puntaje ($Chi$, $umb$)
4: $U$ = Calcular la cantidad de características posterior a la reducción($M_U$)
5: $D = U$
6: return $M_U$

## 4.6. Operadores de transformación

### 4.6.1. Operador de selección

Para la selección se utiliza un torneo binario empleando como criterio dominancia y hacinamiento, tal como en [3]. 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 al 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 8, 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 a las parejas para generar dos descendientes. La aplicación del proceso de selección de padres y cruzamiento requiere siempre poblaciones de tamaño múltiplo de cuatro.

Algoritmo 8 Selección por torneo binario con cruzamiento

Input: Población actual $P$ de tamaño $\lambda$

Output: Nueva población $Q$ de tamaño $\lambda$

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

### 4.6.2. 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 [3]. Este es un operador clásico para representación binaria.

En el algoritmo 9 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.

### 4.6.3. Operadores de mutación

Se implementan dos mutaciones para utilizar durante el proceso de NSGA-II. La primera corresponde a la mutación tradicional y la segunda, la propuesta basada en chi-cuadrado implementada.

La primera corresponde a la mutación bit-flip. Esta es tal como en [3], donde cada bit corresponde a una característica. Este operador recorre cada una de las características

Algoritmo 9 Cruzamiento binario de dos puntos

Input: $p_1$, $p_2$, $p_c$, $D$

Output: $c_1$, $c_2$

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

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 [44].

En el algoritmo 10 se muestra la implementación, donde *ind* corresponde al individuo. Este procedimiento se aplica para cada solución de la población.

Algoritmo 10 Mutación bit-flip

Input: $ind$, $p_m$, $D$

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

La segunda corresponde a una nueva mutación bit-flip basada en chi-cuadrado. La idea de este nuevo operador es mejorar a los individuos durante el proceso del algoritmo NSGA-II al ir variando únicamente las características más relevantes a través de la prueba estadística chi-cuadrado.

La motivación detrás de esta propuesta surge a partir de la observación de que la mutación bit-flip tradicional introduce variaciones de forma completamente aleatoria, sin considerar la relevancia de las características respecto a la clase objetivo. En problemas de selección de características con alta dimensionalidad, este comportamiento puede generar modificaciones poco relevantes, afectando la calidad de los individuos. Por este motivo, se propone utilizar chi-cuadrado para enfocar la mutación en las características con mayor relevancia.

Este operador realiza el mismo procedimiento que la mutación aleatoria con la única diferencia de que, en vez de aplicar la probabilidad a cada una de las características de cada individuo, se aplica la probabilidad a las características más relevantes. Por esta razón, se utiliza el vector $M$ creado durante la ejecución de los algoritmos 2 y 3.

En el algoritmo 11 se muestra la implementación, donde $M[j]$ corresponde al índice de la característica a la que se le aplicará la mutación si es que se cumple la condicional. Este procedimiento se aplica a cada solución de la población.

Algoritmo 11 Mutación bit-flip basada en chi-cuadrado

Input: $ind$, $p_m$, $D$, $percent$, $M$

1: $limite = D * percent$
2: for $j = 0$ to $limite$ do
3: if random(0,1) $\leq p_m$ then
4: Invertir $ind[M[j]]$: $1 \leftrightarrow 0$
5: end if
6: end for

# Capítulo 5

# Validación de la Solución

En este capítulo se presentan los conjuntos de datos utilizados para la validación de la propuesta. Se presentan las configuraciones experimentales y las métricas de calidad. Se presenta el algoritmo de predicción utilizado. Por último, se especifican las pruebas que se realizarán previo a los resultados.

## 5.1. Conjuntos de datos

Se utilizaron veinte conjuntos de datos para los experimentos, los cuales se listan en la tabla 5.1. Para cada conjunto se presenta el nombre, cantidad de características, cantidad de muestras y cantidad de clases.

Los primeros diez 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, abarcando distintos tipos de cáncer, tumores y algunos tejidos normales. Estos conjuntos fueron utilizados en [45] y se encuentran disponibles en https://github.com/primekangkang/Genedata.

Tabla 5.1: Conjuntos de datos

| N° | Nombre | Características (D) | Muestras | Clases (N) |
|---|---|---|---|---|
| 1 | SRBCT | 2,308 | 83 | 4 |
| 2 | DLBCL | 5,469 | 77 | 2 |
| 3 | 9Tumors | 5,726 | 60 | 9 |
| 4 | Leukemia1 | 5,327 | 72 | 3 |
| 5 | Leukemia2 | 11,225 | 72 | 3 |
| 6 | Brain Tumor1 | 5,920 | 90 | 5 |
| 7 | Brain Tumor2 | 10,367 | 50 | 4 |
| 8 | Prostate Tumor | 10,509 | 102 | 2 |
| 9 | Lung Cancer | 12,600 | 203 | 5 |
| 10 | 11Tumors | 12,533 | 174 | 11 |
| 11 | warpAR10P | 2,400 | 130 | 10 |
| 12 | warpPIE10P | 2,420 | 210 | 10 |
| 13 | TOX-171 | 5,748 | 171 | 4 |
| 14 | pixraw10P | 10,000 | 100 | 10 |
| 15 | CLL-SUB-111 | 11,340 | 111 | 3 |
| 16 | Chess | 36 | 3,196 | 2 |
| 17 | Coil2000 | 85 | 9,822 | 2 |
| 18 | Penbased | 16 | 10,992 | 10 |
| 19 | Segment | 19 | 2,310 | 7 |
| 20 | Texture | 40 | 5,500 | 11 |

Los conjuntos numerados desde el 11 al 15 provienen de las áreas de biología y reconocimiento de imágenes. $TOX-171$ y $CLL-SUB-111$ provienen del área de la biología y se obtuvieron a través de microarrays, mientras que los conjuntos $warpAR10P$, $warpPIE10P$ y $pixraw10P$ corresponden al área del reconocimiento de imágenes o rostros. Estos conjuntos de datos fueron utilizados en [4] y se encuentran disponibles en [46].

Los últimos cinco conjuntos de datos abarcan varias áreas. El conjunto *chess* entra en

el área de los juegos, ya que representa las posiciones finales de una partida de ajedrez y se busca predecir si el jugador con las piezas blancas puede ganar. *Coil*2000 pertenece al área de marketing y análisis financiero, dado que está basado en la información de los clientes de una compañía de seguros. 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 último, los conjuntos *segment* y *texture* se pueden clasificar en el área de visión artificial, ya que el primero está basado en la clasificación de regiones segmentadas en imágenes exteriores y el segundo realiza distinciones entre tipos de texturas a través de características obtenidas de patrones visuales. Todos estos conjuntos de datos fueron utilizados en [17] 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 50 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.

Es importante mencionar que el conjunto de datos *chess*, a diferencia del resto, no presenta valores numéricos, sino que contiene únicamente variables con valores categóricos. Así, el conjunto de datos fue transformado debido a que el uso de los análisis basados en chi-cuadrado solo acepta valores numéricos. En la transformación propuesta, la variable objetivo que originalmente tiene los valores $[nowin, win]$ se codificó como $[0, 1]$. El resto de columnas presenta valores en el conjunto $[b, f, g, l, n, t, w]$, los cuales se reemplazaron por $[0, 1, 2, 3, 4, 5, 6]$. Una vez realizada la codificación, se descartó el preprocesamiento de discretización para *chess* en la inicialización con chi-cuadrado. La finalidad de esta última decisión fue mantener la integridad del conjunto y evitar la pérdida de información.

## 5.2. 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 5.2 se presentan los valores utilizados para los parámetros de NSGA-II, los cuales incluyen las mismas probabilidades de cruzamiento y mutación que en [3]. 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 2,800 ejecuciones entre las nueve propuestas. 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 [4].

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

Tabla 5.2: Parámetros de NSGA-II.

En la tabla 5.3 se presentan los valores utilizados para los parámetros de la inicialización aleatoria, la inicialización basada en chi-cuadrado y la inicialización híbrida. 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* [47]. 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 1,000 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 $\chi^2$ ($percent$) | 0.5 |

Tabla 5.3: Parámetros utilizados en las inicializaciones.

Cabe destacar que se realizarán dos pruebas distintas para la inicialización basada en chi-cuadrado y la inicialización híbrida. En una de ellas se utilizará únicamente la mutación bit-flip y en la otra se utilizará equitativamente tanto la mutación bit-flip como la mutación bit-flip basada en chi-cuadrado, con un 50 % de probabilidad de utilizarse cada una.

Con respecto a las variantes basadas en la diversidad de características de las primeras tres inicializaciones, sus resultados se consideran aparte y luego se comparan con el resto de los algoritmos. Esto se debe a que no entran a la categoría de 10 % o 50 % de características iniciales. Lo mismo ocurre con la propuesta de reducción de características de chi-cuadrado, ya que se utiliza el 50 % de las características iniciales y se realiza el proceso al igual que el algoritmo NSGA-II base, pero conteniendo una menor cantidad de características desde un inicio. Además, el valor que se utiliza para la variable ***umb*** en el algoritmo 7 es 0.1. Este valor fue obtenido a través de experimentos preliminares y a través de la observación de los valores de chi-cuadrado normalizados para cada uno de los conjuntos de datos.

## 5.3. Métricas de evaluación

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 2.2. Se utilizó el código presentado e implementado en [48, 49]. 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 de la desviación estándar del hipervolumen obtenido en [4] para los conjuntos $warpAR10P$ y $warpPIE10P$, asumiendo que se utilizó el mismo punto de referencia ya que no se menciona en el artículo y utilizando los resultados obtenidos con el 50 % de características iniciales.

## 5.4. Algoritmo de predicción

Para evaluar el acercamiento propuesto, se utiliza el algoritmo k-NN, el cual está detallado en la sección 3.1.2, junto con Stratified K-Fold Cross-Validation (SKCV) [50]. 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 (5.1).

$$d(x, y) = \sqrt{\sum_{i=1}^{M}(x_i - y_i)^2} \tag{5.1}$$

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 [5].

Se realizaron unas variaciones al algoritmo de k-NN debido a problemas presentados en experimentos preliminares utilizando los conjuntos de datos. La tabla 5.4 presenta información más detallada sobre cada conjunto de datos. El porcentaje de clase más pequeño 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 de datos en un subconjunto de entrenamiento y uno de pruebas, puede 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.

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 5.4, donde se puede apreciar que 12 de los 20 conjuntos de datos presentaban problemas.

Tabla 5.4: Información relevante de los conjuntos de datos.

| Conjunto de datos | %Clase más pequeña | %Clase más grande | Errores en Pruebas |
|---|---|---|---|
| SRBCT | 13.25 | 34.94 | 4 |
| DLBCL | 24.68 | 75.32 | 1 |
| 9Tumors | 3.33 | 15.00 | 10 |
| Leukemia1 | 12.50 | 52.78 | 6 |
| Leukemia2 | 27.78 | 38.89 | 1 |
| Brain Tumor1 | 4.44 | 66.67 | 8 |
| Brain Tumor2 | 14.00 | 30.00 | 6 |
| Prostate | 49.02 | 50.98 | 0 |
| Lung Cancer | 2.96 | 68.47 | 5 |
| 11Tumors | 3.45 | 15.52 | 10 |
| warpAR10P | 10.00 | 10.00 | 8 |
| warpPIE10P | 10.00 | 10.00 | 6 |
| TOX-171 | 22.81 | 26.32 | 0 |
| pixraw10P | 10.00 | 10.00 | 10 |
| CLL-SUB-111 | 9.91 | 45.95 | 0 |
| Chess | 47.78 | 52.22 | 0 |
| Coil2000 | 5.97 | 94.03 | 0 |
| Penbased | 9.60 | 10.41 | 0 |
| Segment | 14.29 | 14.29 | 0 |
| Texture | 9.09 | 9.09 | 0 |

# Capítulo 6

# Resultados y análisis

En este capítulo se presentan los resultados obtenidos al utilizar NSGA-II con cada una de las propuestas y se realizan tanto comparaciones como análisis entre estas. Se presentan primero los frentes de Pareto, luego los resultados de hipervolumen y, por último, se realiza una comparación con el algoritmo CNSGA-II. Para cada caso se separan los resultados en un 10 % y un 50 % de características iniciales. En lo que sigue, por claridad, a cada enfoque propuesto se le asigna un identificador único como se muestra en la tabla 6.1.

Es importante aclarar que en ambos casos también estarán presentes las propuestas VA, VC, VH y RC, ya que en las primeras tres no se establece un porcentaje de características iniciales y el último tiene un enfoque distinto.

## 6.1. Frentes de Pareto

En esta sección se muestran los frentes de Pareto obtenidos a través de las instancias para cada una de las propuestas. Es importante señalar que el eje x corresponde al porcentaje de error y el eje y corresponde a la cantidad de características seleccionadas, los cuales están asociados a las funciones objetivo presentadas en las ecuaciones (4.2) y (4.3) de la sección 4.3 del capítulo 4.

| Propuesta | Sigla |
|---|---|
| Inicialización aleatoria | Base |
| Inicialización basada en chi-cuadrado | IC |
| Inicialización híbrida | IH |
| Inicialización basada en chi-cuadrado + nueva mutación | ICM |
| Inicialización híbrida + nueva mutación | IHM |
| Variante aleatoria | VA |
| Variante de chi-cuadrado | VC |
| Variante híbrida | VH |
| Reducción de características | RC |

Tabla 6.1: Resumen de pruebas y sus respectivas siglas

### 6.1.1. 10 % de características iniciales

En las figuras 6.1 y 6.2 se presentan los frentes de Pareto obtenidos en el proceso de entrenamiento y el proceso de prueba, donde cada figura corresponde a una instancia. El algoritmo ***Base*** está representado con círculos rellenos rojos, la propuesta ***IC*** está representada con equis verdes, la ***IC*** con un símbolo más azul, la ***ICM*** con cuadrados rosados, la ***IHM*** con círculos grises, la ***VA*** con líneas verticales color café, la ***VC*** con rombos negros, la ***VH*** con asteriscos amarillos y la ***RC*** con líneas horizontales color naranjo.

En las figuras 6.1(a), 6.1(b), 6.1(e), 6.1(f) y 6.1(ñ) se observa cómo ***RC*** (líneas horizontales de color naranja) domina completamente al resto, obteniendo frentes con menores porcentajes de error y con cantidades mucho menores de características seleccionadas. Esta dominancia se debe a que, al reducir el número de características de los conjuntos de datos, el algoritmo obtiene buenos porcentajes de error y se acerca más al frente de Pareto real.

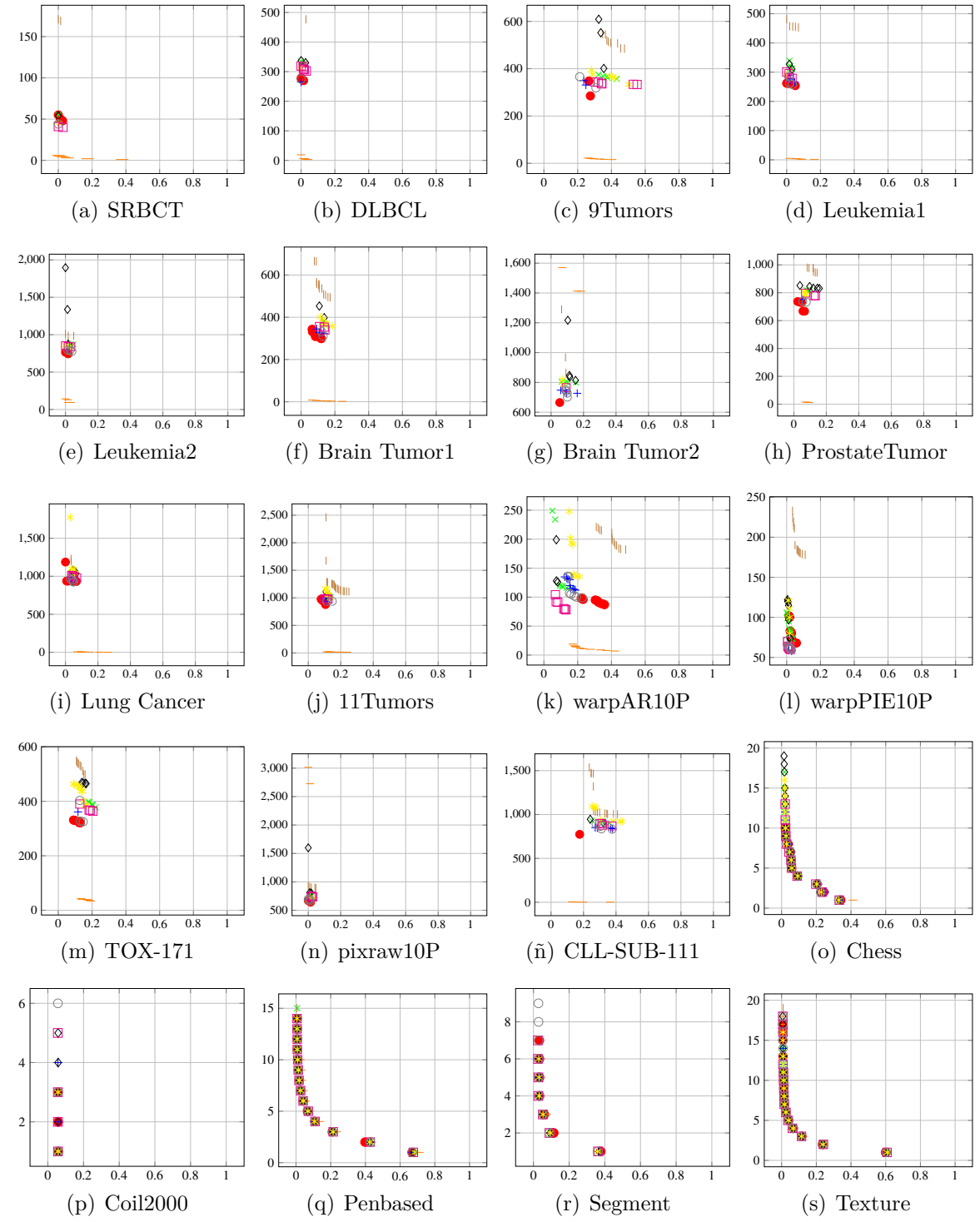



Figura 6.1: Frentes de Pareto - Entrenamiento - 10 % de características iniciales.

Por otra parte, en las figuras 6.1(c), 6.1(d), 6.1(h), 6.1(i), 6.1(j), 6.1(k) y 6.1(m) se puede apreciar que, si bien la propuesta ***RC*** (líneas horizontales de color naranja) sigue siendo una excelente opción, existen otros puntos de las demás implementaciones que también pueden considerarse parte de un conjunto de Pareto que complemente a la propuesta ***RC***. El caso más notable de este es el de *warpAR*10*P*, donde se puede apreciar que la implementación ***ICM*** (cuadrados rosados) e ***IC*** (equis verdes) presentan soluciones que pueden formar un frente de Pareto junto con los puntos de ***RC***.

Ahora bien, al observar las figuras 6.1(g), 6.1(l) y 6.1(n), se puede apreciar cómo la propuesta ***RC*** (líneas horizontales de color naranja) obtiene frentes de Pareto peores en comparación con el resto. Los casos más destacados corresponden a los de *BrainTumor*2 y *pixraw*10*P*, ya que en ambos se puede considerar a esta propuesta como la que entrega el peor frente de Pareto. Esto se puede deber al hecho de que estos conjuntos de datos requieren características que no sean relevantes por sí solas, pero que en conjunto con otras características, entreguen menores porcentajes de error y permitan explorar mejor el espacio de búsqueda para encontrar mejores soluciones, lo que se ve deteriorado por la reducción de características que realiza el método.

En términos generales, al considerar los frentes desde la figura 6.1(a) hasta la figura 6.1(ñ), se puede considerar que la implementación que entregó los mejores resultados fue la propuesta ***RC*** (líneas horizontales de color naranja), mientras que la que tuvo un peor rendimiento fue la propuesta ***VA*** (líneas verticales de color café). Esto se puede deber al hecho de que no pueda explorar de manera eficiente el espacio de búsqueda al generar individuos con distintos porcentajes de características mediante una inicialización aleatoria en vez de una inicialización guiada, lo que provoca una convergencia más lenta durante el proceso de búsqueda del algoritmo. Además, también hay que considerar que las comparaciones de los frentes de Pareto son con respecto a soluciones obtenidas con el 10 % de características iniciales, por lo que es esperable que las primeras cinco propuestas obtengan menos características seleccionadas.

En las figuras 6.1(o)- 6.1(s), se puede observar que todas las propuestas llegan a frentes de Pareto casi iguales. Esto demuestra que las ocho nuevas propuestas en este trabajo son capaces de identificar los frentes en conjuntos de datos con pocas características.

Cabe aclarar que no todos los frentes tienen la misma cantidad de soluciones en las instancias, siendo la propuesta *RC* (líneas horizontales de color naranja) la más notable, pues obtuvo la menor cantidad de soluciones. Esto se debe a que, al reducir considerablemente la cantidad de características desde un inicio, el espacio de búsqueda se acota fuertemente, por lo que suele encontrar una menor diversidad de soluciones.

Ahora bien, si se comparan los resultados obtenidos utilizando un 10 % de características con respecto a las variantes basadas en la diversidad de características, se puede apreciar que en la gran mayoría de los casos se obtuvieron mejores frentes con un 10 %, pero hay casos particulares donde alguna de las variantes obtuvo mejores resultados. Uno de estos casos se puede apreciar en la figura 6.1(a), donde la propuesta *VH* (asteriscos amarillos) dominó por completo a las implementaciones *Base* (círculos rellenos rojos), *IC* (equis verdes) e *IH* (símbolos más de color azul). Esto es algo imprevisto, ya que al iniciar considerando una mayor cantidad de características promedio entre sus individuos, se esperaría que *VH* construyera un frente de Pareto que contuviera más características en cada uno de sus puntos. Sin embargo, al ser este un caso muy particular y observar que en el resto de las primeras quince gráficas no obtiene mejores resultados, se puede considerar que la propuesta *VH* es más útil para la instancia *SRBCT* en comparación con las primeras tres implementaciones.

En general, en la figura 6.2 se pueden apreciar los mismos comportamientos que se presentaron con los conjuntos de entrenamiento, solo que esta vez la propuesta *RC* (líneas horizontales de color naranja) solamente dominó al resto de soluciones en las figuras 6.2(a), 6.2(d) y 6.2(f). Se puede apreciar el mismo efecto de frentes de Pareto en conjunto con otras soluciones en el conjunto de *warpAR*10*P* y también el mismo problema de obtención de peores frentes de Pareto en los conjuntos *BrainTumor*2, *warpPIE*10*P* y *pixraw*10*P*. Además, en los conjuntos de datos desde la figura 6.2(o) hasta 6.2(s) todas las implementaciones logran formar parte del frente de Pareto real, al igual que en el proceso de entrenamiento. Por último, nuevamente en *SRBCT* el frente de Pareto de *VH* (asteriscos amarillos) domina por completo a la propuesta *Base* (círculos rellenos de color rojo) y se encontró cerca de dominar a *IC* (equis verdes) e *IH* (símbolos más de color azul).

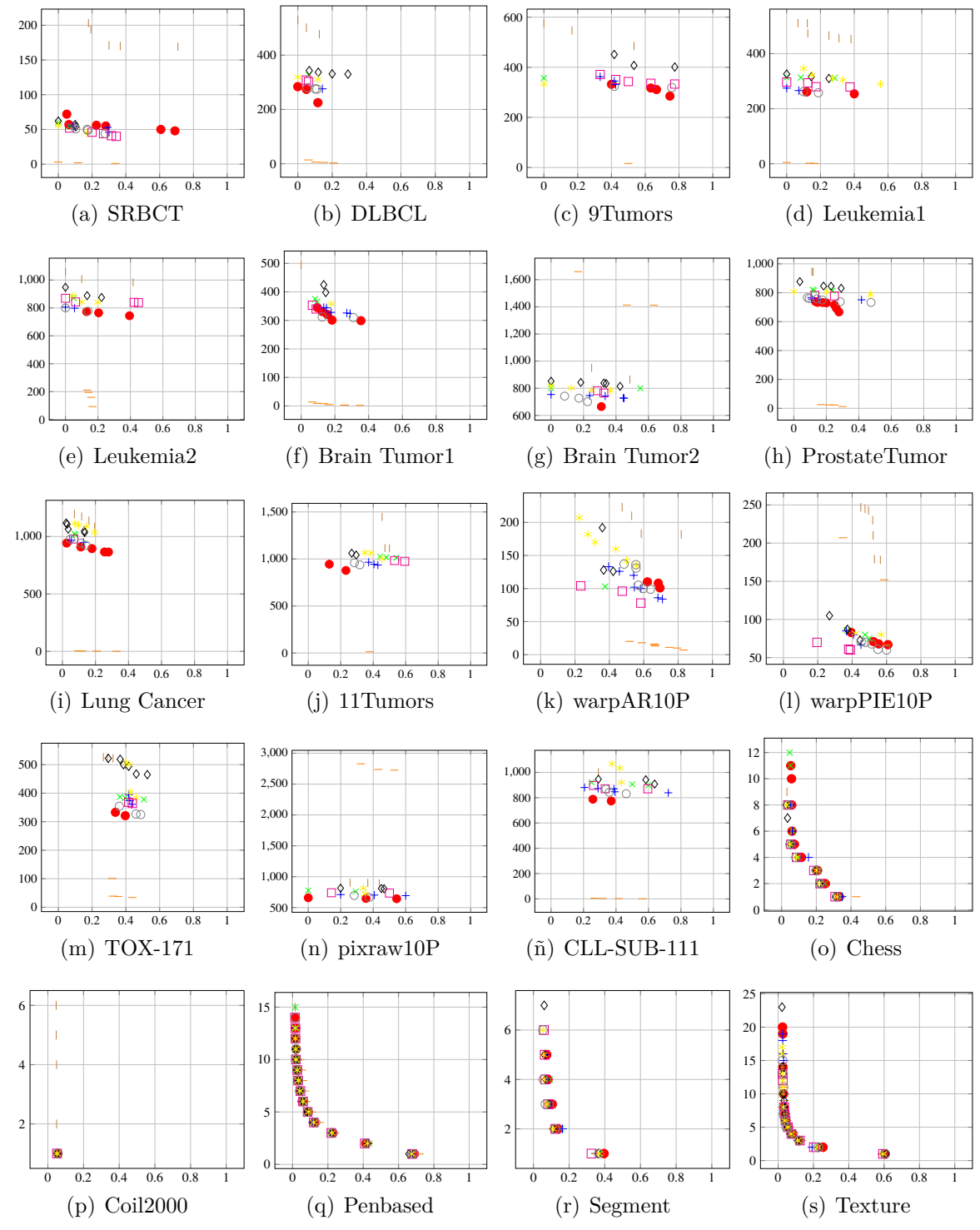



Figura 6.2: Frentes de Pareto - Prueba - 10 % de características iniciales.

Todo esto permite pensar que, al tener casos tan similares en general entre el proceso de entrenamiento y el proceso de pruebas, la propuesta ***RC*** puede ser considerada como aquella con la mayor capacidad de generalización en comparación con el resto, aunque no es una opción que se pueda utilizar en todos los tipos de problemas. Por otra parte, la propuesta ***VA*** sigue mostrando ser la que peor rendimiento tiene en general, ya que no logra generar el mejor frente de Pareto en ninguno de los casos. Ahora bien, considerando un tema de estabilidad, se podría decir que tanto la implementación ***Base*** como la ***ICM*** son las más estables. Esto se puede deber a que la primera, al ser 100 % aleatoria y con un porcentaje fijo de características iniciales, permite generar individuos más diversificados desde un inicio. Mientras tanto, la segunda, al tener un enfoque guiado por chi-cuadrado tanto en la inicialización como en parte de la mutación, permite enfocar de mejor manera la búsqueda de mejores soluciones.

### 6.1.2. 50 % de características iniciales

En las figuras 6.3 y 6.4 se observan los frentes de Pareto obtenidos en el proceso de entrenamiento y el proceso de prueba. Cada figura representa un caso de prueba. Los colores y formas utilizados son los mismos que en la sección anterior. El algoritmo ***Base*** está representado a través de círculos rellenos de color rojo, la propuesta ***IC*** está representada con equis verdes, la ***IC*** con símbolos más de color azul, la ***ICM*** con cuadrados rosados, la ***IHM*** con círculos grises, la ***VA*** con líneas verticales café, la ***VC*** con rombos negros, la ***VH*** con asteriscos amarillos y la ***RC*** con líneas horizontales.

En la figura 6.3 se puede observar que en los frentes desde la figura 6.3(a) hasta la figura 6.3(ñ) las primeras cinco propuestas se encuentran separadas del resto de soluciones. Esto tiene sentido, ya que los frentes fueron obtenidos utilizando un 50 % de características iniciales. Por esta razón, es esperable que se entreguen frentes con una mayor cantidad de características con respecto a los otros cuatro métodos con los que se comparan.

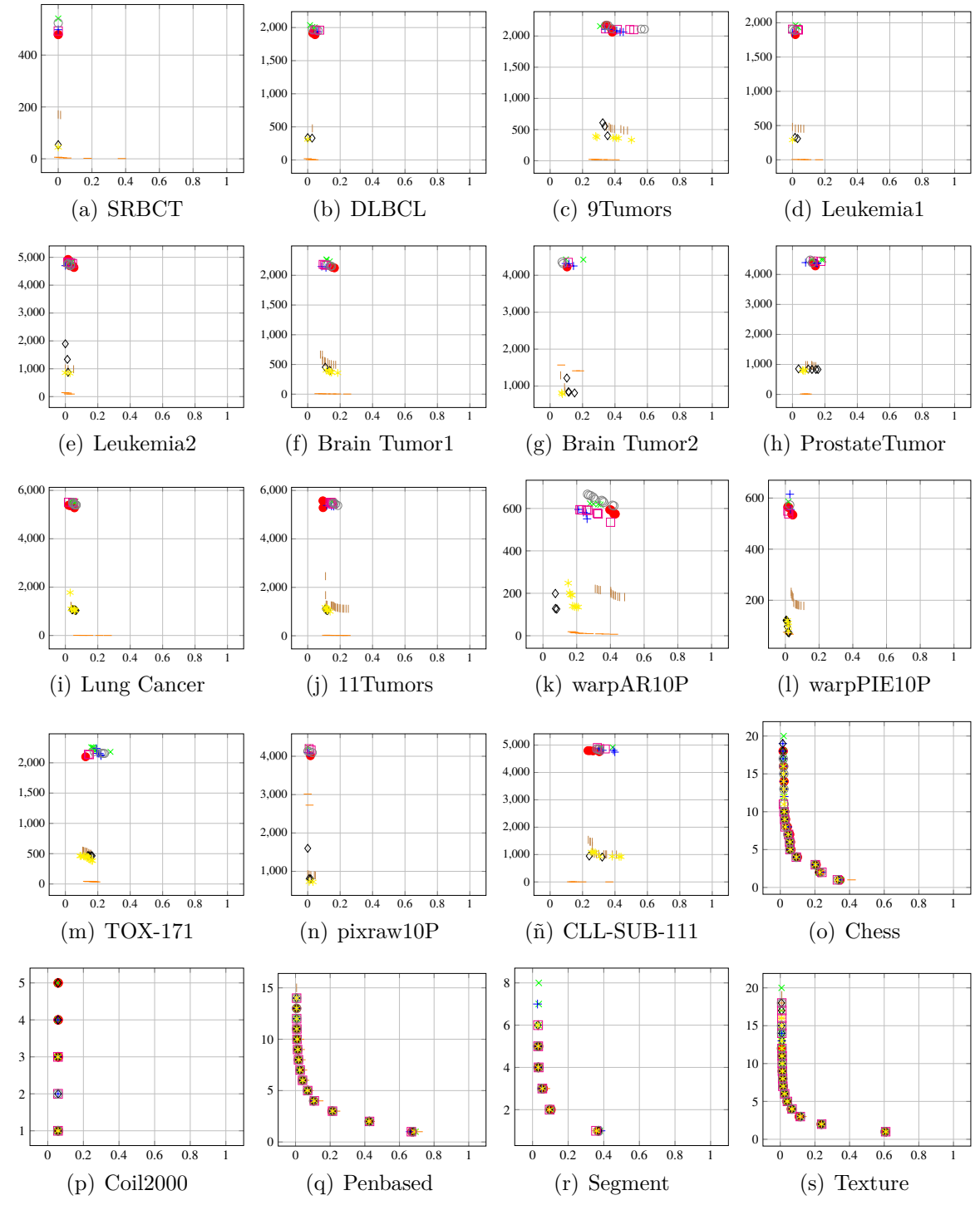



Figura 6.3: Frentes de Pareto - Entrenamiento - 50 % de características iniciales.

Esto permite observar que, por parte de los primeros cinco métodos, las propuestas *Base* (círculos rellenos de color rojo), *IH* (símbolos más de color azul) e *ICM* (cuadrados rosados) son las que presentaron mejores frentes de Pareto. Las implementaciones *Base* e *ICM* refuerzan lo mencionado en los frentes con 10 % de características iniciales, demostrando que son las más estables debido al enfoque que presentan tanto en aleatoriedad como en el enfoque guiado por chi-cuadrado. Sin embargo, se observa también que *IH* (símbolos más de color azul) obtiene buenos frentes de Pareto en algunas de las instancias, como es el caso de *BrainTumor*1 y *warpAR*10*P*. Esto se debe a que, al tener un enfoque híbrido, se mezcla la exploración de la inicialización aleatoria con la explotación de la inicialización basada en chi-cuadrado, permitiendo alcanzar mejores puntos en algunos conjuntos de datos en comparación con el resto.

Por otra parte, en las últimas figuras de la 6.3(o) a la 6.3(s) se puede observar el mismo efecto que con el 10 % de características iniciales, que es que los frentes que se construyeron con un 50 % de características iniciales también fueron capaces de llegar al frente de Pareto real. Con esto queda más que claro que los algoritmos propuestos tienen la capacidad de resolver problemas de selección de características que poseen una baja cantidad de características y una gran cantidad de muestras.

Ahora bien, si se comparan los frentes obtenidos con un 10 % de características iniciales en comparación con los obtenidos con un 50 %, es claro que los mejores frentes fueron obtenidos con el 10 %, ya que redujeron en gran medida la cantidad de características utilizadas y mantuvieron porcentajes de error similares. Lo mismo ocurre si se comparan las últimas cuatro propuestas con los frentes obtenidos con el 50 %, ya que claramente se obtuvieron frentes con una menor cantidad de características utilizando las variantes. Incluso se puede observar que la propuesta *VA* (líneas verticales café) obtiene frentes que dominan al resto de soluciones en algunos conjuntos, como es el caso de *SRBCT*, *Leukemia*1 y *pixraw*10*P*.

Con respecto a los frentes de Pareto en la figura 6.4, se puede observar cómo ocurre lo mismo que en la figura 6.3 para las comparaciones entre los resultados obtenidos con el 50 % de características iniciales y las variantes basadas en la diversidad de características.

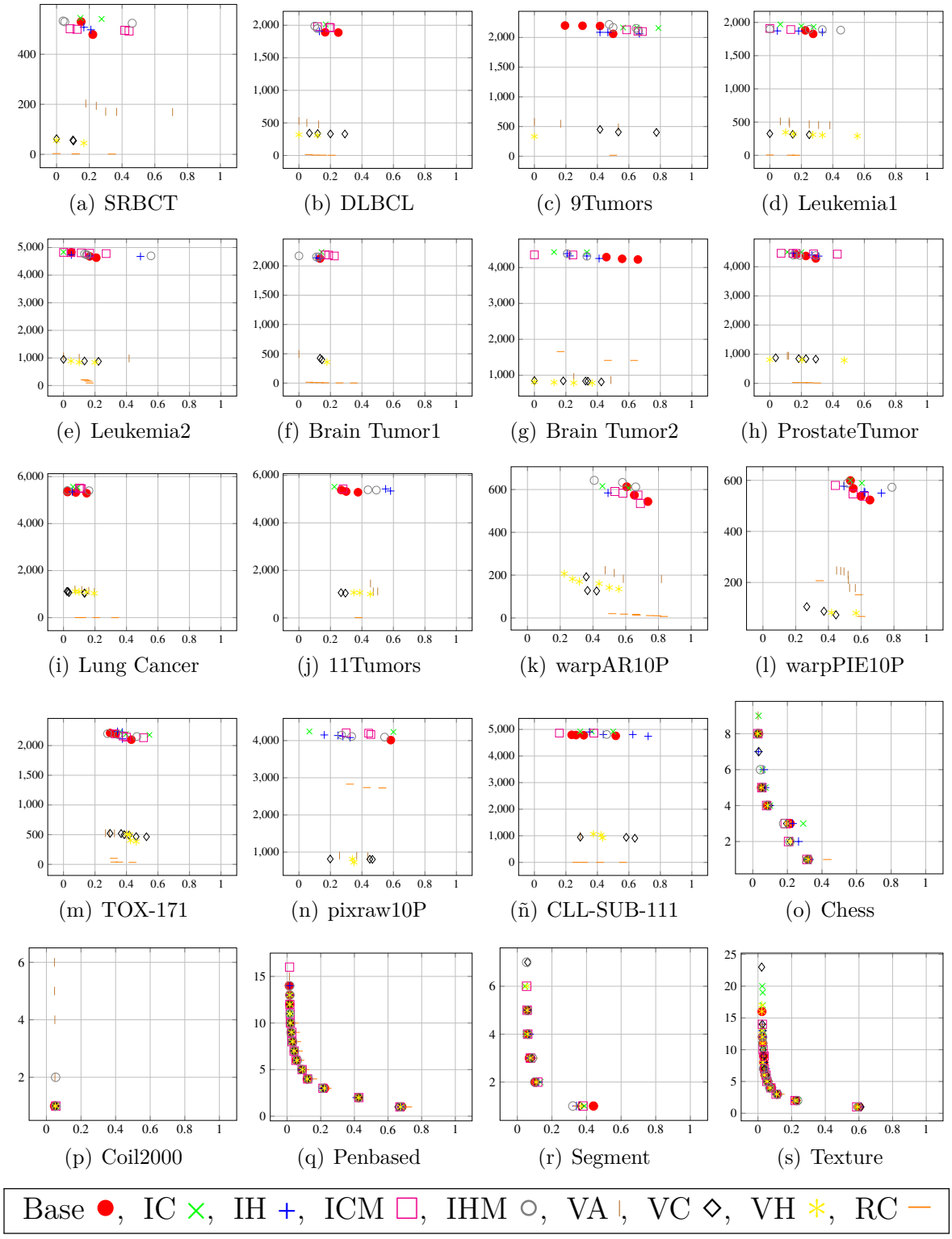



Figura 6.4: Frentes de Pareto - Prueba - 50 % de características iniciales.

Además, ocurre lo mismo para los últimos cinco conjuntos de datos, con la diferencia de que en ***Chess*** no todos los puntos se mantienen en el frente de Pareto real. Entre estos puntos, se destacan un punto de la propuesta ***IC*** (equis verdes), un punto de ***IH*** (símbolos más de color azul) y un punto de ***RC*** (líneas horizontales de color naranja), los cuales son los más notables al observar la figura y demuestran que se encuentran dominados por el resto de frentes. En el caso de las primeras dos inicializaciones, es muy probable que esto se deba a que no hayan sido capaces de explorar mejores combinaciones para obtener un menor porcentaje de error. Por otra parte, la propuesta ***RC*** obtuvo ese resultado debido a que, al reducir considerablemente la cantidad de características desde un inicio, este conjunto se vio afectado y se quedó estancado en un óptimo local durante gran parte de la ejecución.

Al observar los puntos obtenidos con un 50 % de características iniciales, se puede observar que no hubo una sola implementación que dominara al resto en ninguna de las figuras desde la 6.4(a) hasta la 6.4(ñ). Esto demuestra que todas las implementaciones tienen una buena capacidad para abordar el problema dependiendo del conjunto de datos que se esté utilizando. Sin embargo, igual se destaca la propuesta ***Base*** (círculos rellenos rojos), ya que de las quince figuras, la única en la que no obtuvo ningún punto que formara parte de un frente de Pareto no dominado en conjunto con otras soluciones fue en el conjunto ***warpAR10P***. Esto puede deberse al hecho de que este conjunto de datos requiere ser guiado por algún otro método para encontrar las características más relevantes y así obtener mejores resultados.

## 6.2. Hipervolumen

### 6.2.1. 10 % de características iniciales

En las tablas 6.2 y 6.3 se encuentran los resultados de hipervolumen obtenidos tanto para los conjuntos de entrenamiento como para los de prueba. En cada uno de los casos se presenta el promedio de 10 ejecuciones independientes de cada propuesta en cada caso de prueba. Los promedios más altos por cada instancia se presentan subrayados.

Con respecto a la tabla 6.2, se puede observar que en doce de los veinte conjuntos de datos la propuesta de reducción de características obtuvo un mejor promedio en comparación con el resto. Esto complementa los frentes de Pareto presentados anteriormente, donde aquí se puede volver a considerar que la implementación ***RC*** es la que en general obtiene mejores resultados. Cabe resaltar que también se obtienen mejores resultados de hipervolumen debido a que se obtiene una menor cantidad de características seleccionadas en comparación con las propuestas que utilizan un 10 % de características iniciales. También cabe mencionar que, de estos doce conjuntos de datos, en SRBCT se obtuvo un promedio de **1.00**. Esto se debe a que, al realizarse la aproximación a dos decimales y con el promedio muy cercano a **1.00**, se terminó realizando la aproximación y quedó de esa manera.

| Propuesta | Base | IC | IH | ICM | IHM | VA | VC | VH | RC |
|---|---|---|---|---|---|---|---|---|---|
| Instancia | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE |
| SRBCT | 0.97 ± 0.01 | 0.97 ± 0.00 | 0.98 ± 0.00 | 0.98 ± 0.00 | 0.98 ± 0.00 | 0.91 ± 0.01 | 0.97 ± 0.00 | 0.98 ± 0.00 | 1.00 ± 0.01 |
| DLBCL | 0.94 ± 0.07 | 0.92 ± 0.02 | 0.94 ± 0.01 | 0.93 ± 0.01 | 0.94 ± 0.01 | 0.87 ± 0.01 | 0.92 ± 0.02 | 0.94 ± 0.01 | 0.98 ± 0.01 |
| 9Tumors | 0.60 ± 0.06 | 0.56 ± 0.05 | 0.63 ± 0.07 | 0.57 ± 0.08 | 0.60 ± 0.07 | 0.52 ± 0.05 | 0.53 ± 0.05 | 0.59 ± 0.06 | 0.67 ± 0.05 |
| Leukemia1 | 0.93 ± 0.02 | 0.91 ± 0.01 | 0.92 ± 0.02 | 0.93 ± 0.01 | 0.93 ± 0.01 | 0.89 ± 0.01 | 0.91 ± 0.01 | 0.92 ± 0.02 | 0.95 ± 0.02 |
| Leukemia2 | 0.92 ± 0.01 | 0.89 ± 0.02 | 0.91 ± 0.01 | 0.90 ± 0.02 | 0.90 ± 0.01 | 0.89 ± 0.02 | 0.90 ± 0.01 | 0.90 ± 0.02 | 0.97 ± 0.01 |
| Brain Tumor1 | 0.85 ± 0.02 | 0.79 ± 0.02 | 0.82 ± 0.03 | 0.81 ± 0.02 | 0.81 ± 0.01 | 0.80 ± 0.04 | 0.79 ± 0.03 | 0.80 ± 0.02 | 0.91 ± 0.02 |
| Brain Tumor2 | 0.85 ± 0.03 | 0.80 ± 0.03 | 0.81 ± 0.04 | 0.80 ± 0.03 | 0.82 ± 0.02 | 0.79 ± 0.05 | 0.79 ± 0.02 | 0.81 ± 0.03 | 0.73 ± 0.03 |
| ProstateTumor | 0.88 ± 0.02 | 0.82 ± 0.02 | 0.85 ± 0.02 | 0.81 ± 0.02 | 0.86 ± 0.01 | 0.79 ± 0.02 | 0.82 ± 0.04 | 0.85 ± 0.01 | 0.89 ± 0.02 |
| Lung Cancer | 0.92 ± 0.01 | 0.87 ± 0.01 | 0.88 ± 0.01 | 0.87 ± 0.01 | 0.88 ± 0.02 | 0.87 ± 0.01 | 0.86 ± 0.01 | 0.87 ± 0.01 | 0.89 ± 0.02 |
| 11Tumors | 0.84 ± 0.01 | 0.79 ± 0.01 | 0.82 ± 0.01 | 0.82 ± 0.01 | 0.81 ± 0.01 | 0.79 ± 0.01 | 0.81 ± 0.01 | 0.81 ± 0.01 | 0.85 ± 0.02 |
| warpAR10P | 0.66 ± 0.05 | 0.83 ± 0.03 | 0.78 ± 0.03 | 0.86 ± 0.02 | 0.78 ± 0.03 | 0.58 ± 0.04 | 0.83 ± 0.03 | 0.77 ± 0.01 | 0.78 ± 0.03 |
| warpPIE10P | 0.94 ± 0.01 | 0.94 ± 0.01 | 0.95 ± 0.01 | 0.96 ± 0.01 | 0.95 ± 0.01 | 0.87 ± 0.01 | 0.94 ± 0.01 | 0.94 ± 0.01 | 0.90 ± 0.04 |
| TOX-171 | 0.82 ± 0.03 | 0.73 ± 0.03 | 0.77 ± 0.03 | 0.74 ± 0.05 | 0.78 ± 0.03 | 0.76 ± 0.04 | 0.73 ± 0.03 | 0.78 ± 0.04 | 0.82 ± 0.02 |
| pixraw10P | 0.92 ± 0.01 | 0.90 ± 0.01 | 0.91 ± 0.01 | 0.89 ± 0.01 | 0.92 ± 0.01 | 0.89 ± 0.01 | 0.91 ± 0.01 | 0.91 ± 0.01 | 0.70 ± 0.01 |
| CLL-SUB-111 | 0.71 ± 0.03 | 0.58 ± 0.06 | 0.61 ± 0.03 | 0.60 ± 0.04 | 0.62 ± 0.03 | 0.64 ± 0.03 | 0.61 ± 0.04 | 0.61 ± 0.04 | 0.78 ± 0.07 |
| Chess | 0.92 ± 0.01 | 0.92 ± 0.01 | 0.92 ± 0.01 | 0.91 ± 0.01 | 0.91 ± 0.01 | 0.91 ± 0.01 | 0.93 ± 0.00 | 0.93 ± 0.00 | 0.57 ± 0.00 |
| Coil2000 | 0.93 ± 0.00 | 0.93 ± 0.00 | 0.93 ± 0.00 | 0.93 ± 0.00 | 0.93 ± 0.00 | 0.92 ± 0.01 | 0.93 ± 0.00 | 0.93 ± 0.00 | 0.93 ± 0.00 |
| Penbased | 0.83 ± 0.00 | 0.83 ± 0.00 | 0.83 ± 0.00 | 0.83 ± 0.00 | 0.83 ± 0.00 | 0.83 ± 0.00 | 0.83 ± 0.00 | 0.83 ± 0.00 | 0.82 ± 0.00 |
| Segment | 0.90 ± 0.00 | 0.90 ± 0.00 | 0.89 ± 0.00 | 0.89 ± 0.00 | 0.89 ± 0.00 | 0.89 ± 0.00 | 0.89 ± 0.00 | 0.89 ± 0.00 | 0.84 ± 0.02 |
| Texture | 0.94 ± 0.00 | 0.94 ± 0.00 | 0.94 ± 0.00 | 0.94 ± 0.00 | 0.94 ± 0.00 | 0.93 ± 0.02 | 0.94 ± 0.00 | 0.94 ± 0.00 | 0.94 ± 0.00 |

Tabla 6.2: Hipervolumen en proceso de entrenamiento — 10 % de características iniciales.

Otra cosa que también se puede apreciar con respecto a *RC* tiene que ver con las instancias *BrainTumor*2, *warpPIE*10*P* y *Chess*, ya que estos son los mismos conjuntos que se mencionó en la sección de frentes de Pareto como los casos en los que esta propuesta no es capaz de explorar bien el espacio de búsqueda y por eso termina obteniendo peores resultados. Esto mismo se comprueba con los resultados de hipervolumen, ya que en los tres casos obtuvo el peor promedio de hipervolumen. De estas tres instancias, la más impactante es la de *Chess*, ya que obtuvo un promedio de hipervolumen mucho peor que el del resto. Esto se puede alinear con lo comentado para la figura 6.4, que tiene que ver con que la reducción provocó que el algoritmo se quedara estancado en óptimos locales. Otra prueba para esto mismo es la desviación estándar, ya que justamente en este caso la desviación estándar es 0.00.

Ahora bien, con respecto a las instancias que van desde *Chess* hasta *Texture*, se puede apreciar que se obtuvieron promedios muy similares, lo cual concuerda con el hecho de que todas las propuestas llegaban al frente de Pareto real. Además, la gran mayoría de las desviaciones estándar fueron 0.00.

Con respecto a los resultados obtenidos en los conjuntos de datos que van desde *SRBCT* hasta $CLL-SUB-111$, se puede apreciar que ninguna de las variantes basadas en la diversidad de características fue capaz de obtener el mejor promedio. Esto concuerda con el hecho de que es mejor mantener un porcentaje de características fijo en la inicialización en vez de variar los individuos. Sin embargo, aunque este sea el caso, aun así las propuestas *VC* y *VH* obtuvieron muy buenos promedios en comparación con el resto. La única que quedó por detrás en su gran mayoría fue la implementación *VA*, lo cual concuerda con lo analizado en los frentes de Pareto, donde en la mayoría de los casos esta propuesta fue la que obtuvo los peores puntos.

También es importante resaltar que, entre los resultados obtenidos utilizando un 10 % de características iniciales, los que destacaron son la propuesta *Base*, la *ICM* y la *IHM*. Con esto se pueden observar dos cosas. La primera es que el algoritmo de NSGA-II base sigue obteniendo buenos resultados a pesar de ser una implementación de hace más de dos décadas, obteniendo mejores promedios en ocho de los veinte conjuntos de datos. La segunda es que la mutación basada en chi-cuadrado generó un aporte a los

valores de hipervolumen obtenidos, ya que las implementaciones de $ICM$ e $IHM$ entregaron mejores resultados en diversas propuestas en comparación con $IC$ e $IH$ e incluso lograron obtener el mejor promedio en dos y en un conjunto de datos, respectivamente.

A diferencia de la tabla 6.2, se puede ver que en la tabla 6.3 ocurre un fenómeno distinto. Esta vez solo se obtuvo un mejor promedio utilizando $RC$ en tres de los veinte conjuntos de datos. Esto se contradice con lo visto en los frentes de Pareto, ya que con esto no se puede considerar que esta propuesta es buena también para generalizar, aunque igual mantiene buenos promedios en general. Eso sí, se vuelve a ver el mismo fenómeno en los conjuntos de $BrainTumor2$, $pixraw10P$ y $Chess$, donde es esta implementación la que obtiene los peores promedios. Uno de los aspectos en los que sí coincide con lo observado en los frentes de Pareto es el hecho de que dominaba a una menor cantidad de puntos en el proceso de prueba. Esto coincide con que $RC$ obtuviera una menor cantidad de mejores promedios y estos se encontraran repartidos entre el resto de propuestas, ya que se muestra que existen más opciones para escoger en base a si alguien busca enfocarse en un menor error o en una menor cantidad de características.

Ahora bien, con respecto al resto de resultados, se puede observar que el que obtuvo los mejores promedios corresponde a $VC$, con ocho de los veinte conjuntos de datos, seguido de $Base$, que tiene siete. Incluso se puede ver que $VH$ obtuvo mejores promedios en cinco instancias y que $VA$ obtuvo el mejor promedio en $CLL\_SUB\_111$. Con esto se puede considerar que, aunque las variantes basadas en la diversidad de características no obtengan tan buenos resultados en el proceso de entrenamiento, tienen una mayor capacidad de generalización. Esto se puede explicar por la variedad de características en los individuos iniciales obtenidos con el uso de chi-cuadrado, ya que tanto $VC$ como $VH$ comienzan con una población bien diversa y que está guiada por las soluciones más relevantes.

Con base en los valores de desviación estándar obtenidos, se puede observar que esta vez los valores en general son mucho mayores en comparación con los obtenidos en la tabla 6.2, lo cual coincide con lo observado en la figura 6.2, ya que se obtienen frentes más diversificados en comparación con la figura 6.1.

| Propuesta | Base | IC | IH | ICM | IHM | VA | VC | VH | RC |
|---|---|---|---|---|---|---|---|---|---|
| Instancia | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE |
| SRBCT | 0.72 ± 0.17 | 0.77 ± 0.20 | 0.72 ± 0.11 | 0.76 ± 0.11 | 0.69 ± 0.15 | 0.45 ± 0.03 | 0.63 ± 0.03 | 0.55 ± 0.03 | 0.86 ± 0.12 |
| DLBCL | 0.85 ± 0.07 | 0.84 ± 0.05 | 0.76 ± 0.05 | 0.77 ± 0.11 | 0.80 ± 0.07 | 0.76 ± 0.09 | 0.79 ± 0.06 | 0.77 ± 0.11 | 0.82 ± 0.13 |
| 9Tumors | 0.30 ± 0.14 | 0.51 ± 0.23 | 0.45 ± 0.14 | 0.33 ± 0.16 | 0.33 ± 0.12 | 0.36 ± 0.21 | 0.32 ± 0.12 | 0.35 ± 0.24 | 0.36 ± 0.10 |
| Leukemia1 | 0.64 ± 0.12 | 0.79 ± 0.12 | 0.76 ± 0.11 | 0.76 ± 0.11 | 0.74 ± 0.10 | 0.67 ± 0.09 | 0.81 ± 0.09 | 0.68 ± 0.13 | 0.75 ± 0.16 |
| Leukemia2 | 0.65 ± 0.10 | 0.71 ± 0.15 | 0.83 ± 0.10 | 0.70 ± 0.17 | 0.82 ± 0.07 | 0.70 ± 0.11 | 0.68 ± 0.15 | 0.71 ± 0.20 | 0.76 ± 0.11 |
| Brain Tumor1 | 0.74 ± 0.09 | 0.72 ± 0.10 | 0.73 ± 0.06 | 0.75 ± 0.10 | 0.69 ± 0.11 | 0.72 ± 0.12 | 0.65 ± 0.12 | 0.69 ± 0.08 | 0.79 ± 0.11 |
| Brain Tumor2 | 0.64 ± 0.00 | 0.56 ± 0.17 | 0.60 ± 0.18 | 0.46 ± 0.13 | 0.57 ± 0.18 | 0.48 ± 0.11 | 0.57 ± 0.19 | 0.65 ± 0.15 | 0.48 ± 0.14 |
| ProstateTumor | 0.70 ± 0.09 | 0.70 ± 0.09 | 0.71 ± 0.14 | 0.69 ± 0.10 | 0.69 ± 0.13 | 0.72 ± 0.08 | 0.63 ± 0.15 | 0.73 ± 0.12 | 0.70 ± 0.09 |
| Lung Cancer | 0.77 ± 0.08 | 0.80 ± 0.05 | 0.76 ± 0.06 | 0.77 ± 0.06 | 0.78 ± 0.05 | 0.73 ± 0.07 | 0.82 ± 0.06 | 0.75 ± 0.06 | 0.80 ± 0.07 |
| 11Tumors | 0.70 ± 0.05 | 0.45 ± 0.03 | 0.55 ± 0.03 | 0.42 ± 0.02 | 0.64 ± 0.02 | 0.45 ± 0.03 | 0.63 ± 0.03 | 0.64 ± 0.02 | 0.56 ± 0.06 |
| warpAR10P | 0.29 ± 0.06 | 0.48 ± 0.09 | 0.40 ± 0.10 | 0.44 ± 0.13 | 0.36 ± 0.08 | 0.24 ± 0.13 | 0.40 ± 0.15 | 0.40 ± 0.15 | 0.33 ± 0.09 |
| warpPIE10P | 0.35 ± 0.13 | 0.49 ± 0.09 | 0.47 ± 0.09 | 0.55 ± 0.10 | 0.38 ± 0.10 | 0.31 ± 0.11 | 0.57 ± 0.08 | 0.41 ± 0.08 | 0.38 ± 0.12 |
| TOX-171 | 0.51 ± 0.06 | 0.47 ± 0.08 | 0.47 ± 0.06 | 0.43 ± 0.08 | 0.48 ± 0.07 | 0.46 ± 0.14 | 0.49 ± 0.09 | 0.48 ± 0.06 | 0.57 ± 0.07 |
| pixraw10P | 0.57 ± 0.15 | 0.52 ± 0.18 | 0.53 ± 0.10 | 0.52 ± 0.14 | 0.49 ± 0.11 | 0.51 ± 0.11 | 0.49 ± 0.13 | 0.46 ± 0.10 | 0.36 ± 0.07 |
| CLL-SUB-111 | 0.53 ± 0.13 | 0.48 ± 0.13 | 0.51 ± 0.14 | 0.50 ± 0.12 | 0.49 ± 0.08 | 0.58 ± 0.08 | 0.48 ± 0.12 | 0.43 ± 0.11 | 0.55 ± 0.12 |
| Chess | 0.89 ± 0.00 | 0.89 ± 0.01 | 0.89 ± 0.01 | 0.90 ± 0.02 | 0.90 ± 0.01 | 0.89 ± 0.01 | 0.91 ± 0.01 | 0.91 ± 0.01 | 0.55 ± 0.00 |
| Coil2000 | 0.93 ± 0.01 | 0.93 ± 0.01 | 0.93 ± 0.01 | 0.93 ± 0.01 | 0.93 ± 0.01 | 0.92 ± 0.01 | 0.93 ± 0.01 | 0.93 ± 0.01 | 0.92 ± 0.02 |
| Penbased | 0.82 ± 0.00 | 0.82 ± 0.00 | 0.82 ± 0.00 | 0.82 ± 0.00 | 0.82 ± 0.00 | 0.82 ± 0.01 | 0.82 ± 0.00 | 0.82 ± 0.00 | 0.79 ± 0.00 |
| Segment | 0.86 ± 0.00 | 0.84 ± 0.00 | 0.85 ± 0.01 | 0.86 ± 0.01 | 0.86 ± 0.01 | 0.86 ± 0.01 | 0.86 ± 0.01 | 0.85 ± 0.01 | 0.81 ± 0.03 |
| Texture | 0.92 ± 0.01 | 0.92 ± 0.01 | 0.92 ± 0.00 | 0.92 ± 0.00 | 0.92 ± 0.00 | 0.91 ± 0.02 | 0.92 ± 0.00 | 0.92 ± 0.01 | 0.91 ± 0.00 |

Tabla 6.3: Hipervolumen en proceso de prueba — 10 % de características iniciales.

### 6.2.2. 50 % de características iniciales

En las tablas 6.4 y 6.5 se encuentran los resultados de hipervolumen obtenidos tanto para los conjuntos de entrenamiento como para los de prueba. En cada uno de los casos se presenta el promedio de 10 ejecuciones independientes de cada propuesta en cada caso de prueba. Los promedios más altos por cada instancia se encuentran subrayados.

Con respecto a la tabla 6.4, se puede observar nuevamente que se obtuvieron mejores promedios utilizando ***RC***. Esto es esperable ya que tanto las variantes basadas en diversidad de características como la propuesta de reducción de características obtuvieron una menor cantidad de características seleccionadas en sus soluciones, lo que provoca que obtengan un mayor promedio en comparación con los resultados obtenidos con un 50 % de características iniciales.

| Propuesta | Base | IC | IH | ICM | IHM | VA | VC | VH | RC |
|---|---|---|---|---|---|---|---|---|---|
| Instancia | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE |
| SRBCT | 0.77 ± 0.01 | 0.76 ± 0.01 | 0.77 ± 0.01 | 0.78 ± 0.01 | 0.77 ± 0.01 | 0.91 ± 0.01 | 0.97 ± 0.00 | 0.98 ± 0.00 | 1.00 ± 0.01 |
| DLBCL | 0.62 ± 0.01 | 0.61 ± 0.01 | 0.59 ± 0.01 | 0.60 ± 0.01 | 0.61 ± 0.01 | 0.87 ± 0.01 | 0.92 ± 0.02 | 0.94 ± 0.01 | 0.98 ± 0.01 |
| 9Tumors | 0.38 ± 0.03 | 0.32 ± 0.05 | 0.36 ± 0.03 | 0.36 ± 0.03 | 0.35 ± 0.04 | 0.52 ± 0.05 | 0.53 ± 0.05 | 0.59 ± 0.06 | 0.67 ± 0.05 |
| Leukemia1 | 0.64 ± 0.01 | 0.61 ± 0.01 | 0.64 ± 0.01 | 0.62 ± 0.02 | 0.62 ± 0.02 | 0.89 ± 0.01 | 0.91 ± 0.01 | 0.92 ± 0.02 | 0.95 ± 0.02 |
| Leukemia2 | 0.55 ± 0.01 | 0.54 ± 0.01 | 0.56 ± 0.01 | 0.55 ± 0.01 | 0.56 ± 0.00 | 0.89 ± 0.02 | 0.90 ± 0.01 | 0.90 ± 0.02 | 0.97 ± 0.01 |
| Brain Tumor1 | 0.54 ± 0.00 | 0.53 ± 0.02 | 0.54 ± 0.02 | 0.55 ± 0.02 | 0.54 ± 0.01 | 0.80 ± 0.04 | 0.79 ± 0.03 | 0.80 ± 0.02 | 0.91 ± 0.02 |
| Brain Tumor2 | 0.52 ± 0.01 | 0.45 ± 0.03 | 0.49 ± 0.03 | 0.46 ± 0.03 | 0.49 ± 0.03 | 0.79 ± 0.05 | 0.79 ± 0.02 | 0.81 ± 0.03 | 0.73 ± 0.03 |
| ProstateTumor | 0.50 ± 0.01 | 0.46 ± 0.02 | 0.49 ± 0.02 | 0.48 ± 0.01 | 0.48 ± 0.03 | 0.79 ± 0.02 | 0.82 ± 0.04 | 0.85 ± 0.01 | 0.89 ± 0.02 |
| Lung Cancer | 0.55 ± 0.01 | 0.53 ± 0.01 | 0.54 ± 0.01 | 0.54 ± 0.01 | 0.53 ± 0.01 | 0.87 ± 0.01 | 0.86 ± 0.01 | 0.87 ± 0.01 | 0.89 ± 0.02 |
| 11Tumors | 0.51 ± 0.01 | 0.47 ± 0.00 | 0.49 ± 0.01 | 0.48 ± 0.01 | 0.46 ± 0.01 | 0.79 ± 0.01 | 0.81 ± 0.01 | 0.81 ± 0.01 | 0.85 ± 0.02 |
| warpAR10P | 0.43 ± 0.01 | 0.51 ± 0.02 | 0.57 ± 0.01 | 0.52 ± 0.03 | 0.48 ± 0.03 | 0.58 ± 0.04 | 0.83 ± 0.03 | 0.77 ± 0.01 | 0.78 ± 0.03 |
| warpPIE10P | 0.73 ± 0.02 | 0.73 ± 0.01 | 0.73 ± 0.01 | 0.74 ± 0.01 | 0.72 ± 0.01 | 0.87 ± 0.01 | 0.94 ± 0.01 | 0.94 ± 0.01 | 0.90 ± 0.04 |
| TOX171 | 0.51 ± 0.03 | 0.48 ± 0.02 | 0.49 ± 0.01 | 0.49 ± 0.03 | 0.48 ± 0.02 | 0.76 ± 0.04 | 0.73 ± 0.03 | 0.78 ± 0.04 | 0.82 ± 0.02 |
| pixraw10P | 0.58 ± 0.00 | 0.57 ± 0.01 | 0.58 ± 0.01 | 0.56 ± 0.01 | 0.57 ± 0.01 | 0.89 ± 0.01 | 0.91 ± 0.01 | 0.91 ± 0.01 | 0.70 ± 0.01 |
| CLL-SUB-111 | 0.40 ± 0.03 | 0.35 ± 0.03 | 0.36 ± 0.02 | 0.36 ± 0.03 | 0.40 ± 0.01 | 0.64 ± 0.03 | 0.61 ± 0.04 | 0.61 ± 0.04 | 0.78 ± 0.07 |
| Chess | 0.91 ± 0.01 | 0.93 ± 0.00 | 0.93 ± 0.00 | 0.92 ± 0.00 | 0.92 ± 0.00 | 0.91 ± 0.01 | 0.93 ± 0.00 | 0.93 ± 0.00 | 0.57 ± 0.00 |
| Coil2000 | 0.93 ± 0.00 | 0.93 ± 0.00 | 0.93 ± 0.00 | 0.93 ± 0.00 | 0.93 ± 0.00 | 0.92 ± 0.01 | 0.93 ± 0.00 | 0.93 ± 0.00 | 0.93 ± 0.00 |
| Penbased | 0.83 ± 0.00 | 0.83 ± 0.00 | 0.83 ± 0.00 | 0.83 ± 0.00 | 0.83 ± 0.00 | 0.83 ± 0.00 | 0.83 ± 0.00 | 0.83 ± 0.00 | 0.82 ± 0.00 |
| Segment | 0.89 ± 0.00 | 0.89 ± 0.00 | 0.90 ± 0.00 | 0.89 ± 0.00 | 0.89 ± 0.00 | 0.89 ± 0.00 | 0.89 ± 0.00 | 0.89 ± 0.00 | 0.84 ± 0.02 |
| Texture | 0.94 ± 0.00 | 0.94 ± 0.00 | 0.94 ± 0.00 | 0.94 ± 0.00 | 0.94 ± 0.00 | 0.93 ± 0.02 | 0.94 ± 0.00 | 0.94 ± 0.00 | 0.94 ± 0.00 |

Tabla 6.4: Hipervolumen en proceso de entrenamiento — 50 % de características iniciales

Ahora bien, con respecto a los promedios obtenidos por las primeras cinco propuestas, se puede observar que la implementación ***Base*** obtuvo mejores resultados en general en comparación con el resto. Tanto estos resultados como los frentes de la figura 6.3 permiten mostrar que, a la hora de utilizar un 50 % de características iniciales, los resultados obtenidos con NSGA-II base son mejores que el resto de las implementaciones. Sin embargo, existe una instancia donde ***Base*** no se desempeña tan bien, la cual corresponde a ***warpAR10P***. Esto se puede asociar con lo observado en los frentes de Pareto referente a que este conjunto de datos requiere que el algoritmo esté guiado y comience con características más relevantes desde un inicio, sin dejar de perder la aleatoriedad. Por esta misma razón es que la propuesta ***IH*** es la que obtiene el mejor promedio, ya que balancea las características más relevantes con la aleatoriedad.

Al fijarse en los valores de desviación estándar obtenidos, se puede observar lo mismo que en la tabla 6.2, donde los valores obtenidos son bastante bajos. Esto se refuerza al ver los frentes de Pareto en la figura 6.3, donde se puede apreciar que no hubo tanta diversificación de puntos.

Con respecto a la tabla 6.3, se puede observar nuevamente que las primeras cinco propuestas no obtuvieron el mejor promedio en ninguna de las primeras quince instancias. También se puede apreciar nuevamente el mismo tema con respecto a la generalización, donde las variantes basadas en diversidad de características obtuvieron los mejores promedios en varias instancias. Con esto se confirma que ***RC*** no es tan bueno para generalizar comparado con los buenos resultados presentados en los procesos de entrenamiento. También se puede confirmar que ***VA***, ***VC*** y ***VH*** tienen una buena capacidad de generalización pese a no tener tan buen desempeño en el entrenamiento en comparación con ***RC***.

Con respecto a las primeras cinco propuestas, se obtuvo una mayor cantidad de mejores promedios en ***ICM***, con 10 de 20 instancias con mejores promedios en comparación con las otras cuatro propuestas. Esto permite observar que enfocar una gran parte del algoritmo en las características más relevantes permite que el algoritmo tenga una mayor capacidad de generalización. La otra que también generalizó bien fue la implementación ***Base***, lo cual permite observar que el uso de aleatoriedad durante todo el proceso también permite generalizar mejor.

Con base en los valores de desviación estándar obtenidos, se vuelve a observar que los valores en general son mucho mayores en comparación con los obtenidos en la tabla 6.4, lo cual coincide con lo observado en la figura 6.4, ya que se obtienen frentes más diversificados en comparación con los de la figura 6.3.

## 6.3. Comparaciones con el estado del arte

Se presentan los promedios de hipervolumen y sus correspondientes desviaciones estándar para los conjuntos de *warpAR*10*P* y *wwarpPIE*10*P*. Se utilizan estos conjuntos de

| Propuesta | Base | IC | IH | ICM | IHM | VA | VC | VH | RC |
|---|---|---|---|---|---|---|---|---|---|
| Instancia | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE |
| SRBCT | 0.51 ± 0.11 | 0.56 ± 0.09 | 0.54 ± 0.07 | 0.41 ± 0.00 | 0.30 ± 0.02 | 0.45 ± 0.03 | 0.63 ± 0.03 | 0.55 ± 0.03 | 0.86 ± 0.12 |
| DLBCL | 0.50 ± 0.03 | 0.48 ± 0.02 | 0.54 ± 0.02 | 0.49 ± 0.06 | 0.54 ± 0.03 | 0.76 ± 0.09 | 0.79 ± 0.06 | 0.77 ± 0.11 | 0.82 ± 0.13 |
| 9Tumors | 0.28 ± 0.11 | 0.20 ± 0.06 | 0.24 ± 0.08 | 0.16 ± 0.07 | 0.22 ± 0.08 | 0.36 ± 0.21 | 0.32 ± 0.12 | 0.35 ± 0.24 | 0.36 ± 0.10 |
| Leukemia1 | 0.43 ± 0.04 | 0.45 ± 0.08 | 0.49 ± 0.11 | 0.52 ± 0.07 | 0.49 ± 0.09 | 0.67 ± 0.09 | 0.81 ± 0.09 | 0.68 ± 0.13 | 0.75 ± 0.16 |
| Leukemia2 | 0.40 ± 0.09 | 0.44 ± 0.09 | 0.42 ± 0.09 | 0.48 ± 0.06 | 0.43 ± 0.08 | 0.70 ± 0.11 | 0.68 ± 0.15 | 0.71 ± 0.20 | 0.76 ± 0.11 |
| Brain Tumor1 | 0.55 ± 0.00 | 0.47 ± 0.04 | 0.52 ± 0.05 | 0.46 ± 0.06 | 0.51 ± 0.09 | 0.72 ± 0.12 | 0.65 ± 0.12 | 0.69 ± 0.08 | 0.79 ± 0.11 |
| Brain Tumor2 | 0.25 ± 0.04 | 0.37 ± 0.07 | 0.34 ± 0.08 | 0.36 ± 0.12 | 0.31 ± 0.09 | 0.48 ± 0.11 | 0.57 ± 0.19 | 0.65 ± 0.15 | 0.48 ± 0.14 |
| ProstateTumor | 0.44 ± 0.03 | 0.41 ± 0.06 | 0.41 ± 0.04 | 0.43 ± 0.07 | 0.43 ± 0.05 | 0.72 ± 0.08 | 0.63 ± 0.15 | 0.73 ± 0.12 | 0.70 ± 0.09 |
| Lung Cancer | 0.50 ± 0.05 | 0.49 ± 0.03 | 0.48 ± 0.04 | 0.47 ± 0.03 | 0.50 ± 0.05 | 0.73 ± 0.07 | 0.82 ± 0.06 | 0.75 ± 0.06 | 0.80 ± 0.07 |
| 11Tumors | 0.39 ± 0.02 | 0.43 ± 0.00 | 0.24 ± 0.01 | 0.41 ± 0.00 | 0.30 ± 0.02 | 0.45 ± 0.03 | 0.63 ± 0.03 | 0.64 ± 0.02 | 0.56 ± 0.06 |
| warpAR10P | 0.20 ± 0.07 | 0.25 ± 0.10 | 0.26 ± 0.09 | 0.27 ± 0.07 | 0.25 ± 0.09 | 0.24 ± 0.13 | 0.40 ± 0.15 | 0.40 ± 0.15 | 0.33 ± 0.09 |
| warpPIE10P | 0.25 ± 0.08 | 0.26 ± 0.06 | 0.25 ± 0.06 | 0.32 ± 0.05 | 0.25 ± 0.09 | 0.31 ± 0.11 | 0.57 ± 0.08 | 0.41 ± 0.08 | 0.38 ± 0.12 |
| TOX-171 | 0.31 ± 0.08 | 0.32 ± 0.04 | 0.32 ± 0.06 | 0.34 ± 0.05 | 0.32 ± 0.08 | 0.46 ± 0.14 | 0.49 ± 0.09 | 0.48 ± 0.06 | 0.57 ± 0.07 |
| pixraw10P | 0.25 ± 0.00 | 0.31 ± 0.10 | 0.35 ± 0.10 | 0.29 ± 0.06 | 0.30 ± 0.10 | 0.51 ± 0.11 | 0.49 ± 0.13 | 0.46 ± 0.10 | 0.36 ± 0.07 |
| CLL-SUB-111 | 0.34 ± 0.07 | 0.30 ± 0.09 | 0.26 ± 0.07 | 0.34 ± 0.07 | 0.29 ± 0.01 | 0.58 ± 0.08 | 0.48 ± 0.12 | 0.43 ± 0.11 | 0.55 ± 0.12 |
| Chess | 0.91 ± 0.01 | 0.91 ± 0.00 | 0.91 ± 0.00 | 0.91 ± 0.01 | 0.91 ± 0.01 | 0.89 ± 0.01 | 0.91 ± 0.01 | 0.91 ± 0.01 | 0.55 ± 0.00 |
| Coil2000 | 0.93 ± 0.01 | 0.93 ± 0.01 | 0.93 ± 0.00 | 0.93 ± 0.01 | 0.93 ± 0.01 | 0.92 ± 0.01 | 0.93 ± 0.01 | 0.93 ± 0.01 | 0.92 ± 0.02 |
| Penbased | 0.82 ± 0.00 | 0.82 ± 0.00 | 0.82 ± 0.00 | 0.82 ± 0.00 | 0.82 ± 0.00 | 0.82 ± 0.01 | 0.82 ± 0.00 | 0.82 ± 0.00 | 0.79 ± 0.00 |
| Segment | 0.87 ± 0.00 | 0.88 ± 0.00 | 0.87 ± 0.00 | 0.86 ± 0.01 | 0.86 ± 0.01 | 0.86 ± 0.01 | 0.86 ± 0.01 | 0.85 ± 0.01 | 0.81 ± 0.03 |
| Texture | 0.92 ± 0.01 | 0.92 ± 0.00 | 0.92 ± 0.01 | 0.92 ± 0.00 | 0.92 ± 0.00 | 0.91 ± 0.02 | 0.92 ± 0.00 | 0.92 ± 0.01 | 0.91 ± 0.00 |

Tabla 6.5: Hipervolumen en proceso de prueba — 50 % de características iniciales

datos para poder realizar comparaciones con los resultados obtenidos por el algoritmo de CNSGA-II [4]. Estas comparaciones se realizan tanto para el 10 % de características iniciales como para el 50 %, donde este último corresponde a la comparación más fiel. Esto se debe a que los resultados obtenidos por CNSGA-II se obtuvieron utilizando un 50 % de características iniciales.

### 6.3.1. 10 % de características iniciales

En las tablas 6.6 y 6.7 se presentan las comparaciones con CNSGA-II utilizando un 10 % de características iniciales para las primeras cinco propuestas.

En la tabla 6.6 se puede observar cómo en ambos conjuntos de datos se obtiene el

mejor promedio en la propuesta ***ICM***. Por otra parte, en la tabla 6.7 se obtiene el mejor promedio de ***warpAR10P*** en ***IC*** y el mejor de ***warpPIE10P*** en ***VC***. Los resultados en las tablas no solo demuestran que la mayoría de los algoritmos propuestos fueron capaces de obtener mejores resultados y generalizar de mejor manera en comparación con CNSGA-II, sino que también se refuerza la idea de que la utilización de chi-cuadrado durante el proceso de NSGA-II es más beneficioso en términos generales. Mientras tanto, lo otro que también se puede destacar es que tanto ***VA*** como ***VH*** no tuvieron tanta capacidad de generalización.

| Propuesta | Base | IC | IH | ICM | IHM | VA | VC | VH | RC | CNSGA-II |
|---|---|---|---|---|---|---|---|---|---|---|
| Instancia | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE |
| warpAR10P | 0.66 ± 0.05 | 0.83 ± 0.03 | 0.78 ± 0.03 | 0.86 ± 0.02 | 0.78 ± 0.03 | 0.58 ± 0.04 | 0.83 ± 0.03 | 0.77 ± 0.01 | 0.78 ± 0.03 | 0.54 ± 0.03 |
| warpPIE10P | 0.94 ± 0.01 | 0.94 ± 0.01 | 0.95 ± 0.01 | 0.96 ± 0.01 | 0.95 ± 0.01 | 0.87 ± 0.01 | 0.94 ± 0.01 | 0.94 ± 0.01 | 0.90 ± 0.04 | 0.65 ± 0.01 |

Tabla 6.6: Comparación de hipervolumen en proceso de entrenamiento con CNSGA-II – 10 % de características iniciales.

| Propuesta | Base | IC | IH | ICM | IHM | VA | VC | VH | RC | CNSGA-II |
|---|---|---|---|---|---|---|---|---|---|---|
| Instancia | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE |
| warpAR10P | 0.29 ± 0.06 | 0.48 ± 0.09 | 0.40 ± 0.10 | 0.44 ± 0.13 | 0.36 ± 0.08 | 0.24 ± 0.13 | 0.40 ± 0.15 | 0.40 ± 0.15 | 0.33 ± 0.09 | 0.33 ± 0.04 |
| warpPIE10P | 0.35 ± 0.13 | 0.49 ± 0.09 | 0.47 ± 0.09 | 0.55 ± 0.10 | 0.38 ± 0.10 | 0.31 ± 0.11 | 0.57 ± 0.08 | 0.41 ± 0.08 | 0.38 ± 0.12 | 0.43 ± 0.04 |

Tabla 6.7: Comparación de hipervolumen en proceso de prueba con CNSGA-II – 10 % de características iniciales.

### 6.3.2. 50 % de características iniciales

Al observar la tabla 6.8, se puede apreciar que, entre las primeras cinco propuestas, solamente ***IH*** fue capaz de obtener mejores resultados que CNSGA-II en ***warpAR10P***, mientras que en ***warpPIE10P*** todas las implementaciones fueron capaces de superar al estado del arte. Esto muestra que, en comparaciones equitativas con respecto al porcentaje de características iniciales, las propuestas tuvieron mayores dificultades con respecto al conjunto ***warpAR10P*** debido a la complejidad que se ha estado comentando en los otros resultados presentados.

| Propuesta | Base | IC | IH | ICM | IHM | VA | VC | VH | RC | CNSGA-II |
|---|---|---|---|---|---|---|---|---|---|---|
| Instancia | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE |
| warpAR10P | 0.43 ± 0.01 | 0.51 ± 0.02 | 0.57 ± 0.01 | 0.52 ± 0.03 | 0.48 ± 0.03 | 0.58 ± 0.04 | 0.83 ± 0.03 | 0.77 ± 0.01 | 0.78 ± 0.03 | 0.54 ± 0.03 |
| warpPIE10P | 0.73 ± 0.02 | 0.73 ± 0.01 | 0.73 ± 0.01 | 0.74 ± 0.01 | 0.72 ± 0.01 | 0.87 ± 0.01 | 0.94 ± 0.01 | 0.94 ± 0.01 | 0.90 ± 0.04 | 0.65 ± 0.01 |

Tabla 6.8: Comparación de hipervolumen en proceso de entrenamiento con CNSGA-II – 50 % de características iniciales.

Con respecto a las comparaciones entre las variantes, ***RC*** y las primeras cinco propuestas, queda claro que utilizar un mayor porcentaje de características iniciales va a generar menores promedios de hipervolumen, razón por la cual las últimas cuatro implementaciones entregan mejores resultados.

Ahora bien, al observar la tabla 6.9, se puede apreciar que ninguna de las primeras cinco propuestas fue capaz de generalizar mejor que el estado del arte en ninguno de los dos conjuntos de datos, a diferencia del resto, donde los tres promedios subrayados en la tabla son los únicos promedios de hipervolumen que fueron capaces de obtener mejores resultados que el estado del arte. Entre esos promedios, cabe resaltar que ***VC***, la propuesta que obtuvo el mejor promedio en ambos casos, demostró tener una gran capacidad de generalización, obteniendo en total 3 de 4 mejores promedios en el proceso de prueba entre las tablas 6.7 y 6.9.

| Propuesta | Base | IC | IH | ICM | IHM | VA | VC | VH | RC | CNSGA-II |
|---|---|---|---|---|---|---|---|---|---|---|
| Instancia | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE | Prom ± DE |
| warpAR10P | 0.20 ± 0.07 | 0.25 ± 0.10 | 0.26 ± 0.09 | 0.27 ± 0.07 | 0.25 ± 0.09 | 0.24 ± 0.13 | 0.40 ± 0.15 | 0.40 ± 0.15 | 0.33 ± 0.09 | 0.33 ± 0.04 |
| warpPIE10P | 0.25 ± 0.08 | 0.26 ± 0.06 | 0.25 ± 0.06 | 0.32 ± 0.05 | 0.25 ± 0.09 | 0.31 ± 0.11 | 0.57 ± 0.08 | 0.41 ± 0.08 | 0.38 ± 0.12 | 0.43 ± 0.04 |

Tabla 6.9: Comparación de hipervolumen en el proceso de prueba con CNSGA-II – 50 % de características iniciales.

## 6.4. Tiempos obtenidos

En las tablas 6.10 y 6.11 se presentan los tiempos medidos por ejecución en segundos por cada inicialización en cada conjunto de datos.

En estas tablas se puede apreciar que la propuesta ***RC*** fue la que obtuvo menores tiempos de error en la gran mayoría de las instancias. Esto se debe a que esta implementación, al eliminar una gran cantidad de características desde el inicio del algoritmo, permite que el proceso de NSGA-II sea menos costoso al tener que manejar un menor volumen de información durante cada una de las generaciones.

Por otra parte, al observar las filas detenidamente, se puede apreciar que, si bien hay algunas filas donde los tiempos de ejecución entre las primeras cinco propuestas y los tiempos entre las variantes se mantuvieron parejos, hay otras filas donde se observan diferencias notorias. Estas diferencias de tiempo se deben a que los tiempos presentados fueron aquellos obtenidos durante la ejecución de las propuestas en paralelo, por lo que las limitaciones del equipo donde se ejecutaron las pruebas provocaron diversas fluctuaciones. Por esta razón, se cree que si cada uno de los casos se ejecuta por separado, se obtendrán menores tiempos de ejecución.

| Instancia | Base | IC | IH | ICM | IHM | VA | VC | VH | RC |
|---|---|---|---|---|---|---|---|---|---|
| SRBCT | 3,135 | 3,053 | 3,129 | 2,801 | 2,766 | 3,525 | 3,137 | 3,123 | 2,941 |
| DLBCL | 3,334 | 3,347 | 3,302 | 4,401 | 4,457 | 3,817 | 3,427 | 3,367 | 2,936 |
| 9Tumors | 3,181 | 3,234 | 3,201 | 5,836 | 5,800 | 3,736 | 3,253 | 3,209 | 2,953 |
| Leukemia1 | 7,156 | 3,267 | 3,287 | 3,323 | 3,518 | 3,716 | 3,373 | 3,254 | 2,839 |
| Leukemia2 | 8,866 | 4,211 | 4,019 | 4,023 | 3,904 | 4,828 | 4,667 | 4,381 | 3,206 |
| Brain Tumor1 | 7,344 | 3,374 | 3,344 | 6,203 | 5,688 | 3,812 | 3,458 | 3,509 | 2,919 |
| Brain Tumor2 | 8,381 | 3,794 | 3,689 | 6,710 | 7,033 | 4,295 | 4,015 | 4,032 | 4,436 |
| ProstateTumor | 9,085 | 4,127 | 4,129 | 3,891 | 3,857 | 4,977 | 4,502 | 4,485 | 3,386 |
| Lung Cancer | 7,076 | 7,116 | 7,124 | 5,653 | 6,137 | 7,610 | 6,555 | 6,865 | 3,376 |
| 11Tumors | 4,688 | 4,666 | 4,642 | 8,718 | 8,348 | 6,082 | 5,280 | 5,584 | 3,100 |
| warpAR10P | 3,412 | 3,413 | 3,391 | 2,813 | 2,825 | 3,570 | 3,204 | 3,252 | 2,973 |
| warpPIE10P | 3,607 | 3,643 | 3,602 | 3,008 | 2,966 | 3,910 | 3,502 | 3,353 | 3,750 |
| TOX-171 | 8,363 | 4,374 | 4,260 | 4,016 | 3,986 | 5,354 | 5,036 | 4,783 | 3,393 |
| pixraw10P | 3,719 | 3,777 | 3,773 | 3,385 | 3,319 | 4,448 | 3,900 | 3,702 | 3,319 |
| CLL-SUB-111 | 5,182 | 5,443 | 5,157 | 7,934 | 7,422 | 6,360 | 5,970 | 5,727 | 3,018 |
| Chess | 4,878 | 4,889 | 4,890 | 7,609 | 7,607 | 5,059 | 4,374 | 4,626 | 4,319 |
| Coil2000 | 9,445 | 9,523 | 9,398 | 13,343 | 13,606 | 8,870 | 8,444 | 8,157 | 7,977 |
| Penbased | 8,589 | 8,555 | 8,679 | 10,616 | 10,686 | 9,718 | 9,070 | 8,789 | 8,789 |
| Segment | 5,193 | 5,170 | 5,205 | 3,945 | 3,969 | 4,604 | 4,189 | 4,170 | 4,050 |
| Texture | 6,831 | 6,944 | 6,927 | 6,438 | 6,414 | 7,266 | 6,717 | 6,751 | 6,644 |

Tabla 6.10: Tiempo promedio de ejecución de los experimentos (en segundos) — 10 % de características iniciales.

| Instancia | Base | IC | IH | ICM | IHM | VA | VC | VH | RC |
|---|---|---|---|---|---|---|---|---|---|
| SRBCT | 8,474 | 3,606 | 3,400 | 6,779 | 3,477 | 3,525 | 3,137 | 3,123 | 2,941 |
| DLBCL | 5,082 | 5,353 | 5,317 | 4,588 | 4,572 | 3,817 | 3,427 | 3,367 | 2,936 |
| 9Tumors | 3,875 | 4,379 | 4,228 | 5,416 | 4,524 | 3,736 | 3,253 | 3,209 | 2,953 |
| Leukemia1 | 5,927 | 4,177 | 4,198 | 4,152 | 4,394 | 3,716 | 3,373 | 3,254 | 2,839 |
| Leukemia2 | 7,353 | 7,729 | 7,648 | 7,748 | 7,920 | 4,828 | 4,667 | 4,381 | 3,206 |
| Brain Tumor1 | 10,636 | 4,543 | 4,718 | 4,652 | 4,722 | 3,812 | 3,458 | 3,509 | 2,919 |
| Brain Tumor2 | 16,503 | 7,509 | 7,131 | 7,015 | 7,105 | 4,295 | 4,015 | 4,032 | 4,436 |
| ProstateTumor | 19,266 | 7,943 | 7,868 | 13,442 | 8,027 | 4,977 | 4,502 | 4,485 | 3,386 |
| Lung Cancer | 13,480 | 11,740 | 11,766 | 17,659 | 15,701 | 7,610 | 6,555 | 6,865 | 3,376 |
| 11Tumors | 11,542 | 11,297 | 9,681 | 10,684 | 10,633 | 6,082 | 5,280 | 5,584 | 3,100 |
| warpAR10P | 4,150 | 4,222 | 4,088 | 4,963 | 7,038 | 3,570 | 3,204 | 3,252 | 2,973 |
| warpPIE10P | 4,675 | 4,664 | 4,597 | 4,833 | 6,841 | 3,910 | 3,502 | 3,353 | 3,750 |
| TOX-171 | 15,604 | 8,788 | 8,869 | 16,892 | 15,390 | 5,354 | 5,036 | 4,783 | 3,393 |
| pixraw10P | 15,551 | 6,070 | 6,169 | 6,086 | 5,998 | 4,448 | 3,900 | 3,702 | 5,672 |
| CLL-SUB-111 | 26,740 | 25,592 | 25,009 | 23,750 | 22,306 | 6,360 | 5,970 | 5,727 | 3,018 |
| Chess | 4,866 | 4,912 | 4,953 | 4,269 | 4,801 | 5,059 | 4,374 | 4,626 | 4,319 |
| Coil2000 | 21,548 | 9,383 | 9,951 | 8,535 | 10,524 | 8,870 | 8,444 | 8,157 | 7,977 |
| Penbased | 8,815 | 8,793 | 8,828 | 10,411 | 10,406 | 9,718 | 9,070 | 8,789 | 9,827 |
| Segment | 5,217 | 5,220 | 5,215 | 7,134 | 4,105 | 4,604 | 4,189 | 4,170 | 4,050 |
| Texture | 6,780 | 6,601 | 6,757 | 10,050 | 6,294 | 7,266 | 6,717 | 6,751 | 6,644 |

Tabla 6.11: Tiempo promedio de ejecución de los experimentos (en segundos) — 50 % de características iniciales.

# Capítulo 7

# Conclusiones

Este trabajo abordó el problema de selección de características en escenarios de alta dimensionalidad desde una perspectiva de optimización multi-objetivo, empleando el algoritmo evolutivo NSGA-II como técnica base. A partir del análisis de los desafíos asociados a la inicialización de la población y al diseño de los operadores de transformación, se propuso un enfoque que integra métodos de selección de características basados en filtros, específicamente la prueba estadística de chi-cuadrado, con el objetivo de mejorar la eficiencia y la calidad del proceso de búsqueda. En este contexto, se evaluaron distintas estrategias de inicialización utilizando esta métrica, incluyendo una estrategia puramente basada en chi-cuadrado, una estrategia híbrida que combina información estadística con un factor de aleatoriedad, y variantes orientadas a fomentar la diversidad en la población inicial. Adicionalmente, se propuso un proceso de reducción de dimensionalidad previo a la etapa de inicialización, orientado a reducir el espacio de búsqueda y facilitar la exploración evolutiva. Finalmente, se incorporó una variante del operador de mutación bit-flip guiada por chi-cuadrado, con el propósito de orientar la búsqueda hacia regiones más informativas del espacio de soluciones. Asimismo, se analizó el impacto del uso de distintos porcentajes iniciales de características (10 % y 50 %) sobre la calidad de las soluciones finales obtenidas.0000

Los resultados obtenidos a lo largo de la evaluación experimental indican que la hipótesis planteada puede considerarse validada, aunque no en su totalidad, dado que las

estrategias propuestas no presentan un comportamiento completamente consistente en todas las instancias evaluadas, sino que su desempeño varía entre conjuntos de datos en términos de reducción de dimensionalidad y error de clasificación. En particular, la incorporación de métodos de selección de características basados en filtros, como la prueba estadística de chi-cuadrado, permitió descartar características irrelevantes desde etapas tempranas del proceso, ya sea mediante un filtrado previo o a través de estrategias de inicialización y mutación guiadas. Esta reducción del espacio de búsqueda se reflejó en frentes de Pareto más compactos y en soluciones que lograron un mejor equilibrio entre el número de características seleccionadas y el error de clasificación, especialmente en escenarios de alta dimensionalidad y al utilizar porcentajes iniciales reducidos de características, observándose en general un mejor desempeño al emplear un 10 % de características iniciales en comparación con el 50 %. En términos numéricos, se observó que el método de reducción de características superó al método base en las tablas de hipervolumen 6.2, 6.4 y 6.5, donde obtuvo mayores valores promedio en 11, 15 y 15 conjuntos de datos, respectivamente. En contraste, en la tabla 6.3 ambos métodos presentaron un desempeño comparable, ya que la propuesta logró mejores resultados en 10 de los 20 conjuntos de datos evaluados.

De igual manera, el análisis de los tiempos de cómputo observado en la sección 6.4 del capítulo 6 muestra que las mejoras observadas en la calidad de las soluciones no implicaron un aumento significativo del costo computacional. Por el contrario, la reducción de dimensionalidad basada en la prueba estadística de chi-cuadrado contribuyó a mejorar la eficiencia del proceso evolutivo al disminuir la complejidad del espacio explorado, lo que permitió mantener e incluso reducir los tiempos de ejecución en comparación con la versión base de NSGA-II y con la mayoría de las propuestas evaluadas. En conjunto, estos resultados evidencian que el uso de estrategias de reducción de características, inicialización y operadores de mutación guiados por métodos de filtro representa una alternativa efectiva para mejorar el rendimiento global del proceso de selección de características, sin introducir sobrecostos computacionales relevantes, respaldando así la hipótesis planteada.

De acuerdo con los resultados obtenidos, es posible afirmar que el objetivo general

de este trabajo fue cumplido, dado que se logró adaptar el algoritmo NSGA-II para abordar el problema de selección de características como una tarea de optimización multi-objetivo. Esta adaptación se llevó a cabo a través de la incorporación de estrategias de inicialización basadas en métodos de filtro, un proceso explícito de reducción de dimensionalidad y operadores de mutación guiados, lo que permitió integrar información estadística relevante dentro del proceso evolutivo sin alterar la estructura base del algoritmo. De esta forma, NSGA-II fue modificado de manera efectiva para abordar escenarios de alta dimensionalidad, manteniendo un equilibrio entre reducción de características, desempeño en clasificación y eficiencia computacional.

En relación con los objetivos específicos planteados, los resultados obtenidos permiten extraer las siguientes conclusiones:

- Se lograron proponer e implementar estrategias de inicialización y operadores de mutación guiados que incorporan métodos de selección de características basados en filtros, integrando la prueba estadística de chi-cuadrado tanto en la etapa inicial como durante el proceso evolutivo. Estas estrategias permitieron orientar la búsqueda desde etapas tempranas y mejorar la calidad de las soluciones a lo largo de las generaciones.

- El rendimiento de las adaptaciones propuestas fue evaluado utilizando conjuntos de datos de distintas áreas y con diversas dimensionalidades, lo que permitió analizar su comportamiento en diversos escenarios y evidenciar que su efectividad depende del problema y del porcentaje inicial de características consideradas.

- Las comparaciones realizadas con la versión base de NSGA-II y con enfoques del estado del arte mostraron que las propuestas alcanzan resultados competitivos en términos de calidad de las soluciones y capacidad de generalización, particularmente en conjuntos de datos de alta dimensionalidad. En concreto, al comparar con CNSGA-II, se observa que varias de las estrategias propuestas logran mejores valores promedio de hipervolumen en los procesos de entrenamiento y prueba, especialmente al utilizar un 10 % de características iniciales. Por ejemplo, propuestas como $IC$, $ICM$ y $VC$ obtuvieron los mejores promedios de hipervolumen

en las cuatro comparaciones realizadas con un 10 % de características iniciales, mientras que ***VC*** y ***VH*** fueron las que obtuvieron los mejores en el 50 %.

En conjunto, los resultados obtenidos permiten reforzar las principales contribuciones de este trabajo al problema de selección de características desde una perspectiva de optimización multi-objetivo. En particular, se propone una estrategia de inicialización para NSGA-II que integra la prueba estadística de chi-cuadrado como un método de selección de características basado en filtro, junto con un proceso explícito de reducción de dimensionalidad aplicado previo al inicio del algoritmo, lo que permitió disminuir el espacio de búsqueda de manera efectiva. Asimismo, se incorporó un operador de mutación guiado por chi-cuadrado, orientado a equilibrar la exploración y la explotación durante el proceso evolutivo. Estas propuestas fueron validadas a través de una evaluación experimental, que consideró distintos porcentajes iniciales de selección de características y conjuntos de datos del mundo real provenientes de diversas áreas y con distintas dimensionalidades, evidenciando el impacto de las estrategias propuestas en términos de calidad de las soluciones, generalización de las implementaciones y eficiencia computacional del algoritmo.

## 7.1. Limitaciones

Si bien los resultados obtenidos permiten obtener conclusiones relevantes respecto al desempeño de las estrategias propuestas, es importante reconocer algunas limitaciones con respecto al alcance y a las condiciones experimentales del estudio. En primer lugar, la efectividad de las estrategias de inicialización, reducción de dimensionalidad y mutación guiada depende en gran medida de las características propias de cada conjunto de datos, tales como su dimensionalidad, el número de muestras y la relación entre las características y la variable objetivo. En este sentido, no todas las configuraciones evaluadas mostraron mejoras consistentes en todas las instancias, lo que quiere decir que el comportamiento del algoritmo puede variar según el contexto del problema abordado.

De la misma manera, la reducción de dimensionalidad propuesta se basa exclusivamente en el uso de la prueba estadística de chi-cuadrado como método de filtro. Si bien esta métrica es simple, robusta y ampliamente utilizada en la literatura, su naturaleza univariada implica que no considera de forma explícita las posibles interacciones entre características, lo que podría limitar su efectividad en escenarios donde esas interacciones juegan un rol determinante en el desempeño del modelo de clasificación. En mismo provoca que los resultados obtenidos reflejen el impacto de este método de filtro en particular y no necesariamente puede generalizarse a otros enfoques estadísticos o multivariados.

Por otra parte, los tiempos de cómputo presentados pueden verse influenciados por factores externos al diseño del algoritmo, como la disponibilidad de recursos computacionales en el momento de la ejecución de los experimentos. En este sentido, si bien esta variabilidad no afecta las conclusiones generales del estudio, no es posible asegurar que todas las ejecuciones hayan estado sujetas a condiciones idénticas de ejecución, lo que podría introducir pequeñas diferencias en los tiempos observados.

Finalmente, la validación experimental se llevó a cabo utilizando un conjunto acotado de algoritmos de clasificación y configuraciones específicas de parámetros para NSGA-II. Si bien estas elecciones son representativas y coherentes con trabajos relacionados, los resultados podrían variar al considerar otros clasificadores, métricas de evaluación o configuraciones alternativas del algoritmo evolutivo, lo que limita el alcance de las conclusiones obtenidas. Además, se utiliza únicamente un algoritmo del estado del arte para comparar los resultados obtenidos, lo que limitó el alcance comparativo del estudio.

## 7.2. Trabajo futuro

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

- Incorporación de nuevos algoritmos de filtro: Analizar el impacto de nuevas estrategias de inicialización basadas en filtros distintos al chi-cuadrado, como mutual information [19] (information mutua), con el fin de comparar su influencia sobre la calidad y generalización de las soluciones. La información mutua es uno de los métodos más utilizados en el estado del arte ya que es capaz de capturar relaciones lineales y no lineales entre las características y la variable objetivo, a diferencia de métricas basadas solo en correlación.

- 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.

- 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.

- Uso de distintos clasificadores y métricas de evaluación: Analizar el desempeño del algoritmo utilizando diferentes modelos de clasificación y métricas, con el objetivo de estudiar la respuesta de las soluciones obtenidas frente a cambios en el modelo de aprendizaje y en los criterios de evaluación empleados.

- Evaluación en entornos de ejecución controlados: Realizar experimentos en entornos de ejecución controlados que permitan analizar con mayor precisión el impacto computacional de las estrategias propuestas y obtener mediciones más fidedignas de los tiempos de cómputo.

En resumen, esta investigación muestra que la incorporación de pruebas estadísticas dentro del algoritmo NSGA-II contribuye a mejorar la calidad de las soluciones obtenidas en el problema de selección de características, particularmente en escenarios de alta dimensionalidad. Asimismo, los resultados obtenidos abren el camino hacia futuras mejoras orientadas a fortalecer la exploración, la generalización y la eficiencia del

proceso evolutivo, mediante la integración de información estadística en las distintas etapas del algoritmo.

Parte del trabajo desarrollado en esta investigación fue presentado en la conferencia CSCI 2025 [51] y publicado en la editorial Springer Nature [52], bajo el título Solving Large Feature Selection Problems using an Improved NSGA-II.

# Bibliografía


[1] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," Neurocomputing, vol. 300, pp. 70–79, 2018.

[2] J. Tang, S. Alelyani, and H. Liu, Feature selection for classification: A review, pp. 37–64. CRC Press, Jan. 2014. Publisher Copyright: © 2015 by Taylor Francis Group, LLC.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182–197, 2002.

[4] S. Z. Miyandoab, S. Rahnamayan, and A. A. Bidgoli, "Compact nsga-ii for multi-objective feature selection," in 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, Oct. 2023.

[5] Y. Xue, H. Zhu, and F. Neri, "A feature selection approach based on nsga-ii with relieff," Applied Soft Computing, vol. 134, p. 109987, 2023.

[6] I. Kononenko, "Estimating attributes: Analysis and extensions of relief," in Machine Learning: ECML-94, (Berlin, Heidelberg), pp. 171–182, Springer Berlin Heidelberg, 1994.

[7] X. Jin, A. Xu, R. Bie, and P. Guo, "Machine learning techniques and chi-square feature selection for cancer classification using sage gene expression profiles," in Data Mining for Biomedical Applications, pp. 106–115, Springer, 2006.

[8] B. Tran, B. Xue, and M. Zhang, "A new representation in pso for discretization-based feature selection," IEEE Transactions on Cybernetics, vol. 48, no. 6, pp. 1733–1746, 2018.

[9] Y. Xu, C. Yan, H. Liu, J. Wang, Z. Yang, and Y. Jiang, "Smart energy systems: A critical review on design and operation optimization," Sustainable Cities and Society, vol. 62, p. 102369, 2020.

[10] Y. S. Perera, D. Ratnaweera, C. H. Dasanayaka, and C. Abeykoon, "The role of artificial intelligence-driven soft sensors in advanced sustainable process industries: A critical review," Engineering Applications of Artificial Intelligence, vol. 121, p. 105988, 2023.

[11] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," Swarm and Evolutionary Computation, vol. 1, no. 1, pp. 32–49, 2011.

[12] C. Coello, G. Lamont, and D. V. Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems. New York, NY: Springer, 2007.

[13] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in Proc. 20th Int. Conf. Machine Learning (ICML), pp. 856–863, 2003.

[14] M. A. Hall, "Correlation-based feature selection for discrete and numeric class machine learning," in Proceedings of the Seventeenth International Conference on Machine Learning, ICML '00, (San Francisco, CA, USA), p. 359–366, Morgan Kaufmann Publishers Inc., 2000.

[15] M. Dash, H. Liu, and H. Motoda, "Consistency based feature selection," in Knowledge Discovery and Data Mining. Current Issues and New Applications, (Berlin, Heidelberg), pp. 98–109, Springer Berlin Heidelberg, 2000.

[16] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in Proc. 14th Int. Conf. Machine Learning (ICML), pp. 412–420, 1997.

[17] H. Gong, Y. Li, J. Zhang, B. Zhang, and X. Wang, "A new filter feature selection algorithm for classification task by ensembling pearson correlation coefficient and mutual information," Engineering Applications of Artificial Intelligence, vol. 131, p. 107865, 2024.

[18] W. Kirch, Encyclopedia of Public Health. Springer Dordrecht, 1 ed., 2008.

[19] J. R. Vergara and P. A. Estévez, "A review of feature selection methods based on mutual information," Neural Computing and Applications, vol. 24, no. 1, pp. 175–186, 2014.

[20] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning. Springer Series in Statistics, Springer New York, NY, 2 ed., 2009.

[21] M. Kelly, R. Longjohn, and K. Nottingham, "The uci machine learning repository," 2023.

[22] J. Alcalá-Fdez, L. Sánchez, S. García, M. J. del Jesus, S. Ventura, J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V. M. Rivas, J. C. Fernández, and F. Herrera, "KEEL: A software tool to assess evolutionary algorithms for data mining problems," Soft Computing, vol. 13, no. 3, pp. 307–318, 2009.

[23] M. Dash and H. Liu, "Consistency-based search in feature selection," Artificial Intelligence, vol. 151, no. 1, pp. 155–176, 2003.

[24] P. Cunningham and S. J. Delany, "k-nearest neighbour classifiers – a tutorial," ACM Comput. Surv., vol. 54, no. 6, 2021.

[25] HDS, "K-nearest neighbors algorithm: Classification and regression star," 2021.

[26] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.

[27] J. R. Quinlan, "Induction of decision trees," Machine Learning, vol. 1, no. 1, pp. 81–106, 1986.

[28] I. Inza, P. Larrañaga, R. Blanco, and A. J. Cerrolaza, "Filter versus wrapper gene selection approaches in dna microarray domains," Artificial Intelligence in Medicine, vol. 31, no. 2, pp. 91–103, 2004. Data Mining in Genomics and Proteomics.

[29] J. Kittler, "Feature set search algorithms," in Pattern Recognition and Signal Processing, pp. 41–60, Alphen aan den Rijn, Netherlands: Sijthoff & Noordhoff, 1978.

[30] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," Machine Learning, vol. 6, no. 1, pp. 37–66, 1991.

[31] P. Langley, W. Iba, and K. Thompson, "An analysis of bayesian classifiers," in Proceedings of the Tenth National Conference on Artificial Intelligence, (San Jose, CA, USA), pp. 223–228, AAAI Press, 1992.

[32] S. L. Salzberg, "C4.5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993," Machine Learning, vol. 16, no. 3, pp. 235–240, 1994.

[33] P. Clark and T. Niblett, "The cn2 induction algorithm," Machine Learning, vol. 3, no. 4, pp. 261–283, 1989.

[34] J. R. Quinlan, "Induction of decision trees," Machine Learning, vol. 1, no. 1, pp. 81–106, 1986.

[35] S. Ma and J. Huang, "Penalized feature selection and classification in bioinformatics," Briefings in Bioinformatics, vol. 9, pp. 392–403, 06 2008.

[36] Z. Zhao, F. Morstatter, S. Sharma, S. Alelyani, A. Anand, and H. Liu, "Advancing feature selection research," ASU Feature Selection Repository Arizona State University, pp. 1–28, 01 2010.

[37] J. Johnson, M. Douze, and H. Jégou, "Faiss: A library for efficient similarity search." https://engineering.fb.com/2017/03/29/data-infrastructure/faiss-a-library-for-efficient-similarity-search/, 2017.

[38] M. Amoozegar and B. Minaei-Bidgoli, "Optimizing multi-objective pso based feature selection method using a feature elitism mechanism," Expert Systems with Applications, vol. 113, pp. 499–514, 2018.

[39] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proceedings of ICNN'95 - International Conference on Neural Networks, vol. 4, pp. 1942–1948 vol.4, 1995.

[40] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," IEEE Transactions on Cybernetics, vol. 43, no. 6, pp. 1656–1671, 2013.

[41] Y. Zhang, D.-W. Gong, and J. Cheng, "Multi-objective particle swarm optimization approach for cost-based feature selection in classification," IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM, vol. 14, 09 2015.

[42] L. Cagnina, S. Esquivel, and C. A. C. Coello, "A particle swarm optimizer for multi-objective optimization," Journal of Computer Science and Technology, vol. 5, no. 04, pp. 204–210, 2005.

[43] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," Journal of Global Optimization, vol. 11, no. 4, pp. 341–359, 1997.

[44] A. J. Adoptante, A. Baes, J. C. Catilo, P. K. Lucero, A. L. De Ocampo, A. S. Alon D.Eng, and R. Dellosa, "Spoken-digit classification using artificial neural network," ASEAN Engineering Journal, vol. 13, pp. 93–99, 02 2023.

[45] A. Statnikov, C. F. Aliferis, I. Tsamardinos, D. Hardin, and S. Levy, "A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis," Bioinformatics, vol. 21, pp. 631–643, 09 2004.

[46] Z. Zhao, F. Morstatter, S. Sharma, S. Alelyani, A. Anand, and H. Liu, "Advancing feature selection research," ASU Feature Selection Repository Arizona State University, pp. 1–28, 01 2010.

[47] Scikit-learn, "sklearn.preprocessing.kbinsdiscretizer — scikit-learn documentation," 2024.

[48] C. Fonseca, L. Paquete, and M. Lopez-Ibanez, "An improved dimension-sweep algorithm for the hypervolume indicator," in 2006 IEEE International Conference on Evolutionary Computation, pp. 1157–1163, 2006.

[49] N. Beume, C. M. Fonseca, M. Lopez-Ibanez, L. Paquete, and J. Vahrenhold, "On the complexity of computing the hypervolume indicator," IEEE Transactions on Evolutionary Computation, vol. 13, no. 5, pp. 1075–1082, 2009.

[50] S. Prusty, S. Patnaik, and S. Dash, "Skcv: Stratified k-fold cross-validation on ml classifiers for predicting cervical cancer," Frontiers in Nanotechnology, vol. 4, p. 972421, 08 2022.

[51] "International conference on computational science and computational intelligence (csci 2025)." https://www.american-cse.org/csci2025/, 2025.

[52] "Springer nature." https://www.springernature.com/, 2025.