Curso 2006/07
CIENCIAS Y TECNOLOGÍAS/23
I.S.B.N.: 978-84-7756-770-7

MIGUEL GARCÍA TORRES

# Aplicación de técnicas metaheurísticas en minería de datos

Directores
JOSÉ MARCOS MORENO VEGA
BELÉN MELIÁN BATISTA

SERVICIO DE PUBLICACIONES
UNIVERSIDAD DE LA LAGUNA

SOPORTES AUDIOVISUALES E INFORMÁTICOS
Serie Tesis Doctorales

# Índice general

# Índice de figuras

# Índice de tablas

# Agradecimientos

Me gustaría agradecer a todas aquellas personas que me han ayudado a desarrollar y terminar la Tesis Doctoral.

Me gustaría agradecer a los directores de esta tesis, José Marcos Moreno Vega y Belén Melián Batista y a José A. Moreno Pérez por el apoyo recibido tanto durante los trabajos de investigación que hemos desarrollado conjuntamente como en la elaboración de esta Memoria de Tesis. Gracias a su apoyo he podido madurar como investigador.

Me gustaría también agradecer a Antonio Bahamonde, Catedrático de Universidad del Área de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Oviedo, por los consejos y recomendaciones dados, así como a todos los miembros de su grupo de investigación por haberme acogido como un compañero más. También me gustaría agradecer a Emilio Carrizosa, Catedrático de Estadística e Investigación Operativa de la Universidad de Sevilla, por sus aportaciones y buenos consejos. Agradezco a Joaquín Pacheco, de la Universidad de Burgos, por permitir que utilice un conjunto de datos que ha recopilado.

Agradezco a todos aquellos con los que he colaborado durante mi período de investigación en la Universidad de La Laguna, en particular a Félix C. García López, así como al resto de los compañeros del Departamento de Estadística, I. O. y Computación. Además, agradezco al Grupo de Computación Paralela por permitirme usar sus recursos de cómputo, financiado mediante el proyecto $TIN2005-08818-C04-04$. A todos ellos quisiera agradecerles el buen ambiente de trabajo en el que podido realizar esta memoria. También me gustaría agradecer al *Intelligent Systems Group*, dirigido por Pedro La-

rrañaga, por la ayuda prestada para la ejecución de la estrategia propuesta por ellos, permitiéndonos hacer uso del código que han implementado.

Para concluir, me gustaría dedicar esta memoria a mi hermano Juan José, a mi padre Juan José, a mis abuelos Miguel y Paula, al resto de mi familia, a Ana, a Enrique, a los amigos y muy especialmente a mi madre María Luisa y a Hilda por el apoyo y comprensión recibidos durante todo este período.

# Aportaciones

A lo largo de esta memoria se presentan los resultados de diversos trabajos de investigación que han sido publicados en revistas internacionales y presentados en congresos nacionales e internacionales.

El capítulo 3 estudia el rendimiento de la metaheurística Búsqueda Dispersa para el problema de selección de atributos. De dicha metaheurística, a pesar de su popularidad en problemas de optimización, sólo tenemos constancia de su aplicación al problema de selección de atributos en genes en combinación de otros algoritmos evolutivos como los Algoritmos Genéticos [92]. En el trabajo "Búsqueda Dispersa para el Problema de la Selección de Variables" [52] presentado en la Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA2003), se propuso una adaptación de dicha estrategia al problema. Este trabajo fue seleccionado para ser publicado en un volumen especial de la serie *Lecture Notes in Computer Science*, apareciendo bajo el título "Scatter Search for the Feature Selection Problem" [53]. Dicha adaptación fue evolucionando hasta la propuesta que se presenta en esta memoria, donde se estudian dos métodos de combinación y una paralelización que hace uso de ambos métodos. La comparativa entre la Búsqueda Dispersa y el Algoritmo Genético aparece en el trabajo "Búsqueda Dispersa y Algoritmo Genético para el Problema de la Selección de Variables" [48], presentado en el Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB2004). Las versiones secuenciales y paralelas de las Búsqueda Dispersa son analizadas en "Parallel Scatter Search for Solving Machine Learning Problems" [49], trabajo que se presentó en el Encuentro Anual organizado por el *Institute for Operations Research and the Management Science*

(INFORMS2004). Todos estos resultados se publicaron bajo el título "Solving Feature Selection Problem by a Parallel Scatter Search" [51] en la revista *European Journal of Operational Research*. Posteriormente se presentó una revisión de las distintas propuestas de paralelización de la Búsqueda Dispersa para el libro "Parallel metaheuristics: A New Class of Algorithms", apareciendo la adaptación presentada en este trabajo en el capítulo "Parallel Scatter Search" [50]. Por último se publicó el trabajo "Data Mining with Scatter Search" en *Lecture Notes in Computer Science*. [25].

En el capítulo 5 se estudia la metaheurística Búsqueda por Entorno Variable para el problema de selección de atributos. Como sucede con el caso anterior, a pesar de la popularidad de esta estrategia no se tiene constancia de que se haya aplicado a dicho problema con anterioridad. En este trabajo se propone una Búsqueda por Entorno Variable y una hibridación de ésta con la Búsqueda Tabú, introducciendo memoria a corto plazo. Además dichos resultados son comparados con el Algoritmo Genético. Este estudio fue presentada en el taller de metaheurísticas híbridas del congreso *European Conference on Artificial Intelligence* (ECAI2004) bajo el título "Solving Feature Subset Selection Problem by a Hybrid Metaheuristic" [54]. Este trabajo aún está en fase de mejoramiento y ampliación.

Ambas estrategias aparecieron en el trabajo "Búsquedas Dispersa y de Entorno Variable en Minería de Datos" [108] presentado en el taller de minería de datos en el Congreso Español De Informática (CEDI2005).

# Resumen

Con el paso de los años, la capacidad de almacenamiento y de adquisición de información se ha ido abaratando. Esto ha permitido que se quiera almacenar todos los datos posibles relacionados con el problema que se trata. Sin embargo, no todos estos datos contienen información útil que extraer. Con el aumento exponencial de la cantidad de datos que se pueden almacenar, surge la necesidad de explorar nuevas metodologías que sean capaces de tratarlos de una forma automática como hace el proceso de descubrimiento de conocimiento en bases de datos (KDD, *Knowledge Discovery in Databases*).

El proceso de descubrimiento de conocimiento en bases de datos permite la selección, limpieza, transformación y proyección de los datos; analizar los datos para extraer patrones y modelos adecuados; evaluar e interpretar los patrones para convertirlos en conocimiento; consolidar el conocimiento resolviendo posibles conflictos con conocimiento previamente extraído; y hacer el conocimiento disponible para su uso. Es el proceso global de descubrir conocimiento mientras que la etapa en la que se extraen los patrones y modelos corresponde a la minería de datos, que es la etapa en la que se aplican los métodos de aprendizaje y estadísticos.

Este trabajo de investigación tiene como objetivo el análisis, estudio y propuesta de estrategias metaheurísticas para el problema de la selección de atributos en tareas de clasificación.

El capítulo 1 introduce el proceso de extracción de conocimiento, profundizando en la etapa de minería de datos y más concretamente en la tarea de clasificación. Se explican los clasificadores estándares que serán usados a lo largo de los experimentos considerados, así como los métodos estánda-

res de evaluación que suelen usarse. Finalmente, se explica el aprendizaje automático y su relación con la minería de datos.

El capítulo 2 presenta una visión global de las metaheurísticas. Se definen y exponen las características deseables al diseñar una metaheurística. En función de las propiedades presentes se da una taxonomía de las metaheurísticas más extendidas en la literatura. El capítulo concluye introduciendo cada una de estas metaheurísticas con objeto de tener una visión global.

El problema de selección de atributos para tareas de clasificación es presentado en el capítulo 3. Para saber qué tipo de atributos tiene como objetivo la selección de atributos, se da una revisión de las diferentes definiciones de relevancia. La dificultad en definir el concepto de relevancia ha provocado que existan múltiples definiciones. Sin embargo, tal y como apuntan algunos autores, en este problema no suele perseguirse la selección de atributos relevantes sino óptimos para tareas de clasificación. Finalmente, se hace una revisión bibliográfica de las estrategias propuestas en la literatura para el problema de selección de atributos.

La primera propuesta que hacemos es una metaheurística Búsqueda Dispersa y en el capítulo 4 se describe su adaptación al problema. Se consideran dos métodos de combinación muy similares entre sí y una versión paralela que hace uso de ambos métodos de combinación. El rendimiento de las distintas versiones se estudia en la sección dedicada a la experiencia computacional. Los objetivos de los experimentos fueron estudiar los valores de los parámetros más convenientes para obtener el mejor rendimiento, estudiar el comportamiento de la versión paralela frente a las versiones secuenciales y por último comparar la Búsqueda Dispersa con otros algoritmos evolutivos. Las pruebas fueron realizadas sobre conjuntos de datos artificiales y reales.

En el capítulo 5 se presenta una adaptación de la metaheurística Búsqueda por Entorno Variable. A lo largo del capítulo se desarrolla su adaptación al problema de selección de atributos. Además, se propone una versión híbrida con la Búsqueda Tabú incorporando memoria a corto plazo. El objetivo de incorporar memoria es evitar que la búsqueda explore regiones del espacio previamente examinadas. El estudio del rendimiento de ambas propuestas nos sugieren que ambas tienen un rendimiento similar, pero la incorpora-

ción de memoria permite a la estrategia converger antes. El rendimiento de la Búsqueda por Entorno Variable es comparada con el de la búsqueda local empleada y otros algoritmos evolutivos. Al igual que con la Búsqueda Dispersa, los experimentos se realizaron en conjuntos de datos artificiales y reales.

Este trabajo termina con las conclusiones de los resultados mostrados a lo largo de este trabajo, y con las líneas de investigación futuras que se consideran más prometedoras.

# Capítulo 1

# Introducción

La minería de datos surge como consecuencia del reconocimiento del valor de la gran cantidad de datos que pueden almacenarse en un sistema informático. En diversas tareas, los datos son el objetivo perseguido, mientras que aquí es la fuente de la que se parte. Por lo tanto, los datos pasan de ser el producto a ser la materia prima y el producto es el conocimiento útil que ayuda a tomar decisiones en los ámbitos de donde fueron extraídos los datos.

La minería de datos ha sido usada como sinónimo de descubrimiento de conocimiento en bases de datos (del inglés *Knowledge Discovery in Databases*, KDD). Sin embargo, corresponde a una de las fases de todo el proceso de descubrimiento, encargada de hacer uso de técnicas de aprendizaje automático para desarrollar algoritmos capaces de aprender y extraer conocimiento de los datos.

En la siguiente sección describiremos en mayor detalle el proceso de descubrimiento (o extracción) de conocimiento, detallando las etapas de las que consta. Posteriormente nos centraremos en profundizar en la minería de datos. Por último, describiremos algunas de las técnicas de minería de datos que tienen una amplia aceptación en el campo.

## 1.1 La extracción de conocimiento

El objetivo fundamental del proceso de extracción de conocimiento en bases de datos es encontrar modelos inteligibles a partir de los datos. Los retos son trabajar con un gran volumen de datos, de modo que el proceso debería ser total o parcialmente automático y emplear técnicas con las que analizar los datos y extraer conocimiento útil. La automatización del proceso se enfrenta con la complejidad de los datos, así como con otros problemas como el ruido, los valores perdidos o la imprecisión. Una definición de dicho proceso es la siguiente:

**Definición 1.1** *(Fayyad et al. [39]). El descubrimiento de conocimiento en bases de datos es un proceso no trivial de identificación de patrones válidos, novedosos, parcialmente útiles y, en última instancia, comprensibles a partir de los datos.*

De la definición anterior se deducen una serie de propiedades que debería cumplir el conocimiento extraído:

- Válido. Los patrones encontrados deben describir datos nuevos.
- Novedoso. Debe aportar conocimiento nuevo.
- Potencialmente útil. La información debe ayudar en la toma de decisiones futuras.
- Comprensible. Los patrones encontrados deben ser suficientemente comprensibles para que proporcione conocimiento.

El proceso de descubrimiento de conocimiento se organiza en torno a cinco fases (Figura 1.1):

- Fase de integración y recopilación de datos. En esta etapa se analiza el dominio de aplicación y la información relevante que exista a priori. También se identifican los objetivos perseguidos por el usuario. Posteriormente, los datos se codifican en un formato común que permita trabajar de forma operativa con toda la información recogida sin que haya inconsistencias.

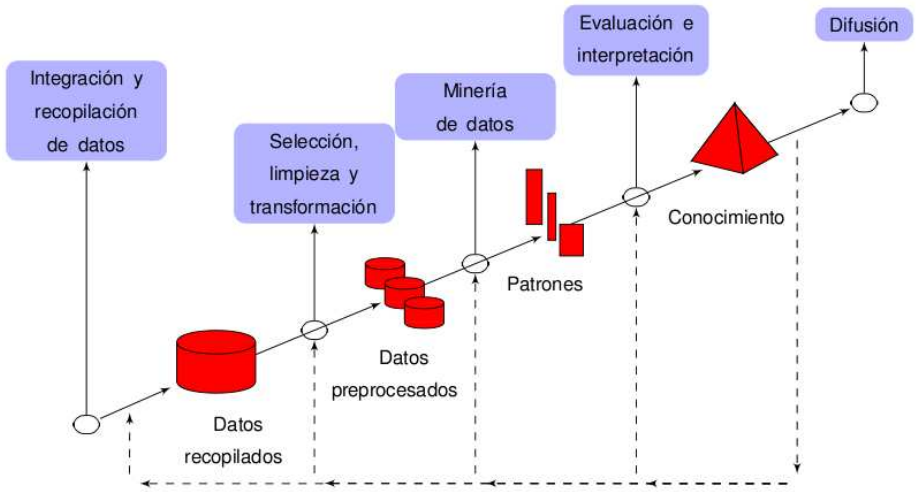


Figura 1.1: Etapas del proceso de extracción de conocimiento en bases de datos.

- Fase de selección, limpieza y transformación. Consiste en eliminar o corregir datos incorrectos, eliminar ruido si se considera apropiado, decidir estrategias para tratar valores perdidos, o seleccionar un subconjunto de variables o de ejemplos con objeto de facilitar la tarea de minería de datos .

- Fase de minería de datos. En esta etapa se define la tarea a llevar a cabo con los datos preprocesados en la etapa anterior. Algunas de estas tareas son clasificar, agrupar, etc.

- Fase de evaluación e interpretación. Se analizan y evalúan los patrones obtenidos. Una vez interpretados puede ser necesario volver a una etapa anterior.

- Fase de difusión. Se aplica el conocimiento extraído al dominio del problema, pudiendo ayudar en futuras tomas de decisiones.

A continuación pasamos a describir en mayor detalle cada una de estas etapas.

### 1.1.1 Fase de integración y recopilación de datos

Esta primera fase se centra en recopilar los datos necesarios para llevar a cabo un proceso KDD. Lo normal es que dichos datos procedan de fuentes distintas, con lo que usarán diferentes formatos de registro, diferentes tipos de error, etc. Así, para poder facilitar su manipulación posterior hay que integrarlos mediante tecnologías de almacenamiento de datos (en inglés *data warehousing*).

Estos almacenes de datos permiten agregar y cruzar eficientemente la información contenida en los datos, posibilitando su análisis multidimensional para verificar los patrones y las pautas hipotéticas sugeridas por el usuario. Este razonamiento deductivo difiere del razonamiento inductivo usado en la fase de minería de datos para descubrir patrones.

Ambas herramientas pueden complementarse para mejorar nuestro conocimiento acerca de las peculiaridades de los datos y así incrementar el éxito del proceso KDD.

### 1.1.2 Fase de selección, limpieza y transformación

En esta etapa se analizan los datos recopilados para poder seleccionarlos y prepararlos, ya que en la calidad del conocimiento encontrado también influye la calidad de los datos disponibles. Uno de los objetivos de este paso es prescindir de aquellos datos irrelevantes o innecesarios para la extracción de conocimiento. Algunas de las tareas de esta fase son la detección de valores anómalos, el tratamiento de valores perdidos, la reducción del número de ejemplos, la selección de atributos, la construcción de atributos y la modificación de los valores de los atributos.

Los valores anómalos pueden deberse tanto a un error en los datos, como a casos peculiares. Dependiendo de la tarea que estemos realizando, tras su detección, se pueden eliminar o tener en cuenta.

Los valores perdidos hay que tratarlos de forma especial. En algunos casos se puede prescindir de aquellos datos con valores perdidos, mientras que en otros se suelen hacer aproximaciones para otorgarles un valor determinado.

En caso de que la cantidad de datos sea excesiva, se puede recurrir a

considerar una muestra de los datos. Con esto aumentaríamos la eficiencia del proceso de minería de datos. La selección de esta muestra debe ser aleatoria y representativa de la de partida.

La selección de atributos es uno de los preprocesamientos que más interés recibe de la comunidad. Esto se debe a que permite la mejora de la eficiencia del proceso y posibilita una mayor comprensibilidad y eficacia del problema que se está tratando.

La construcción de atributos consiste en crear atributos a partir de los originales. Esto es útil cuando los originales tienen bajo poder predictivo por sí solos.

Por último, se puede modificar el tipo de valores que representa cada atributo. Las operaciones más conocidas son la discretización de atributos con valores reales o la numerización de valores discretos. La principal motivación es poder aplicar técnicas que sólo pueden aplicarse a un tipo de valores determinado.

### 1.1.3 Fase de minería de datos

El objetivo principal de esta etapa es descubrir patrones y relaciones en los datos proporcionados. Conviene aclarar, antes de continuar, que las tareas de minería de datos representan los distintos tipos de problemas que se tratan y cada una de estas tareas pueden ser abarcadas mediantes distintas técnicas.

De esta forma el proceso KDD conlleva resolver una tarea en la fase de minería de datos mediante un algoritmo que se encuadra en una de las técnicas existentes, las cuales llevan asociadas un paradigma. Cada paradigma consta de diversas propuestas de resolución de una tarea como la clasificación, la regresión, el agrupamiento, etc. En la sección dedicada a la minería de datos veremos esto con mayor detalle.

### 1.1.4 Fase de evaluación e interpretación

Los patrones descubiertos en la etapa anterior tienen que ser evaluados para medir su calidad. No existe un único criterio de evaluación, sino varios y la elección del criterio dependerá del contexto en el que nos encontremos. En clasificación, por ejemplo, suele usarse la precisión predictiva y en regresión el error cuadrático medio del valor predicho respecto al valor real. En tareas de agrupamiento las medidas de evaluación dependen del método considerado y por regla general suelen usarse medidas que tienen en cuenta la cohesión dentro de cada grupo y la separación entre grupos. En la literatura pueden encontrarse otras medidas adaptadas a las necesidades del problema.

En tareas predictivas como la clasificación, el modelo generado debe ser independiente del conjunto de datos utilizado en su evaluación para evitar resultados optimistas. Esto se consigue utilizando un conjunto de entrenamiento, con el que se induce el modelo, independiente del conjunto de prueba, con el que se mide la bondad del modelo.

Existen situaciones en las que la precisión no da buena información del modelo generado. Consideremos, por ejemplo, el caso en el que los datos pueden pertenecer a dos clases, y los pertenecientes a una de ellas representan el 90 % del total de los datos. Si el modelo generado clasifica todos los ejemplos en la clase más numerosa, tendremos una precisión del 90 %. Esto, en circunstancias normales, nos indicaría que tenemos un buen modelo predictivo. Sin embargo, esto no es cierto en este caso pues sólo asigna una clase. Para evitar este problema suele emplearse la matriz de confusión, que nos muestra el recuento de casos de las clases predichas.

La matriz de confusión $C$ es una matriz que nos detalla los errores de la predicción. Las filas representan cada una de las clases presentes en los datos y las columnas las clases en las que son predichas las instancias. El elemento $c_{ij}$ representa el número de ejemplos de la clase $i$ clasificados en la clase $j$. Por ejemplo, supongamos que tenemos tres clases $clase_1$, $clase_2$ y $clase_3$ y tenemos que predecir la clase de 100 instancias de modo que, tras ser clasificadas, tienen asociadas la siguiente matriz de confusión

| | | Clase predicha | | |
|---|---|---|---|---|
| | | $clase_1$ | $clase_2$ | $clase_3$ |
| Clase real | $clase_1$ | 25 | 0 | 6 |
| | $clase_2$ | 2 | 30 | 4 |
| | $clase_3$ | 8 | 0 | 25 |

Los elementos de la diagonal principal son aquellos que han sido correctamente clasificados (un total de 80). Un total de 6 ejemplos (en la posición $c_{13}$) con etiqueta $clase_1$ han sido clasificados como $clase_3$, teniendo una interpretación análoga el resto de los elementos.

### 1.1.5 Fase de difusión

Una vez construido el modelo, suele hacerse uso de éste, bien para tomar decisiones futuras, bien para aplicarlo sobre nuevos conjuntos de datos. A lo largo del tiempo debe medirse la evolución del modelo, pues puede que vayan surgiendo nuevos casos y esto provoque una degradación progresiva del modelo. Esto es importante en aplicaciones relacionadas con el comercio, donde el gusto de los consumidores se ve influenciado por factores externos no predecibles.

## 1.2 Minería de datos

En [146] se define la minería de datos como "el proceso de extraer conocimiento útil y comprensible desde grandes cantidades de datos almacenados en distintos formatos". Por tanto, el objetivo principal es encontrar modelos inteligibles a partir de estos datos.

Se caracteriza, tal y como señala Mitchell [105], por ser un área multidisciplinar en el que se aplican métodos de varias disciplinas como los presentes en sistemas de bases de datos, estadística, aprendizaje automático, visualización de datos, obtención de información, computación de altas prestaciones, reconocimiento de patrones, programación lógica inductiva, etc.

### 1.2.1 Tareas de la minería de datos

Las tareas pueden clasificarse como predictivas o descriptivas. Entre las predictivas encontramos la clasificación y la regresión, mientras que en las descriptivas destaca el agrupamiento.

En las tareas predictivas cada ejemplo de entrenamiento tiene asociado un conjunto de atributos de entrada y un atributo de salida, cuyo valor suele ser una categoría o un valor numérico. El objetivo es predecir el valor de salida, usando los valores de entrada que describen el ejemplo. Aquí se encuadran las tareas de clasificación y regresión. En las tareas descriptivas, los ejemplos se presentan sin etiquetar ni enumerar; con lo que sólo tienen atributos de entrada y el objetivo es describir estos datos. Algunas de estas tareas consisten en agrupar los datos como es el caso del agrupamiento.

A continuación pasamos a describir estas tareas destacadas:

- Clasificación. Cada caso tiene asociado un valor discreto. El valor de la etiqueta se suele llamar clase del caso. El objetivo es maximizar el poder predictivo de la clasificación de los nuevos casos.

- Regresión. Esta tarea es predictiva como el caso anterior, pero cada caso tiene asociado un valor real. El objetivo es entonces aprender una función real. Para ello, se puede minimizar el error entre el valor predicho y el valor real.

- Agrupamiento. Consiste en obtener los grupos que surgen de forma natural al aplicar criterios de similitud en los datos. Estos datos no tienen asociado ningún valor de salida a priori, sino que a lo largo del proceso se van agrupando, maximizando la similitud de los datos que pertenecen a un mismo grupo, y minimizando la de aquellos que pertenecen a grupos distintos.

### 1.2.2 Técnicas de la minería de datos

Como dijimos en la sección 1.1.3, existen diferentes técnicas con la que abarcar una tarea. Existen diferentes paradigmas detrás de las técnicas utilizadas.

Cada uno de estos paradigmas incluye diferentes algoritmos y variaciones de los mismos, así como restricciones que hacen que la efectividad del algoritmo dependa del dominio de aplicación. A continuación pasamos a mencionar algunas de estas técnicas relacionadas con la tarea de clasificación, por encuadrarse el presente trabajo en dicha tarea:

- Métodos bayesianos. El modelo aprendido asocia, a cada nuevo caso, una probabilidad de pertenencia a cada una de las clases existentes. Finalmente, se asigna a estos nuevos casos la clase que maximiza dicha probabilidad.

- Métodos basados en casos. No hay inducción del modelo sino que todos los datos son almacenados en memoria de modo que cada caso nuevo se intenta relacionar con los almacenados. En este tipo de aprendizaje hay que elegir una métrica.

- Métodos basados en núcleo. El modelo busca un discriminante lineal que maximice la distancia a los ejemplos fronterizos de las distintas clases.

- Métodos basados en árboles de decisión. El modelo consiste en una serie de decisiones organizadas jerárquicamente en forma de árbol. Para clasificar un nuevo caso se parte de la raíz del árbol y, comprobando en cada nodo el valor del atributo asociado a dicho nodo, se va descendiendo por la rama correspondiente al valor del atributo para el caso a clasificar.

### 1.2.3 El aprendizaje automático y su relación con la minería de datos

A continuación pasamos a describir el aprendizaje automático debido a que es una disciplina íntimamente relacionada con la minería de datos. Comenzaremos dando algunas de las definiciones que se encuentran en la literatura, y posteriormente se comentará su relación con la minería de datos. También expondremos los distintos paradigmas existentes.

### El aprendizaje automático.

El concepto de aprendizaje tiene un amplio abanico de definiciones debido a que es un término muy general. Sin embargo, todas las definiciones coinciden en que es un proceso en el que se aumenta el conocimiento y/o mejora la capacidad de actuación. Ribas et al. [126] dan la siguiente definición de aprendizaje.

**Definición 1.2** ***Aprendizaje:*** *Proceso cognitivo mediante el cual un agente adquiere conocimiento o aumenta la calidad y/o cantidad de su conocimiento, o mejora sus habilidades para realizar una tarea.*

Para diferenciarlo del procedimiento biológico de aprendizaje de los seres vivos, se denomina aprendizaje automático al campo cuyo objetivo es el estudio de programas que mejoran con la experiencia, aunque a partir de ahora, sin posibilidad de confusión, nos referirnos al aprendizaje automático como aprendizaje. Este campo de estudio es multidisciplinar puesto que los conceptos que se manejan abarcan diversas disciplinas como la estadística, la inteligencia artificial, la filosofía, la teoría de la información, la biología, la ciencia cognitiva, etc. Aunque todas estas disciplinas pretendan explicar la naturaleza del problema desde distintos puntos de vista, todas tienen en común que el objetivo perseguido en el aprendizaje es el de incrementar el conocimiento o las habilidades para cumplir una determinada tarea.

A lo largo de la bibliografía encontramos diversas definiciones del término. Una de las más recientes definiciones es la de Mitchell [105]:

**Definición 1.3** *Un programa de ordenador se dice que aprende de la experiencia E respecto a un tipo de tarea T y a la medida de rendimiento P, si su rendimiento en la tarea T, medida mediante P, mejora con la experiencia E.*

Sin embargo, existen otras definiciones como las que a continuación se enumeran.

**Definición 1.4** *(Simon [135]). Aprendizaje denota cambios adaptativos en el sistema en el sentido de que le permiten realizar la misma tarea, o una*

*tomada de la misma población, de una forma más eficiente y efectiva la próxima vez.*

**Definición 1.5** *(Carbonell [15]). Aprendizaje se puede definir operacionalmente como la habilidad para realizar nuevas tareas que no podía realizar anteriormente, o realizar anteriores tareas mejor, (más rápidas, más exactas, etc.) como resultado de los cambios producidos por el proceso de aprendizaje.*

**Definición 1.6** *(Ribas et al. [126]). Un sistema que aprende de forma automatizada (o aprendiz) es un artefacto (o un conjunto de algoritmos) que, para resolver problemas, toma decisiones basadas en la experiencia acumulada (en los casos resueltos anteriormente) para mejorar su actuación.*

A pesar de las diversas definiciones aquí expuestas, y de otras presentes en la bibliografía, encontramos coincidencias en diversos aspectos fundamentales: el de mejorar una tarea y el de realizar dicha mejora mediante un proceso que es análogo a la experiencia. El objetivo del aprendizaje automático puede verse como el de crear un programa que se adapte al problema considerado para mejorar la tarea a realizar. A lo largo del procedimiento se van persiguiendo algunas características:

- Generalidad. El ideal sería un programa cuya tarea de aprendizaje se pudiera adaptar a problemas de cualquier dominio. Sin embargo, encontramos que el procedimiento escogido puede ser más o menos conveniente según el dominio en el que nos encontremos. En el peor de los casos tenemos sistemas aplicables a un solo dominio.

- Eficiencia. El coste por llevar a cabo la tarea debe ser el menor posible. La medida puede ser el tiempo, o bien el número de reglas consideradas. En el peor de los casos una tarea sería procesada en un tiempo infinito aunque esto puede evitarse mediante algoritmos heurísticos.

- Robustez. Esta medida nos informa de la habilidad del sistema para poder llegar a un buen resultado con datos que puedan contener ruido e información parcialmente equivocada.

- Eficacia. Es una medida que nos informa de lo bien que es capaz de realizar la tarea encomendada. En parte engloba las medidas anteriores.

Otro aspecto importante es el dominio al que será aplicado. Dicho dominio puede englobarse en los siguientes grupos:

- Aprendizaje de conceptos. Consiste en obtener una función que relacione cada caso, de un determinado dominio, con un concepto, a través de un grado de pertenencia a dicho concepto. La mejora se enfoca en la tasa de error obtenida por la función de pertenencia.

- Clasificación. Es una generalización del caso anterior, de modo que se persigue encontrar durante el proceso de aprendizaje una función que relacione cada caso del dominio correspondiente con uno o más conceptos. En caso de que sólo pueda relacionarse con un solo concepto equivaldrá a tener un número de problemas de aprendizaje de conceptos igual al número de conceptos totales.

- Resolución de problemas. El objetivo es aprender una secuencia de reglas que permita resolver un determinado problema.

- Descripción de conceptos. Pretende llegar a la descripción bien mediante el descubrimiento de patrones, o por la formulación de una teoría que caracterice el conjunto de casos.

### Paradigmas de aprendizaje

El paradigma de aprendizaje nos informa de la filosofía del método seguido para llevar a cabo el aprendizaje. La clasificación de los métodos existentes depende de las propiedades o características a considerar. Ribas et al. [126], basándose en el tipo de selección y adaptación que un sistema realiza sobre la información disponible, distingue entre los siguientes tipos de aprendizaje:

- Deductivo o analítico. Deduce, a partir de un conjunto de hechos o reglas conocidas, otros conjuntos de hechos o reglas.

- Analógico. Resuelve un problema mediante el conocimiento adquirido al resolver otros similares.

- Inductivo. Usa una serie de ejemplos y contraejemplos para inducir la descripción de un concepto.

- Mediante descubrimiento. Aprendizaje sin ayuda externa. Uno de los procedimientos que más éxito tiene es el de agrupar los objetos más similares según el criterio considerado.

- Mediante Algoritmos Genéticos. Son un conjunto de técnicas inspiradas en los procesos de combinación y mutación ocurridos en los mecanismos biológicos. Tienen como uno de sus objetivos principales, dentro del aprendizaje automático, el descubrimientos de reglas mediante mecanismos evolutivos inspirados en la evolución natural de los seres vivos.

- Conexionista. Inspirado en las Redes Neuronales, considera al sistema como una red de nodos interconectados cuyos arcos tienen asociados unos valores, denominados pesos. El conjunto de entrenamiento modifica los pesos con el objetivo de obtener una salida lo más correcta posible.

Duda et al. [35], tomando en consideración el tipo de estrategia y las ayudas que recibe el sistema de aprendizaje, distingue tres tipos de aprendizaje:

- Supervisado. El aprendizaje de la tarea se realiza con un conjunto de ejemplos etiquetados, y posteriormente el sistema se enfrente a nuevos casos sin etiquetar.

- No supervisado. El objetivo es descubrir regularidades en los datos. Se espera que, debido a estos patrones, los ejemplos con mayor similitud se agrupen de forma natural en los mismos grupos.

- Mediante refuerzos. El aprendizaje se realiza mediante indicaciones o refuerzos positivos o negativos según realice correctamente o no la tarea.

### Relación con la minería de datos

Tal y como hemos visto en los apartados anteriores, tanto en la minería de datos como en el aprendizaje automático, el objetivo perseguido es el aprendizaje, por parte del programa, de un modelo que posteriormente sea usado para resolver el problema. Esto se debe a que la minería de datos hace uso de los modelos de aprendizaje para extraer conocimiento.

Sin embargo, el aprendizaje automático presenta ciertos problemas conceptuales cuando la definición se refiere al proceso de adquirir conocimiento. Una definición más operativa relega el aprendizaje a cuestiones de comportamiento en vez de conocimiento. De esta forma, sí podremos probar si un programa ha sido capaz o no de aprender, comparándolo con los resultados que obtenía antes del aprendizaje. Sin embargo, el aprendizaje en sí implica pensamiento, propósito, intencionalidad y esto no puede aplicarse a día de hoy a computadoras.

La minería de datos no tiene esta carga filosófica pues trata el aprendizaje desde un punto de vista práctico.

## 1.3 La tarea de clasificación

En la clasificación se dispone de un conjunto de variables predictoras $X$, denominadas atributos o características o variables de entrada, y otra categórica $\mathcal{C}$, denominada variable clase, etiqueta o variable de salida, cuyo valor es el que se pretende predecir.

El objetivo de la clasificación es generar un modelo predictivo, llamado clasificador, de modo que la variable clase sea expresada en función de las variables predictoras. Dado un conjunto de datos etiquetados $\mathcal{E}$, el clasificador representa una correspondencia $\mathcal{L} : X \rightarrow \mathcal{C}$ entre el conjunto de variables predictoras y la variable clase.

Las nuevas instancias podrán ser asignadas a las distintas clases existentes en función de los valores presentes en las variables de entrada.

En clasificación se tiene interés en modelizar las superficies de separación entre clases. Sin embargo, la modelización variará dependiendo de la técnica

de clasificación empleada.

### 1.3.1 Clasificadores

A continuación pasamos a describir los clasificadores que serán usados en este trabajo para evaluar el rendimiento de los algoritmos de selección de atributos.

#### Clasificador Naïve Bayes

El clasificador Naïve Bayes [34, 89] se encuadra en las técnicas bayesianas y hace uso del Teorema de Bayes [8] para generar el modelo predictivo. Este clasificador se fundamenta en la suposición de que todas las variables predictivas son estadísticamente independientes entre sí. A pesar de que esta suposición casi nunca se cumple, algunos estudios como el de Michie et al. [103] demuestran que los resultados que pueden obtenerse son competitivos e incluso superiores a otros clasificadores en algunos problemas.

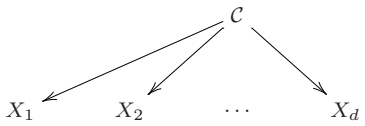


Figura 1.2: Topología del clasificador Naïve Bayes.

La independencia estadística entre los atributos da lugar a un modelo gráfico probabilístico en el que se tiene un nodo raíz (la variable clase), y tantos nodos hojas como atributos, que tienen como único padre la variable clase. La Figura 1.2 muestra esta estructura.

Haciendo uso del Teorema de Bayes y conociendo las probabilidades a priori, este clasificador estima la probabilidad de pertenencia de cada instancia a cada una de las clases posibles. Finalmente, la instancia se asigna a una clase en función de una regla de decisión. En problemas reales, las probabilidades son desconocidas; con lo que hay que estimarlas con el conjunto de ejemplos.

Dado el ejemplo $\mathbf{e} = (e_1, \ldots, e_d)$, con $e_i$ el valor observado para el i-ésimo atributo, la probabilidad a posteriori de que dados los valores de las variables predictivas $e_1, \ldots, e_d$, el ejemplo pertenezca a la clase $c_l \in D(\mathcal{C}) = \{c_1, \ldots, c_k\}$ con $D(\mathcal{C})$ el dominio de la variable clase, viene dado, según la regla de Bayes, por la siguiente expresión:

$$P(c_l|e_1, \ldots, e_d) = \frac{P(c_l) \prod_{i=1}^{d} P(e_i|c_l)}{P(e_1, \ldots, e_d)}$$

donde $P(c_l)$ es la probabilidad a priori de la clase $c_l$ y $P(e_i|c_l)$ es la probabilidad de observar el valor $e_i$ en el atributos $X_i$ en ejemplos que pertenecen a la clase $c_l$.

Si el atributo es discreto, las probabilidades se estiman basándose en la frecuencia de aparición. Si es continuo, se suele considerar que los valores siguen una distribución normal.

La clasificación de una instancia a una clase $c^*$ viene dada por la expresión:

$$c^* = \arg \max_{c_i \in \mathbf{C}} P(c_i|e_1, \ldots, e_d)$$

El clasificador Naïve Bayes es sencillo, rápido y proporciona buenos resultados, con lo que su uso en los experimentos está muy extendido.

### Clasificador del vecino más próximo

El clasificador del vecino más próximo [2] pertenece a los métodos basados en casos. Éstos se basan en la hipótesis de que los ejemplos de una clase determinada comparten propiedades y características entre sí. De esta forma, haciendo uso de una métrica determinada, las instancias se irán asignando al ejemplo con el que comparta más características (esté más próximo según la métrica considerada). Los fundamentos de este clasificador datan de la década de los 50 [41, 42] aunque la regla no se formuló formalmente hasta 1967 [20]. Desde entonces, la popularidad de este clasificador ha ido en aumento.

Estos clasificadores se caracterizan porque no construyen un modelo para todo el espacio de hipótesis, sino que realizan distintas aproximaciones para

distintas zonas del espacio. Esto hace que sean adecuados para problemas complejos y difíciles de modelar, aunque tiene como inconveniente el gran consumo de recursos. No llevan a cabo un aprendizaje propiamente dicho, sino que almacenan en memoria todos los ejemplos y a cada nueva instancia a clasificar se le asocia la clase del ejemplo más próximo almacenado en memoria. La elección de la métrica es muy importante pues los resultados pueden variar significativamente en un mismo conjunto de datos. Aunque la distancia Euclídea suele ser la más usada, dependiendo de las características de los datos, puede ser más apropiada alguna otra.

Otro factor que influye en los resultados es la escala de las variables predictivas. Si las variables se miden en diferentes escalas, suelen normalizarse para que su valor pertenezca al rango $[0, 1]$, mediante la relación

$$e_{ij}^n = \frac{e_{ij} - \text{mín}\, X_i}{\text{máx}\{X_i\} - \text{mín}\{X_i\}}$$

donde $e_{ij}^n$ es el valor normalizado de la variable $i$-ésima del caso $j$-ésimo ($e_{ij}$), mín$\{X_i\}$ es el menor valor de la variable $i$-ésima y máx$\{X_i\}$ el mayor.

Si la variable es categórica, no es necesario normalizar pues puede considerarse que, cuando tienen el mismo valor, la distancia es 0, y es 1 en otro caso.

Este clasificador puede generalizarse para el caso en que la clase asignada dependa de los $k$ vecinos más próximos. En este caso se asignará la clase mayoritaria presente en los vecinos. Un problema que surge es la elección adecuada de $k$. Para solventar este problema se han propuesto métodos que estiman, para un conjunto de datos, su valor adecuado [144] y alternativas como la de asociar pesos a los vecinos [36] en función de su proximidad a la instancia a clasificar.

### Clasificador C4.5

El clasificador C4.5 fue propuesto por Quinlan [123] como una mejora del anterior ID3 [120, 121, 122] y pertenece a la familia de los árboles de decisión. Un árbol de decisión no es más que un conjunto de decisiones organizadas en una estructura jerárquica. Dicha estructura está formada por un conjunto de

nodos, donde cada nodo que no es hoja contiene una condición sobre un atributo. El aprendizaje de esta estructura se lleva a cabo mediante algoritmos de partición o de Divide y Vencerás.

La clasificación de una instancia a una clase determinada se realiza partiendo de la raíz del árbol y en función del valor, en la instancia, del atributo asociado al nodo, ir descendiendo por la rama correspondiente hasta alcanzar el nodo hoja, que contiene la información de la clase que hay que asignar. En la Figura 1.3 podemos ver un ejemplo de árbol de decisión. En él, el objetivo es pronosticar si se va a poder jugar al golf o no en función del valor de un conjunto de variables predictivas relacionadas con el tiempo que va a hacer. Dependiendo del pronóstico, tendremos que considerar o no otras variables para para saber si podremos jugar o no.

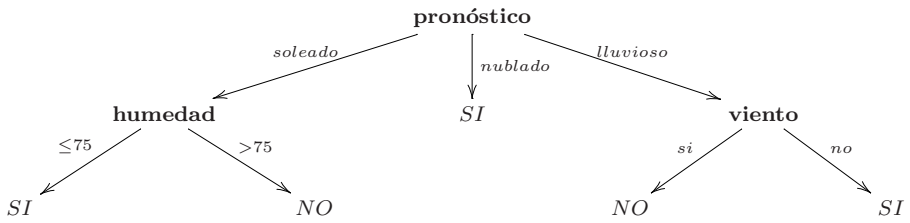


Figura 1.3: Árbol de Decisión.

Debido a que la clasificación trata con clases disjuntas, un árbol de decisión conducirá un ejemplo hasta una sola clase, con lo que las particiones existentes en el árbol son disjuntas y una instancia cumple o no una condición. Otras características de este clasificador son su robustez en un gran número de dominios, su bajo coste computacional, que lo han hecho muy popular, y su alta comprensibilidad.

Las mejoras principales que introduce el C4.5 respecto al ID3 son la alta eficacia con la que trata los valores perdidos, el manejo de atributos continuos considerando técnicas de discretización y la corrección de la tendencia del ID3 de seleccionar los atributos con muchos valores distintos.

### 1.3.2 Evaluación de un clasificador

La importancia de la evaluación de los algoritmos de aprendizaje radica en que posibilita el progreso de las técnicas desarrolladas. De esta forma, dado un dominio, se puede conocer cuál es la técnica más apropiada a utilizar una vez hecho unos estudios previos. La medida más usada es la tasa de acierto o precisión ($\Gamma$) (o su complementaria la tasa de error).

Dado una muestra $\mathcal{E}$, en la que cada uno de los $n$ ejemplos viene representado por la dupla $e(i) = \langle \mathbf{e_i}, c_i \rangle$, con $i = 1, \ldots, n$ y un clasificador $\mathcal{L}$, la tasa de acierto vendrá dada por la expresión:

$$\Gamma(\mathcal{E}, \mathcal{L}) = \frac{1}{n} \sum_{j=1}^{d} (\mathcal{L}(\mathbf{e_i}) = c_i) \tag{1.1}$$

donde la expresión $\mathcal{L}(\mathbf{e_i}) = c_i$ es 1, si es cierta, y 0 en otro caso.

Una vez obtenida la medida de evaluación, tenemos que considerar la técnica de evaluación que vamos a considerar durante los experimentos. Evaluar el rendimiento en el mismo conjunto de datos en el que ha sido inducido el clasificador nos da una estimación optimista [43] y en algunos casos irreal. Por ejemplo, en el caso del clasificador basado en la regla del vecino más próximo, tendríamos que el ejemplo más cercano a cada ejemplo a clasificar es el propio ejemplo, con lo que siempre se obtendría una precisión $\Gamma = 1$. El error obtenido en este caso se llama error aparente.

Para poder estimar el rendimiento de un clasificador es necesario usar un conjunto de ejemplos que no fue usado durante el proceso de inducción para garantizar la independencia entre ambos conjuntos. El conjunto considerado para la inducción del clasificador se denomina conjunto de entrenamiento y el que se destina a medir el rendimiento conjunto de prueba.

#### Métodos de evaluación

Los distintos métodos de evaluación proponen distintas estrategias de partición del conjunto de ejemplos para poder estimar el rendimiento de un clasificador.

La **validación simple** consiste en dividir el conjunto de ejemplos en uno de entrenamiento y otro de prueba para estimar la precisión. Es frecuente que se entrene con 2/3 de los ejemplos y se pruebe con el 1/3 restante. El principal inconveniente es que pierde información en el proceso de inducción del clasificador y puede que el conjunto de entrenamiento no sea representativo y esto por regla general no se conoce. Puede suceder que durante el proceso de partición, una clase se quede sin representación en uno de los conjuntos. Esto puede solventarse considerando la estratificación, que consiste en asegurar que todas las clases estén apropiadamente representadas en todas las particiones.

La **validación cruzada** es otro método que consiste en dividir el conjunto de ejemplos en $k$ pliegues o particiones mutuamente exclusivas, y hacer $k$ evaluaciones. En cada evaluación, se deja una partición como conjunto de prueba y se entrena con los otros $k-1$ conjuntos. La precisión estimada será el promedio de las obtenidas en las $k$ particiones. Como en el caso anterior también suele considerarse la **validación cruzada estratificada** .

Para poder estimar mejor la precisión, en caso de que el coste computacional lo permita, se suelen ejecutar diversas repeticiones de la validación cruzada.

### Comparación del rendimiento

Al aplicar dos clasificadores a un mismo problema, podemos saber, mediante una medida como la precisión, cuál es el que ha obtenido el mejor resultado. Sin embargo, puede que las diferencias se deban al método de evaluación considerado y no a un rendimiento distinto.

Para evitar esto suelen considerarse los test estadísticos, los cuales nos informan, con un cierto grado de confianza, si las diferencias en los resultados se deben a un distinto rendimiento o no.

Dietterich [28] compara diversos métodos de evaluación concluyendo que las pruebas estadísticas deben ser vistas como aproximadas por las dificultades presentes en los métodos de evaluación. Algunas de estas dificultades son el entrenamiento con muestras de menor tamaño que el original, asunción

de independencia, solapamiento de los subconjuntos de entrenamiento, etc. Finalmente, recomienda usar la prueba de McNemar si sólo se puede ejecutar una única vez el algoritmo (por ejemplo en caso de que la muestra sea tan grande que el tiempo que consuma sea excesivo), y la prueba $t$ de Student en caso de que pueda ejecutarse diez veces. Además propone aplicar la **validación cruzada 5x2**, que consiste en repetir 5 veces una validación cruzada con $k = 2$. Posteriormente, Alpaydin [4] propone aplicar la **validación cruzada 5x2** con la distribución $F$ de Snedecor en vez de la $t$ de Student pues concluye que tiene un menor error tipo I y una mayor potencia.

La comparación de los resultados obtenidos en un problema está sesgada debido a la dependencia de los distintos resultados al reusar ejemplos. Por ello Demšar [26] propone hacer las comparaciones sobre los resultados obtenidos sobre una serie de problemas, pues en este caso las discrepancias provienen de las diferencias presentes en los problemas. Además, debido a la independencia entre los problemas, los errores de tipo I no serán importantes. Del estudio realizado concluye que no existe un método estándar para comparar el rendimiento entre algoritmos de aprendizaje y que las pruebas no paramétricas son pruebas más adecuadas en el caso de que no podamos garantizar que los resultados siguen una distribución normal ni que haya homogeneidad de la varianza. Recomienda usar el test de Wilcoxon [145] en caso de que tengamos dos algoritmos, y el de Friedman [45, 46] en caso de que tengamos más de dos algoritmos. En caso de rechazar la hipótesis nula propone comparar los resultados dos a dos con el test de Nemenyi [112] o el de Bonferroni-Dum [37] mediante los procedimientos de Holm [72] o de Hochberg [70] .

# Capítulo 2

# Metaheurísticas

## 2.1 Introducción

Para introducir el concepto de metaheurística es conveniente introducir el de heurística pues el primero es una extensión del segundo relacionado con la forma de proceder en la resolución de un problema. El término heurística está vinculado a la resolución de problemas sin llevar a cabo un análisis riguroso de éste, obteniéndose como resultado una solución de alta calidad. Debido a esto los métodos heurísticos han sido también llamados algoritmos aproximados o métodos inexactos [6].

Dependiendo del área de conocimiento, el término heurístico tendrá una interpretación distinta. Para el área de Inteligencia Artificial, dicho término se asocia con cualquier procedimiento que, sin hacer uso de un análisis formal riguroso del problema, obtiene soluciones de alta calidad a un coste razonable de recursos empleando conocimiento relativo al problema. Sin embargo, en Investigación Operativa dicho calificativo hace referencia a la resolución de un problema de optimización mediante procedimientos no exactos que no garantizan la optimalidad ni factibilidad de las soluciones aunque éstas, una vez encontradas, sean óptimos locales y a veces globales a un coste de recursos razonable.

Un inconveniente importante para su uso es la ausencia de una teoría formal que garantice el grado de optimalidad de las soluciones encontradas; sin embargo su aplicación está en crecimiento y en continuo desarrollo. Las motivaciones para ello son varias [139], de las cuales podemos destacar las siguientes:

- No existencia de un método exacto de resolución o el excesivo uso de tiempo de cómputo o memoria del método exacto existente.
- No necesidad de encontrar el óptimo global, bastando con encontrar un buen óptimo local.
- Baja fiabilidad de los datos, de modo que carezca de interés encontrar la solución exacta y baste un valor aproximado.

Dos aspectos fundamentales relacionados con las heurísticas que exploran el espacio de soluciones son la intensificación y la diversificación. La intensificación se refiere a la capacidad de mejora de la solución durante la exploración del espacio de soluciones, mientras que la diversificación hace referencia a la capacidad de cubrir todo el espacio. Las propiedades deseables de una heurística no han sido estandarizadas, aunque están aceptadas de forma general las que exponemos a continuación [139]:

- Simplicidad, de modo que no presenten dificultad para su comprensión e implementación.
- Robustez, presentando poca sensibilidad a pequeñas variaciones del modelo.
- Generales, siendo aplicables a un amplia variedad de problemas.
- Eficaces, y así obtener una solución de alta calidad.
- Eficientes, para que el uso de los recursos disponibles (principalmente tiempo y memoria) sea razonable.
- Criterios de parada apropiados y dependientes de los resultados obtenidos.

Una solución asociada a un problema de optimización se dice que es heurística si ha sido obtenida mediante una aproximación heurística.

El término metaheurística se obtiene de anteponer a heurística el prefijo meta que significa "más allá" o "a un nivel superior". Una metaheurística es una estrategia que guía y modifica otras heurísticas para producir soluciones más allá de aquellas que se generan normalmente en una búsqueda de optimalidad local. El término metaheurística apareció por primera vez en un artículo sobre la Búsqueda Tabú de Fred Glover en 1986 [57]. Desde entonces han surgido multitud de propuestas de pautas para diseñar procedimientos de resolución de ciertos problemas. Las metaheurísticas pueden ser específicas a un problema determinado o generales. Las primeras tienen como objetivo obtener un mejor rendimiento a costa de una mayor especialización, pero las generales tienen como objetivo la sencillez, las adaptabilidad a una amplia variedad de problemas y robustez frente a pequeños cambios.

## 2.2 Clasificación de las metaheurísticas

Aunque hay una gran variedad de estrategias metaheurísticas, no existe un consenso común para su clasificación. Esto se debe, entre otras cosas, a la aparición de nuevas estrategias que no se ajustan a ninguna clase en concreto, así como a la diversidad de criterios que existen para llevar a cabo esta tarea. Birrattari et al. [10] proponen los criterios que a continuación presentamos:

**Continuidad o discontinuidad.** Si la búsqueda en el espacio de soluciones se desplaza de una solución a otra vecina durante todo el proceso estaremos ante un procedimiento de trayectoria continua mientras que si se producen saltos de una solución a otra más lejana estaremos en el discontinuo. El GRASP, la Búsqueda Local Iterada, la Búsqueda por Entorno Variable y los Algoritmos Evolutivos como el Algoritmo Genético, la Búsqueda Dispersa, la Colonia de Hormigas y el Algoritmo de Estimación de Distribuciones, pertenecen a este último grupo mientras que el Recocido Simulado, la Búsqueda Tabú, la búsqueda local y

la Búsqueda Local Guiada son métodos de trayectoria.

**Solución individual o población** En caso de que la búsqueda en el espacio de soluciones se realice en más de un punto tendremos un población de soluciones. Esta estrategia es utilizada en los Algoritmos Evolutivos. El resto pertenecen al caso de búsqueda mediante un punto, también llamadas búsquedas basadas en recorrido.

**Uso de memoria.** El uso de memoria se emplea para evitar buscar en zonas ya exploradas. Se usa explícitamente en la Búsqueda Tabú, la Colonia de Hormigas y la Búsqueda Local Guiada. En menor medida es usada en la Búsqueda Local Iterada, el Algoritmo Genético y la Búsqueda Dispersa. Para el resto de metaheurísticas aquí presentadas, consideraremos que no se hace uso de la memoria.

**Estructura de uno o varios vecinos.** En la mayoría de los casos el movimiento en el espacio de soluciones implica a una solución vecina como son el caso de la búsqueda local, la Búsqueda Tabú, el GRASP, la Búsqueda Local Guiada, la Colonia de Hormigas y el Recocido Simulado. Sin embargo, la incorporación de más de una vecindad permite una mayor capacidad exploratoria como en el caso de la Búsqueda por Entorno Variable , la Búsqueda Local Iterada y, en parte, el Algoritmo Genético y la Búsqueda Dispersa.

**Función objetivo dinámica o estática.** En algunos procedimientos se introduce una penalización en la función objetivo de modo que dicha función puede considerarse dinámica. Bajo esta consideración encontramos la Búsqueda Local Guiada y, parcialmente, la Búsqueda Tabú.

**Bioinspirado o no.** Una posible clasificación puede ser la de saber si ha sido inspirado por fenómenos naturales o no. Entre los procedimientos bioinspirados encontramos el Algoritmo Genético, la Colonia de Hormigas y el Recocido Simulado.

En la Tabla 2.1 podemos ver un resumen basado en la propuesta anterior.

Tabla 2.1: Resumen de los criterios presentes en las metaheurísticas según Birrattari et al. [10]. Las siglas de las metaheurísticas corresponden a su nombre en inglés. ILS (*Iterated Local Search*)para la Búsqueda Local Iterada, VNS (*Variable Neighbourhood Search*) para la Búsqueda por Entorno Variable, GA (*Genetic Algotrithm*) para el Algoritmo Genético, SS (*Scatter Search*) para la Búsqueda Dispersa, EDA (*Estimation Distribution Algorithm*) para el Algoritmo de Estimación de Distribuciones, SA (*Simulated Annealing*) para el Recocido Simulado, TS (*Tabu Search*) para la Búsqueda Tabú, LS (*Local Search*) para la búsqueda local y GLS (*Guided Local Search*) para la Búsqueda Local Guiada. Se considerará el símbolo – en caso de ausencia del criterio correspondiente, √ si la posee y ¶ si la posee parcialmente.

| Criterio | Continuidad | Poblacional | Memoria | Vecindad | f Dinámica | Bioinspirado |
|---|---|---|---|---|---|---|
| GRASP | – | – | – | – | – | – |
| ILS | – | – | ¶ | √ | – | – |
| VNS | – | – | – | √ | – | – |
| GA | – | √ | ¶ | ¶ | – | √ |
| SS | – | √ | ¶ | ¶ | – | – |
| EDA | – | √ | – | ¶ | – | – |
| SA | √ | – | – | – | – | √ |
| TS | √ | – | √ | – | ¶ | – |
| LS | – | – | – | – | – | – |
| GLS | √ | – | √ | – | √ | – |

Glover y Laguna [62] proponen que la clasificación de las metaheurísticas debe basarse en la memoria, el tipo de exploración y el número de soluciones con las que se trabaja en cada iteración. En una reciente publicación Melián et al. [99] proponen una clasificación basada en el tipo de procedimiento que lleva a cabo la metaheurística. Consideran que los tipos fundamentales son las metaheurísticas de relajación, constructivas, de búsqueda y las evolutivas.

**Metaheurísticas de relajación.** Son aquellas que se basan en la modificación del problema original en otro cuya resolución sea más sencilla. Para abordar un problema, ha de obtenerse un modelo que permita emplear una técnica de resolución adecuada. Sin embargo, en multitud de ocasiones dichos modelos planteados suelen ser de difícil resolución, de modo que habrá que acudir a modelos simplificados del original que permitan encontrar de forma más sencilla buenas soluciones con mayor eficiencia. Una relajación consiste en la simplificación del modelo

original debido a la eliminación, debilitación o modificación de las restricciones. Algunas relajaciones consisten en modificar las restricciones, la función objetivo, el objetivo del problema, etc.

**Metaheurísticas constructivas.** Este tipo de metaheurísticas consisten en una construcción iterativa de la solución mediante la incorporación de elementos a una estructura inicialmente vacía. Para ello hay que seleccionar la estrategia constructiva.

**Metaheurísticas de búsqueda.** Se caracterizan porque partiendo de una solución inicial, recorren el espacio de soluciones haciendo uso de operaciones de transformación y movimiento. Este espacio de soluciones consiste en el conjunto de soluciones candidatas alternativas. Para ello habrá que considerar una codificación que dependerá del problema y en algunos casos de la propia metaheurística. La elección de las transformaciones o movimientos es importante para el éxito de la metaheurística en la resolución del problema.

**Metaheurísticas evolutivas.** Se caracterizan por partir de un conjunto de soluciones, generalmente llamado población, que, en cada iteración, evoluciona a otra población por creación de nuevas soluciones y/o eliminación de otras de la población actual. El tipo de estrategia evolutiva considerada para guiar la evolución será la que distinga entre las distintas propuestas existentes.

Otros tipos de metaheurísticas corresponden a tipos intermedios entre los anteriores descritos. Caben destacar las metaheurísticas de descomposición y las de memoria a largo plazo. Las de descomposición resuelven un problema subdividiéndolo en subproblemas a partir de los cuales se construye una solución del problema original. Este tipo es un híbrido entre las de relajación y las constructivas. Las metaheurísticas de búsqueda con memoria a largo plazo se consideran de búsqueda, de modo que son híbridos de las metaheurísticas Multiarranque y Búsqueda Tabú.

Sin embargo, todas las metaheurísticas existentes pueden concebirse como estrategias aplicadas a un proceso de búsqueda en el espacio de las soluciones

a través de operaciones de movimiento o transformación. Debido a esto en el presente trabajo trataremos todas las metaheurísticas presentadas como de búsqueda, quedándonos con el criterio poblacional anteriormente expuesto como principal elemento diferenciador de las estrategias existentes.

## 2.3 Metaheurísticas de búsqueda

La resolución del problema se realiza mediante una búsqueda en el espacio de soluciones candidatas alternativas. Para ello las soluciones deberán estar representadas mediante una codificación que incluya la información necesaria para su identificación y evaluación. Una búsqueda sobre un espacio consiste en generar una sucesión de puntos del espacio pasando de uno a otro mediante una serie de movimientos.

El proceso de resolución puede resumirse en una etapa inicial, en la que se genera una solución inicial, y otra posterior iterativa, en la que se aplica un movimiento, que finalizará cuando se cumpla la condición de parada. Un aspecto importante es la medida de calidad asociada a cada solución, la cual se lleva a cabo mediante una o varias funciones objetivos que tienen en cuenta las restricciones del problema. La estrategia de búsqueda puede incorporar elementos de una o varias metaheurísticas generales, junto a mecanismos de metaheurísticas específicas del problema.

Dada la generalidad de las metaheurísticas, la descripción y el análisis se llevará a cabo sobre los problemas de optimización. Un problema de optimización es aquel cuya solución implica encontrar, de entre todas las soluciones candidatas alternativas, aquella que satisfaga, en mejor medida, los objetivos del problema. Dichos objetivos suelen formalizarse en una o varias funciones objetivos que habrá que maximizar o minimizar.

Un problema de optimización viene descrito por el espacio de soluciones y una o varias funciones objetivos. El espacio de soluciones alternativas puede expresarse mediante un conjunto finito de variables $X = \{X_i : i = 1, \ldots, d\}$. Sea $U_i$ el dominio de cada una de estas $d$ variables, entonces una solución del problema se obtiene de seleccionar los valores $x_i$ asignados a cada una de

estas variables $X_i$ del dominio $U_i$ de modo que satisfaga ciertas restricciones y optimice la función objetivo $f$. El universo de soluciones consiste en el conjunto $U = \{x = (x_i : i = 1, \ldots, d) : x_i \in U_i\}$. Sin embargo, no todas ellas cumplen las restricciones, de modo que al subconjunto de soluciones que sí lo cumplen se representará por $D \subseteq U$, y es llamado espacio factible. Dados el espacio factible $D$ y la función objetivo $f$, la resolución de un problema de optimización $(D, f)$ consiste en encontrar una solución factible $x^* \in D$ tal que $f(x^*) \leq f(x)\ \forall\ x \in D$ en caso de minimización. Análogo para el caso de maximización.

Las soluciones que se obtienen partiendo de una solución cualquiera mediante la aplicación de un movimiento se denominan soluciones vecinas y constituyen su entorno. El conjunto de movimientos posibles da lugar a una estructura de entornos en el espacio de soluciones. Formalmente una estructura de entornos sobre un espacio de búsqueda $U$ es una función $\mathcal{N} : U \to 2^U$ que asocia a cada solución $x \in U$ un entorno $\mathcal{N}(x) \subseteq U$ de soluciones vecinas a $x$.

En la elección de la estructura de entornos puede tenerse en cuenta sólo el subconjunto de soluciones factible del problema. Esto, a pesar de que suele implicar una mayor complejidad en los movimientos a realizar, también suele ser más eficiente, sobre todo cuando la comprobación de la factibilidad de una función sea costosa. En caso de considerar $D$ como espacio de búsqueda, la estructura de entornos será una función $\mathcal{N} : D \to 2^D$ que asocia a cada solución $x \in D$ un entorno $\mathcal{N}(x) \subseteq D$ de soluciones vecinas a $x$.

Existe una gran diversidad de propuestas de condiciones de parada. Las más simples son aquellas que tienen en cuenta el número de iteraciones y el tiempo transcurrido; sin embargo, aquellas dependientes de la metaheurística y del problema suelen ser más adecuados. La descripción de la búsqueda por entornos se muestra en la Figura 2.1 para problemas de minimización. A partir de ahora consideraremos en los pseudocódigos, sin pérdida de generalidad, que la función a optimizar es de minimización.

A continuación pasamos a describir en mayor detalle las metaheurísticas. Basándonos en el espacio de búsqueda que recorren, podemos considerar las búsquedas locales y las búsquedas globales. Las primeras buscan soluciones

```
1:  Procedimiento Búsqueda por entornos
2:  {
3:  x ← GeneraSolucion(U);
4:  x* ← x;
5:  repetir
6:     x ← SeleccionaSolucion(N(x));
7:     si (f(x) < f(x*))
8:        x* ← x;
9:  hasta (CriterioDeParada)
10: }
```

Figura 2.1: Pseudocódigo de la búsqueda por entornos.

del entorno de la solución actual, mientras que las segundas pretenden escapar de los óptimos locales donde se estancan las primeras.

## 2.4 Búsquedas locales

Se basan en el análisis del entorno de la solución que se dispone. De esta forma, el espacio de soluciones es estudiado de forma local. Según como se seleccione la solución del entorno de la solución actual, pueden distinguirse las búsquedas locales no informadas y las búsquedas locales monótonas . Las primeras sólo tienen en cuenta la estructura de entornos para guiar la búsqueda, mientras que las segundas hacen uso de la evaluación de las soluciones vecinas para guiar la búsqueda.

### Búsquedas no informadas

Llevan a cabo la búsqueda considerando la estructura de entornos en el espacio de búsqueda sin hacer uso de la información proporcionada por la evaluación de la función objetivo. Este tipo de búsquedas aportan estrategias para organizar la exploración del espacio de búsqueda de modo eficiente. Los dos tipos de búsquedas más usuales que se enmarcan en esta categoría

son la búsqueda local exhaustiva y la búsqueda local parcial.

La búsqueda local exhaustiva es aquella que busca la solución óptima de entre todas las soluciones factibles. Se emplea una ordenación de todas las soluciones y una transformación que obtenga, en cada iteración, la siguiente solución de la ordenación considerada. Dichas soluciones pueden reducirse a las soluciones factibles o bien a un conjunto más amplio que contenga dichas soluciones. El tipo de ordenación dependerá de la codificación considerada. Para poder aplicar este tipo de estrategia el espacio debe ser finito y no muy grande para que puedan recorrerse todos los entornos de las soluciones visitadas.

La búsqueda local parcial examina sólo una parte del espacio de búsqueda. Para llevar a cabo esto hay que establecer las pautas para organizar la selección de las soluciones a examinar. En caso de que la selección sea aleatoria tendremos la búsqueda por entornos aleatoria y si la selección es aleatoria pura o uniforme, se llama método de Monte Carlo. En este tipo de estrategias el uso de técnicas que eviten la repetición de soluciones ya examinadas aumentará la eficiencia de la búsqueda. Si la selección es sistemática, la búsqueda será llevada de forma exhaustiva parando en un determinado momento sin que haya sido examinado todo el espacio. El número de soluciones visitadas de cada entorno determina la intensidad de la búsqueda.

### Búsquedas monótonas

Son búsquedas informadas que hacen uso, de forma explícita o implícita, de la evaluación de la función objetivo para guiar la búsqueda. Se caracterizan porque analizan el entorno de la solución actual y sólo aplican movimientos que mejoren dicha solución [1, 65, 115].

La monotonicidad puede ser o no estricta. El segundo caso tiene como ventaja que puede escaparse de zonas llanas del espacio de búsqueda pero a su vez tiene el inconveniente de que puede ciclarse indefinidamente dentro de una de tales mesetas.

En esta categoría podemos destacar la búsqueda local aleatoria, la búsqueda local sistemática y la búsqueda local exhaustiva. La búsqueda local alea-

toria consiste en seleccionar iterativamente una solución al azar del entorno de la solución actual y se sustituirá por la solución encontrada si cumple el criterio de monotonicidad.

La búsqueda local sistemática persigue mejorar el poder de exploración del entorno de una solución. Para ello evitan que las soluciones candidatas se repitan durante el proceso de búsqueda.

Por último, la búsqueda local exhaustiva maximiza el poder de explotación del entorno de una solución. La Búsqueda Voraz y la Búsqueda Ansiosa son las más representativas de esta subcategoría. La Búsqueda Voraz tiene como regla de selección el mejor primero, que consiste en recorrer de forma iterativa todas las soluciones del entorno seleccionando la mejor de ellas. En caso de mejora, o empate si la monotonicidad es no estricta, se sustituye la solución actual por esta. En la Búsqueda Ansiosa la regla de selección es el primero mejor. Dicha regla detiene la búsqueda al encontrar una solución que cumpla el criterio de monotonicidad. En caso de que no encuentre ninguna, el entorno será examinado de forma exhaustiva.

La solución de partida es de gran importancia en este tipo de estrategias para poder llevar a cabo una búsqueda satisfactoria. Una opción sencilla es seleccionarla al azar, aunque en algunos casos se preferiere seleccionar una solución prometedora basándose en algún criterio.

## 2.5 Búsquedas globales

Uno de los principales inconvenientes de las búsquedas locales es el estancamiento en un óptimo local del entorno [1, 148]. Para evitar esto se emplean técnicas que amplían el espacio de exploración durante la búsqueda.

A continuación pasamos a describir este tipo de metaheurísticas agrupándolas, en función del número de soluciones que maneja la estrategia, en búsquedas basadas en recorrido y búsquedas basadas en población tal y como vimos en la sección 2.2.

### 2.5.1 Búsquedas basadas en recorrido

Las principales búsquedas globales que se enmarcan en esta categoría evitan el estancamiento en un óptimo local volviendo a comenzar la búsqueda desde otra solución distinta, modificando la estructura de entorno o permitiendo movimientos de empeoramiento. Estas opciones corresponden a la Búsqueda Multiarranque, Búsqueda por Entorno Variable y búsquedas no monótonas, respectivamente.

#### GRASP

Es una metaheurística constructiva [40] que añade elementos a una estructura, inicialmente vacía, hasta obtener una solución. La elección de los elementos a incorporar se basa en una evaluación heurística. Dicho método heurístico es dependiente del problema y expresa el conocimiento que el experto tiene relativo al problema que considera. Si la evaluación de los elementos depende de los elementos previamente seleccionados, se dice que el método es adaptativo. El método que suele considerarse para seleccionar el elemento a incorporar es una Búsqueda Voraz.

Tal y como podemos ver en la Figura 2.2 esta metaheurística consta de varias etapas. Primero una fase constructiva en la que se seleccionan de forma iterativa y aleatoria los elementos a incorporar de una lista de candidatos. A continuación le sigue una etapa de postprocesamiento en la que suele emplearse una búsqueda local que mejore la solución obtenida anteriormente. Esto se repite hasta que se alcanza el criterio de parada.

#### Búsqueda Multiarranque

Esta estrategia aplica búsquedas locales partiendo de soluciones iniciales distintas. El primer método descrito en la literatura [12, 98] consistía en repetir los procesos de generar la solución inicial y la búsqueda local. Dicho esquema puede generalizarse considerando, en vez de una búsqueda local, un método de búsqueda genérico. En la Figura 2.3 puede verse el esquema general de esta estrategia.

```
1:  Procedimiento GRASP
2:  {
3:  x* ← ∅ / f(x*) ← ∞;
4:  repetir
5:     x ←FaseConstructiva(S);
6:     x ←PostProcesamiento(x);
7:     si (f(x) < f(x*))
8:        x* ← x;
9:  hasta (CriterioDeParada)
10: }
```

Figura 2.2: Pseudocódigo de la Búsqueda GRASP.

Básicamente puede hablarse de dos fases en cada iteración $i$. En la primera se construye una solución $x_i$ y en la segunda se busca mejorar dicha solución aplicándole un método de búsqueda. Como resultado de esta segunda fase obtendremos una nueva solución con la que procedemos a actualizar la mejor solución $x^*$ del método en caso necesario.

Este método ha sido aplicado principalmente a la resolución de dos tipos de problemas de optimización: resolución de problemas no lineales y combinatorios. El primero de los casos se ha enfocado en el estudio de las condiciones de convergencia al óptimo global, mientras que en el segundo caso el estudio se ha orientado a la aplicación de heurísticas que encuentren buenas soluciones en un tiempo de computación relativamente moderado. Esto se debe a que en estos problemas los métodos no presentan propiedades de convergencia al óptimo global.

### Búsqueda por Entorno Variable

Es una metaheurística reciente [67] que está basada en un cambio sistemático de la estructura de entorno de la búsqueda [68, 69]. Esta estrategia está basada en tres hechos simples:

- Un mínimo local con una estructura de entorno no lo es necesariamente

```
1:  Procedimiento Búsqueda Multiarranque
2:  {
3:  i ← 1;
4:  x* ← ∅ / f(x*) ← ∞;
5:  repetir
6:     x ← GeneraSolucion(S);
7:     x ← BusquedaLocal(x);
8:     si (f(x) < f(x*))
9:        x* ← x;
10    i ← i + 1;
11: hasta (CriterioDeParada)
12: }
```

Figura 2.3: Pseudocódigo de la Búsqueda Multiarranque.

con otra.

- Un mínimo global es mínimo local con todas las posibles estructuras de entorno.

- En varios problemas, los mínimos locales con la misma o distinta estructura de entorno están relativamente cerca.

Esta última observación es empírica e implica que los óptimos locales proporcionan información acerca del óptimo global. Por regla general no se conoce esta información.

El cambio de estructuras de entorno ($\mathcal{N}_k,\ k = 1, \ldots, k_{max}$) puede hacerse de forma determinística, estocástica o combinando ambas. Frecuentemente los entornos sucesivos están anidados, de modo que si consideramos $\mathcal{N}_k(x)$ el conjunto de soluciones de $k$-ésimo entorno de $x$, podremos obtener el siguiente entorno haciendo $\mathcal{N}_{k+1}(x) = \mathcal{N}(\mathcal{N}_k(x))$, donde $\mathcal{N}_1(x) = \mathcal{N}(x)$ es el primer entorno de la solución.

El esquema básico de la Búsqueda por Entorno Variable lo podemos ver en la Figura 2.4. Comienza seleccionando el conjunto de estructuras de entorno

```
1:  Procedimiento Búsqueda por Entorno Variable
2:  {
3:  Inicializa N_k,  k = 1, ..., k_max;
4:  x* ← GeneraSolucionInicial(S);
5:  repetir
6:     k ← 1;
7:     repetir
8:        x ← AgitaSolucion(x*) / x ∈ N_k(x*);
9:        x ← BusquedaLocal(x);
10:       si (f(x) < f(x*))
11:          x* ← x;
12:          k ← 1;
13:       si no
14:          k ← k + 1;
15:    hasta (k = k_max)
16: hasta (CriterioDeParada)
17: }
```

Figura 2.4: Pseudocódigo de la Búsqueda por Entorno Variable.

que serán usadas durante la búsqueda. El método de agitación (*AgitaSolucion*) genera al azar una solución del $k$-ésimo entorno de la solución $x$. El objetivo es evitar que la búsqueda sea determinista.

A partir de este esquema básico han surgido diversas extensiones para dotar la búsqueda de características adicionales.

### 2.5.2 Búsquedas no monótonas

Mediante la aceptación de movimientos que no sean de mejora se pretende que se pueda evitar el estancamiento en un óptimo local. En esta categoría se engloban las búsquedas probabilísticas y las búsquedas con memoria. Las primeras asocian una probabilidad de aceptación a la nueva solución encontrada, mientras que las segundas hacen uso del historial de búsqueda para

evitar que se repitan soluciones.

La búsqueda probabilística selecciona aleatoriamente un vecino de la solución actual y ésta la reemplaza atendiendo a un criterio probabilístico. Por regla general, a mejor solución, mayor será la probabilidad de aceptación. El Recocido Simulado [82] es la metaheurística más importante de esta categoría. La función de aceptación es exponencial y la probabilidad de aceptación de soluciones peores va decreciendo conforme avanza la búsqueda. El fundamento de este control se basa en el trabajo de Metrópolis et al. [100] en el campo de la termodinámica estadística. En su trabajo modeló los cambios energéticos en el sistema de partículas conforme decrece la temperatura en el proceso de recocido. Dichos cambios energéticos se producen hasta que converge a un estado estable.

En dicho modelo si hay una caída de la energía, el cambio se acepta mientras que en caso contrario habrá una probabilidad de aceptación en función de la temperatura. En cada iteración la temperatura irá bajando; con lo que la probabilidad de aceptación irá disminuyendo. En la Figura 2.5 podemos ver el esquema básico de esta metaheurística.

La Búsqueda Tabú es la metaheurística más extendida de las búsquedas con memoria. El uso de memoria pretende mejorar el rendimiento de la búsqueda. Está fundamentada en las ideas de F. Glover [57]. En su origen, el propósito era evitar la reiteración de búsqueda en zonas ya exploradas, aunque posteriormente se han propuesto métodos para rentabilizar la memoria a medio y largo plazo. La estrategia Búsqueda Tabú evita la repetición prematura de las soluciones creando una lista tabú $T$ de soluciones temporalmente prohibidas. En ciertos casos se introduce un criterio de aspiraciones $A$ con objeto de que algunos candidatos de dicha lista tabú puedan ignorar su estatus de tabú.

### Búsqueda Local Iterada

La Búsqueda Local Iterada [136] propone una estrategia exploratoria de mejora de las soluciones obtenidas por una determinada heurística mediante la repetición de dicha heurística. Esta idea ha sido propuesta en la literatura

1: **Procedimiento** *Recocido Simulado*
2: {
3: $T \leftarrow T_0$;
4: $x \leftarrow SeleccionaSolucion(S)$;
5: $x^* \leftarrow x$;
6: **repetir**
7: $x' \leftarrow SeleccionaSolucion(x) \ / \ x' \in \mathcal{N}(x)$;
8: **si** $(f(x') < f(x))$
9: $x \leftarrow x'$;
10: **si** $(f(x) < f(x^*))$
11: $x^* \leftarrow x$;
12: **si no**
13: $x \leftarrow CriterioDeAceptacion(p(T, x', x)) \ / \ p(T, x', x) = e^{\frac{f(x')-f(x)}{T}}$;
14: $T \leftarrow Actualiza(T)$;
15: **hasta** (*CriterioDeParada*)
16: }

Figura 2.5: Pseudocódigo de la búsqueda probabilística Recocido Simulado.

con distintas denominaciones como Descenso Iterado, Lin-Kerningan Iterado, Búsqueda Perturbada, etc.

El método considera una heurística base para obtener una solución que es perturbada, repitiéndose este proceso hasta que se cumple el criterio de parada. El proceso se convierte en una búsqueda estocástica por entornos donde dichos entornos vienen determinados por la heurística base en vez de ser definidos de forma explícita. En la Figura 2.7 podemos ver el pseudocódigo. La incorporación de memoria ($\mathcal{H}$) mejora el rendimiento de la metaheurística.

### Búsqueda Local Guiada

Esta metaheurística [140, 141] basa su búsqueda en la iteración de un procedimiento de búsqueda y en la modificación de la función objetivo cada vez que termina la búsqueda local. Para ello generan nuevas restricciones que

1: **Procedimiento** *Búsqueda Tabú*
2: {
3: $k \leftarrow 1$;
4: $x^* \leftarrow SeleccionaSolucion(S)$;
5: **repetir**
6: $x \leftarrow SeleccionaSolucion(x^*) \ / \ x \in \mathcal{N}(x^*, k) = \mathcal{N}(x^*) - T(x^*, k) + A(x^*, k)$;
7: **si** $(f(x) < f(x^*))$
8: $x^* \leftarrow x$;
9: $k \leftarrow k + 1$;
10: **hasta** $(CriterioDeParada)$
11: }

Figura 2.6: Pseudocódigo de la Búsqueda Tabú.

definen mejor el problema haciendo uso de información que se va obteniendo a lo largo de la búsqueda y que se va incorporando en la función objetivo.

### 2.5.3 Búsquedas basadas en población

En este grupo se engloban las estrategias que llevan a cabo la búsqueda mediante una población de soluciones.

#### Algoritmo Genético

Las estrategias de búsqueda basadas en población se iniciaron con el Algoritmo Genético de Holland [71]. Desde entonces esta estrategia ha alcanzado una gran popularidad tal y como puede verse en la extensa bibliografía sobre su aplicación a diversos problemas de optimización [24, 64, 101, 104].

La principal característica es que es un algoritmo iterativo inspirado en la teoría de Darwin de la evolución en que cada iteración $t$ es denominada generación, y se tiene una población de cromosomas (soluciones) a partir de la cual se selecciona un subconjunto con la que se genera una nueva población. Esta nueva población la formarán cromosomas pertenecientes a

1: **Procedimiento** *Búsqueda Local Iterada*
2: {
3: *Inicializa* $\mathcal{N}_k,\ k = 1, \ldots, k_{max}$;
4: $x_0 \leftarrow$ *GeneraSolucionInicial*($S$);
5: $x^* \leftarrow$ *BusquedaLocal*($x_0$);
6: **repetir**
7: $x' \leftarrow$ *PerturbaSolucion*($x^*$,$\mathcal{H}$);
8: $x'' \leftarrow$ *BusquedaLocal*($x'$);
9: $x^* \leftarrow$ *CriterioDeAceptacion*($x$,$x''$,$\mathcal{H}$);
10: **hasta** (*CriterioDeParada*)
11: }

Figura 2.7: Pseudocódigo de la Búsqueda Local Iterada.

la población actual así como a la prole generada en función de la calidad de estos cromosomas.

Existen diversas propuestas de codificación de los cromosomas dependiendo principalmente del problema al que se aplica. Una de las codificaciones más extendidas es la binaria, en la que cada gen (elemento) de un cromosoma puede tomar los valores 0 ó 1.

La evolución de la población se basa en operadores inspirados en la evolución natural de los seres vivos. Los dos más extendidos son el cruce de genes de dos cromosomas y la mutación de los genes. La Figura 2.8 muestra el esquema general del Algoritmo Genético Simple.

### Búsqueda Dispersa

Este procedimiento metaheurístico está basado en formulaciones matemáticas y estrategias introducidas en la década de los sesenta. La primera descripción del método fue publicada en 1977 por Fred Glover [56] estableciendo sus principios. Desde entonces, esta metaheurística ha sido aplicada a diversos problemas [60, 63].

La Búsqueda Dispersa opera sobre un conjunto de soluciones, llamado

```
1:  Procedimiento Algoritmo Genético Simple
2:  {
3:  t ← 0;
4:  P_t ←GeneraPoblacion(S);
5:  P_t ←EvaluaPoblacion(P_t);
6:  repetir
7:     t ← t + 1;
8:     P_t^1 ←SeleccionaProgenitores(P_t);
9:     P_t^2 ←CruzaProgenitores(P_t^1);
10:    P_t^3 ←MutaProle(P_t^2);
11:    P_t ←ActualizaPoblacion(P_t^3);
12:    P_t ←EvaluaPoblacion(P_t);
13: hasta (CriterioDeParada)
14: }
```

Figura 2.8: Pseudocódigo del Algoritmo Genético Simple.

conjunto de referencia ($ConjRef$), que se combinan entre sí para crear nuevas soluciones que mejoren a las que la originaron. Debido a esto se enmarca dentro de los algoritmos evolutivos. Este método se basa en elecciones sistemáticas y estratégicas sobre un conjunto pequeño de soluciones.

En la Figura 2.9 podemos ver el pseudocódigo de este método. La estrategia genera inicialmente un conjunto de soluciones diversas del que se selecciona un subconjunto de éstas llamado conjunto de referencia. La selección de este conjunto se basa en la calidad y la diversidad de las soluciones presentes en este conjunto inicial. El primer criterio permite seleccionar las soluciones con mayor calidad mientras que el segundo favorece seleccionar aquellas con características más diversas a las soluciones presentes en el conjunto de referencia. Posteriormente se seleccionan subconjuntos de soluciones ($SubConj$) a partir de los cuales se generan nuevas soluciones mediante técnicas de combinación de soluciones. A continuación se pasa, a cada nueva solución generada, un método de mejora que suele ser una búsqueda local. Finalmente se actualiza $ConjRef$ teniendo en cuenta las soluciones ya exis-

tentes en $ConjRef$ y las nuevas generadas.

---

1: **Procedimiento** *Búsqueda Dispersa*
2: {
3: $C \leftarrow GeneraConjunto(\mathrm{S})$;
4: $ConjRef \leftarrow GeneraConjuntoDeReferencia(C)$;
5: **repetir**
6: $ConjRef' \leftarrow \emptyset$;
7: **repetir**
8: $SubConj \leftarrow SeleccionaSoluciones(ConjRef)$;
9: $SubConj \leftarrow CombinaSoluciones(SubConj)$;
10: $SubConj \leftarrow MejoraSoluciones(SubConj)$;
11: $ConjRef' \leftarrow ConjRef' \cup SubConj$;
12: **hasta** $(CriterioDeParada_1)$
13: $ConjRef \leftarrow ActualizaConjuntoDeReferencia(ConjRef, ConjRef')$;
14: **hasta** $(CriterioDeParada_2)$
15: }

---

Figura 2.9: Pseudocódigo de la Búsqueda Dispersa.

### Algoritmo de Estimación de Distribuciones

Este método fue introducido en el ámbito de la computación evolutiva por Mühlenbein y Paa$\beta$ [110]. La peculiaridad de esta metaheurística es que la nueva población se genera a partir de una simulación de una distribución de probabilidad. Dicha distribución de probabilidad se estima a partir de las soluciones de la iteración anterior. La estimación de la distribución de probabilidad se hace en función de modelos probabilísticos. En [91] se puede consultar un trabajo sobre su aplicación a diversos problemas de optimización.

Una característica de este método es que las interrelaciones entre las variables se expresan de forma explícita a través de la distribución de probabilidad conjunta asociada a los individuos seleccionados en cada iteración. El modelo

probabilístico será el encargado de aprender estas interdependencias y es el mecanismo de evolución.

El pseudocódigo de la Figura 2.10 muestra los pasos que tiene esta metaheurística. Este algoritmo comienza generando aleatoriamente un conjunto de $M$ soluciones $D_0$. En cada iteración $l$, $N$ soluciones son seleccionadas $D_{l-1}^{Se}$ para aprender un modelo de distribución de probabilidad $p_l$ a partir de la cual se generan $M$ soluciones mediante simulación del modelo aprendido. Finalmente se actualiza la nueva población $D_l$ con las soluciones de la población anterior y las nuevas generadas. Dos de los criterios de parada más extendidos consisten en detener la búsqueda cuando se alcanza un cierto número de iteraciones o cuando las soluciones convergen.

---

1: **Procedimiento** *Algoritmo de Estimación de Distribuciones*
2: {
3: $l \leftarrow 0$;
4: $D_0 \leftarrow$*GeneraSoluciones*(M);
5: **repetir**
6: $l \leftarrow l + 1$;
7: $D_{l-1}^{Se} \leftarrow$ *SeleccionaSoluciones*($N$,$D_{l-1}$) / $N \leq M$;
8: $p_l(x) = p(x|D_{l-1}^{Se}) \leftarrow$ *EstimaDistribucionDeProbabilidad*($D_{l-1}^{Se}$);
9: $D_l \leftarrow$ *MuestreaSoluciones*($M$,$p_l(x)$);
10: **hasta** (*CriterioDeParada*)
11: }

---

Figura 2.10: Pseudocódigo del Algoritmo de Estimación de Distribuciones.

### Colonia de Hormigas

Es un algoritmo de búsqueda propuesto por Marco Dorigo et al. [31, 32, 33] inspirado en el comportamiento de las hormigas, y más concretamente en la forma en la que intercambian información durante el proceso de búsqueda de alimentos. Este comportamiento permite que las hormigas encuentren el camino más corto entre el alimento y el nido. La información se intercambia

en forma de rastro de feromonas que van depositando a lo largo de su trayectoria. La dirección que toman viene influenciada por la concentración de esta sustancia, de modo que a mayor concentración, mayor probabilidad de que tomen ese trayecto.

Este algoritmo simula el intercambio de información entre las hormigas, en su búsqueda de alimentos, construyendo una matriz de feromonas $F$ cuya información se irá modificando a lo largo de la búsqueda. Se considera que las búsquedas de las soluciones $S$ son llevadas a cabo por las hormigas $H$ en analogía a la búsqueda de alimentos.

El funcionamiento de este algoritmo se basa en tres fases que son ejecutadas de forma iterativa hasta cumplir las condiciones del criterio de parada. En la primera de ellas las hormigas construyen las soluciones de forma incremental aplicando decisiones locales estocásticas. En la segunda fase, se actualiza la matriz de feromonas. Este procedimiento puede hacerse a medida que se van construyendo las soluciones o bien una vez que se hayan terminado de construir todas ellas. El último método es opcional y se utiliza para implementar acciones que no pueden ser tomadas por las hormigas de forma individual. En la Figura 2.11 podemos ver su pseudocódigo.

```
1:  Procedimiento Colonia de Hormigas
2:  {
3:    H ←DefineHormigas();
4:    F ← F0;
5:  repetir
6:    S ←GeneraSoluciones(H,F);
7:    F ←ActualizaFeromonas(S,F);
8:    RealizaOtrasAcciones();
9:  hasta (CriterioDeParada)
10: }
```

Figura 2.11: Pseudocódigo de la Colonia de Hormigas.

# Capítulo 3

# Problema de selección de atributos

## 3.1 Introducción

Los conjuntos de datos con los que suelen tratarse para llevar a cabo tareas de clasificación han experimentado últimamente un crecimiento exponencial en el número de variables o atributos. Se ha pasado de tratar con unas pocas decenas de variables a tener en la actualidad cientos e incluso miles de atributos. Sin embargo, la información útil contenida en dichos atributos para el proceso de clasificación no suele ser homogénea. La información de algunos atributos puede ser irrelevante para la tarea o bien redundante debido a la correlación entre variables. La selección de atributos persigue encontrar un subconjunto reducido de atributos con el que llevar a cabo la tarea.

Algunos de los beneficios con los que podemos encontrarnos son, tal y como apunta Reunanen [125], los siguientes:

- Disminución en el consumo de recursos al medir subconjuntos de variables de menor tamaño.
- Posible aumento de la precisión del clasificador.

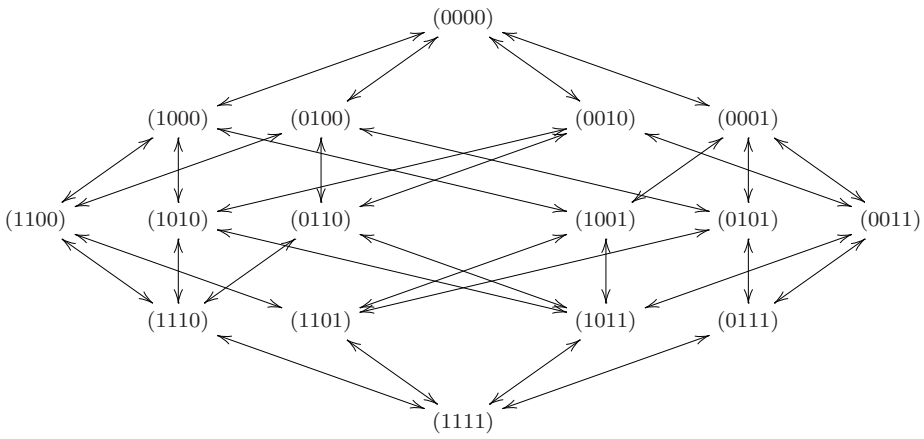


Figura 3.1: Espacio de atributos de 4 dimensiones.

- Aumento de la eficiencia en la construcción del clasificador.
- Posible mejor comprensión de la naturaleza de los datos a tratar.

El proceso de la selección de las variables en las que basar la clasificación puede considerarse como un problema de búsqueda [133] en un espacio de estados, en el que cada estado puede representarse como un vector de tamaño igual al número de variables presentes en el problema y cada elemento del vector puede tomar el valor 1 si la variable correspondiente se selecciona y 0 en caso contrario. El tamaño del espacio de búsqueda de subconjuntos de un conjunto de $d$ variables es $O(2^d)$. La Figura 3.1 representa los 16 estados para el caso de $d = 4$ conectados por los movimientos de incluir o excluir una de las variables.

## 3.2 Relevancia

Antes de poder definir el proceso de selección de atributos, conviene profundizar en los tipos de variables con las que trataremos en función de su capacidad predictiva. El hecho de saber diferenciar entre estos tipos de varia-

bles nos permite una mejor comprensión del rendimiento de una estrategia de selección de atributos.

El concepto de relevancia, a pesar de los esfuerzos realizados por distintos autores, no tiene una definición formal que sea aceptada por todos [9]. Una de las causas de esto, tal y como señalan Blum y Langley [11], es que el concepto de relevancia se asocia a una tarea en concreto dentro del aprendizaje automático.

Una propuesta de relevancia es dada por Michalski [102], que distingue tres situaciones distintas en cuanto a la relevancia de los atributos utilizados en el aprendizaje automático:

**Relevancia Completa.** Ésta se presenta si los atributos son directamente relevantes al problema. En este caso, todos los atributos intervendrían en la tarea de clasificación.

**Relevancia Parcial.** Esta situación se da cuando los objetos tienen atributos irrelevantes o redundantes. Por tanto, para llevar a cabo la tarea de aprendizaje, habrá que seleccionar el conjunto de atributos más relevantes.

**Relevancia Indirecta.** Esta situación se presenta si existen atributos que no son relevantes, pero los atributos nuevos generados a partir de éstos sí lo son.

Un problema que ofrece la distinción anterior es que en la definición de cada una de las categorías aparece el término relevancia referido a atributos individuales. Existen alternativas más formales para introducir el concepto de relevancia como las de Genari et al. [55].

Teniendo en cuenta que sólo disponemos de una muestra $\mathcal{E}$ de ejemplos, frecuentemente se postula una distribución de probabilidad $D_{\mathcal{E}}$ sobre el espacio de las instancias y que existe una función de correspondencia entre los ejemplos y la clase. De esta forma, tal y como señalan Blum y Langley [11], la noción más simple de relevancia está ligada a un objeto, que en nuestro caso es la clase.

**Definición 3.1** ***Relevancia con respeto a la clase***

*Un atributo $X_i$ se dice que es relevante al concepto $\mathcal{C}$ si existen un par de ejemplos $e_j$ y $e_k$ en el espacio de instancias tales que $e_j$ y $e_k$ difieren sólo en el valor de $X_i$ y en la clase $c_j \neq c_k$.*

De esta forma, un atributo $X_i$ se dice relevante si existe un ejemplo tal que modificando el valor del atributo $X_i$, afecta a su clasificación. El principal inconveniente de esta definición radica en que sólo se puede acceder a una muestra del espacio de instancias, de modo que no se puede saber si un atributo es realmente relevante o no. Además, en caso de que la codificación sea redundante, puede que no sea posible que dos ejemplos difieran en el valor de un solo atributo. Estos inconvenientes son solventados por John et al. [78], quienes proponen distinguir entre relevancia fuerte, débil e irrelevancia.

**Definición 3.2** ***Relevancia fuerte con respeto a una muestra***

*Un atributo $X_j$ se dice que es fuertemente relevante a la muestra $\mathcal{E}$ si existen ejemplos $e_i$ y $e_k \in \mathcal{E}$ que difieren en el valor de $X_j$ y pertenecen a clases distintas ($c_i \neq c_k$).*

Esta definición difiere de la anterior en que se requiere que los ejemplos pertenezcan a la muestra.

**Definición 3.3** ***Relevancia débil con respeto a una muestra***

*Un atributo $X_j$ se dice que es débilmente relevante a la muestra $\mathcal{E}$ si se puede eliminar un subconjunto de atributos de modo que $X_j$ sea fuertemente relevante.*

Aquellos atributos que no encajen en las dos definiciones anteriores se dirá que son irrelevantes. De esta forma se considera que los atributos fuertemente relevantes son siempre importantes para clasificar mientras que los débilmente relevantes serán seleccionados dependiendo del subconjunto de atributos ignorados.

### Relevancia y optimalidad

El criterio para seleccionar el subconjunto de atributos para inducir un buen clasificador es que éste sea un subconjunto óptimo de atributos. Kohavi y John [84] introducen la siguiente definición de subconjunto óptimo de atributos.

**Definición 3.4** ***Subconjunto de atributos óptimo***

*Dado un inductor $\mathcal{I}$, y una muestra de instancias $\mathcal{E}$ descritas mediante el conjunto de atributos $X$, un subconjunto de atributos, $S \subseteq X$, es un subconjunto de atributos óptimos si la precisión del clasificador inducido $\mathcal{L} = \mathcal{I}(\mathcal{E})$ es máxima.*

Sin embargo, la elección del subconjunto de atributos relevantes es generalmente subóptima [66] de cara a la inducción del clasificador. Una de las causas es que las anteriores definiciones de relevancia son independientes del algoritmo de aprendizaje considerado, de modo que no hay garantía de que un atributo relevante sea necesariamente útil al algoritmo. Caruana y Freitag [19] introducen el término de Utilidad Incremental, el cual está ligado al algoritmo de aprendizaje.

**Definición 3.5** ***Utilidad Incremental***

*Dada una muestra $\mathcal{E}$, un algoritmo de aprendizaje $\mathcal{L}$, un conjunto de atributos $S$, un atributo $X_j$ se dice que es incrementalmente útil a $\mathcal{L}$ con respecto a $S$ si la tasa de acierto de la hipótesis que $\mathcal{L}$ produce usando el conjunto de atributos $\{X_j\} \cup S$ es mejor que la tasa de acierto obtenida utilizando sólo el conjunto $S$.*

Esta definición se ajusta de forma natural a la selección de atributos como un proceso de búsqueda.

## 3.3 Metodologías de evaluación

Existen diversas taxonomías a lo largo de la literatura en función de diversos criterios. Algunos de estos criterios son la interrelación del proceso de selección de variables con el algoritmo inductivo y el tipo de medida usada.

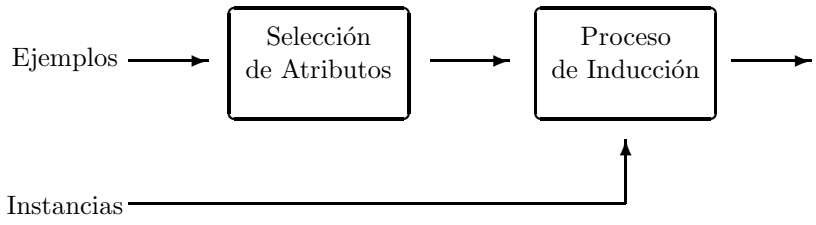


Figura 3.2: Evaluación filtro. El procedimiento de búsqueda es independiente del algoritmo de inducción.

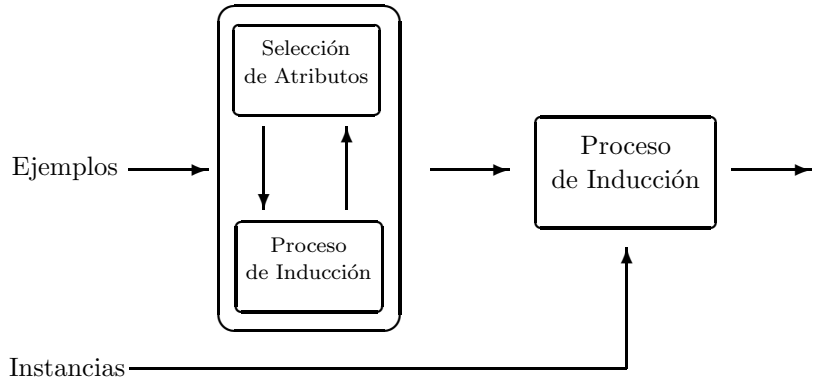


Figura 3.3: Evaluación envolvente. El algoritmo de inducción se usa para guiar la búsqueda.

Langley [88] introduce dos categorías de evaluación distintas en función de la interrelación del proceso de la selección de variables con el algoritmo inductivo: la metodología filtro y la metodología envolvente. Los métodos de la primera categoría (Figura 3.2) consideran la selección de atributos como una etapa previa e independiente al propio algoritmo inductivo, de modo que puede considerarse una etapa de filtrado de atributos. Los métodos envolventes (Figura 3.3), sin embargo, hacen uso del propio algoritmo inductivo para evaluar los subconjuntos de atributos encontrados durante el proceso de búsqueda.

Blum y Langley [11] introducen posteriormente la dependencia existente entre el proceso de selección de los atributos y el de inducción del clasificador como otro aspecto a tener en cuenta. En este esquema se tienen cuatro categorías: los métodos empotrados o incrustados, los métodos de tipo filtro, los métodos envolventes y los métodos de ponderación. Los métodos empotrados agrupan a los procesos inductivos que intrínsecamente llevan a cabo la selección de variables durante la inducción del clasificador. En este caso se encuentran los métodos de inducción que obtienen como clasificadores a los árboles de decisión [124]. Las dos categorías posteriores, los métodos de tipo filtro y envolventes, ya fueron comentados anteriormente. Por último, los métodos de ponderación son los que realizan la selección de los atributos en función de unos pesos asociados a los atributos. El ejemplo típico de los métodos de ponderación está constituido por los métodos de clasificación basados en Redes Neuronales.

Dash y Liu [22] proponen dividir las metodologías de evaluación en cinco categorías en función del tipo de medida que utilizan. De esta forma, se distingue entre los métodos que hacen uso de distancia, medida de información, dependencia estadística, consistencia y error del clasificador. La medida de distancia también es conocida como medida de discriminación o de similitud y la más extendida es la distancia euclídea. Los métodos basados en información utilizan medidas derivadas de la teoría de la información, y los basados en dependencia (o correlación) se basan en el estudio de la relación estadística existente entre los atributos y la clase. La medida de consistencia pretende encontrar el subconjunto mínimo de variables con las que es posible construir

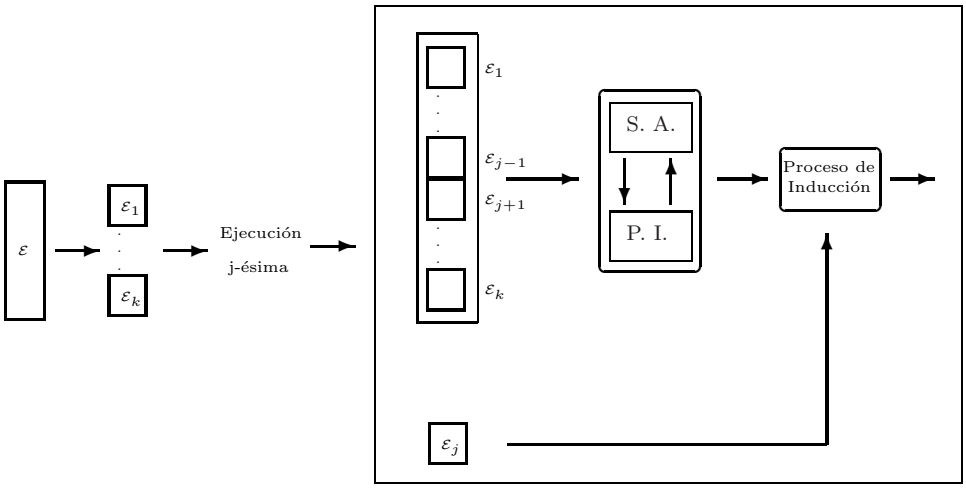


Figura 3.4: Esquema detallado de la i-ésima ejecución de la evaluación Envolvente.

una hipótesis consistente con el conjunto de entrenamiento y por último, los métodos basados en la tasa de error emplean el error del clasificador inducido como medida de calidad.

### Evaluación Envolvente

La evaluación envolvente fue introducida por Kohavi [84] y está basada en el error. El error es estimado considerando una validación cruzada en el propio conjunto de entrenamiento. Cada vez que se va a evaluar un subconjunto de atributos (una solución del problema de selección de atributos), el conjunto de entrenamiento $\mathcal{E}$ es dividido en $k$ particiones disjuntas y el subconjunto se evalúa $k$ veces en $\mathcal{E}$. En la i-ésima evaluación (Figura 3.4), la partición $\mathcal{E}_i$ se considera de validación y el resto de aprendizaje. Finalmente, el valor asociado al subconjunto de entrenamiento viene dado por el promedio de los $k$ resultados. En nuestro caso, debido al alto coste computacional de este método de evaluación, sólo consideramos $k = 2$.

## 3.4 Estado del arte

En la bibliografía puede encontrarse una gran variedad de trabajos relacionados con el problema de la selección de variables. Liu y Yu [97] hacen una revisión en función del método de evaluación así como del tipo de búsqueda considerado. Los métodos de evaluación que diferencian coinciden con los de Dash y Liu [22] tal y como se indicó en el apartado 3.3 (basados en distancia, información, dependencia, consistencia y error). Respecto al tipo de búsqueda se distingue entre algoritmos completos y heurísticos. Los algoritmos completos son los que llegan a la solución óptima aunque, tal y como señalan Narendra y Fukunaga [111], en algunos casos completo no significa una búsqueda exhaustiva debido a que puede existir alguna propiedad, como la de monotonía en la función de evaluación, que haga innecesario el evaluar algunas soluciones. Los algoritmos heurísticos no buscan en todo el espacio de soluciones, distinguiéndose entre búsquedas deterministas y no deterministas. Las primeras de ellas siempre llegan a la misma solución mientras que las últimas no.

A continuación pasamos a mostrar los trabajos más representativos en función del método de evaluación y del tipo de estrategia de búsqueda, diferenciando, para un tipo de evaluación determinada, entre algoritmos completos y heurísticos, y dentro de los heurísticos entre deterministas y no deterministas.

### 3.4.1 Distancia

Cada ejemplo es considerado como un punto en el espacio $d$-dimensional y parte del supuesto de que cada clase pertenece a una región diferenciada del espacio. El conjunto de atributos a elegir será aquel en el que la separación entre dos regiones sea máxima y la separación entre los casos de la misma clase sea mínima. Para atributos continuos la separación de la muestra según su clase puede realizarse mediante un hiperplano en el espacio de atributos. Sin embargo, en la mayoría de los casos, la separación no es lineal con lo que dicha aproximación no es apropiada.

### Completo

El método de Ramificación y Acotación (*Branch&Bound* o B&B) [111] es de los más antiguos, el cual parte del conjunto de todos los atributos y durante el proceso va eliminando atributos hasta que la evaluación del subconjunto esté por debajo de un valor umbral. Este método requiere que el método de evaluación tenga la propiedad de monotonicidad.

Una estrategia similar es la BFF (*Best First Feature*) [147], que propone una estrategia de búsqueda para solucionar el problema de encontrar un camino óptimo en un árbol ponderado mediante un primero mejor. En caso de que el criterio tenga la propiedad de monotonicidad, el algoritmo garantiza encontrar el subconjunto óptimo de atributos sin una búsqueda exhaustiva.

### Heurístico (no determinista)

Una de las estrategias de mayor impacto que se engloban en esta categoría es el algoritmo *Relief* [81], que se basa en asignar un peso a cada atributo y finalmente seleccionar aquellos que superen un determinado valor umbral prefijado. Para estimar el peso de los atributos se selecciona aleatoriamente un subconjunto de ejemplos y se busca, para cada uno, el vecino más cercano de su misma clase y de la clase contraria. Finalmente el peso se estima considerando, para cada ejemplo, la distancia euclídea entre el ejemplo y sus vecinos. Una versión mejorada es el *ReliefF* [86], el cual permite ser aplicado a variables no binarias y tratar con valores perdidos.

Segen [131] propone un método de evaluación que minimiza la suma de una medida estadística de discrepancia y complejidad del atributo. De este modo elige el atributo que mejor distingue las clases e iterativamente va escogiendo aquella variable que en combinación con las anteriores mejora la información de la clase.

El algoritmo EUBAFES [128] (*EUclidean BAsed Feature Selection*) se basa en la distancia euclídea para asignar pesos a los atributos. Es similar a *Relief* pero hace uso de una función distinta para asociar los pesos.

### 3.4.2 Información

Las evaluaciones basadas en teoría de la información han sido utilizadas ampliamente en la selección de atributos. La información contenida en los atributos es tratada como magnitud física y dicha información se caracteriza mediante la entropía. Dicho concepto es una medida de incertidumbre promedio de los valores que puede tomar un atributo.

#### Completo

El algoritmo MDLM (*Minimun Description Length Method*) [132] trata de eliminar aquellos atributos que son innecesarios (irrelevantes y/o redundantes). Si los atributos en un subconjunto $V$ pueden ser expresados mediante una función $F$ de los atributos de otro subconjunto de atributos $U$ de modo que $F$ sea independiente de la clase, entonces los atributos de $V$ pasan a ser innecesarios. Para ello considera el criterio MDLC (*Minimun Description Length Criterion*) [127] que se basa en la correspondencia entre regularidad y compresión (en este caso reducción de variables). Este método lleva a cabo la búsqueda de forma exhaustiva analizando los $2^d$ posibles subconjuntos para encontrar el que satisfaga dicho criterio. Este algoritmo es útil para distribuciones Gaussianas de un atributo.

#### Heurístico (determinista)

En esta categoría encontramos el algoritmo SFG (*Sequential Forward Generation*) [17] que comienza con un subconjunto vacío de variables y en cada iteración añade aquella con la que el subconjunto obtenga mejor evaluación en función del criterio de información.

El método de árbol de decisión DTM (*Decision Tree Method*) [17] adopta una idea similar al SFG para generar subconjuntos de atributos pero hace uso de medidas basadas en la entropía para evaluarlos. Sólo se seleccionarán aquellos atributos que aparezcan en el árbol de decisión.

El método DT-CBL (*Decision Tree for Case-Based Learning*) [16] es un algoritmo enfocado al aprendizaje mediante árboles de decisión que pondera

los atributos en función de la información mutua entre las variables utilizadas durante la inducción.

La estrategia de Koller y Sahami [85] se basa en que cualquier variable que sea redundante o irrelevante debería ser eliminada. Para ello utilizan una aproximación del concepto de recubrimiento de Markov. Un subconjunto $R$ de atributos es un recubrimiento de Markov $B$ de una variable $X_j$ si, dado $R$, $X_j$ es condicionalmente independiente del atributo clase $\mathcal{C}$ y de las variables no presentes en $R$ ($X_j$ incluida). Teóricamente se demuestra que si se encuentra un recubrimiento de Markov $B$ de un atributo $X_j$ en un conjunto $R$, se puede eliminar $X_j$ sin incrementar la divergencia con la distribución original.

Otro algoritmo que hace uso de este concepto es el FCBF (*Fast Correlation-Based Filter*) [150] , el cual hace selección de atributos mediante un análisis de relevancia y redundancia. La estrategia consiste en dos etapas; una primera en la que se selecciona un subconjunto de atributos relevantes y una segunda en la que se seleccionan, de entre los atributos relevantes, aquellos que no tengan ningún recubrimiento de Markov.

El algoritmo MIFS (*Mutual Information Feature Selector*) [7] calcula la información mutua de cada variable con la clase y entre cada par de atributos. Inicialmente se selecciona el atributo con mayor información mutua respecto a la clase y posteriormente se van añadiendo atributos de modo que la información aportada sobre la clase sea máxima sin ser redundante respecto a las anteriormente seleccionadas.

El método denominado CR (*Concept Relevance*) [142] hace uso de una medida de relevancia entre el conjunto de variables y la clase en base a conceptos de la teoría de la información para seleccionar un subconjunto de atributos.

### 3.4.3 Dependencia

Los métodos basados en la dependencia estadística, también denominados de medidas de correlación, tienen como objetivo predecir el valor de una variable en función del valor de otra. Una medida de dependencia clásica es el coeficiente de correlación, y puede ser usada para obtener la correlación

entre un atributo y la clase. A pesar de que las medidas basadas en dependencia pueden incluirse en la categoría de evaluación según la distancia o la información, los autores Liu y Yu [97] lo consideran en un apartado distinto debido a la diferencia del enfoque teórico en la realización de la búsqueda.

#### Heurístico (no determinista)

La estrategia POE+ACC (*Probability of Error & Average Correlation Coefficient*) [109] considera en primer lugar el atributo con menor probabilidad de error ($P_e$, POE). En cada iteración, el siguiente atributo a incluir será aquel cuya suma ponderada de $P_e$ y el coeficiente de correlación promedio (ACC) sea mínima. Como criterio de parada se requiere el número de atributos.

PRESET [106] es un algoritmo que considera los conjuntos rugosos (*rough sets*) para la búsqueda de subconjuntos de atributos.

### 3.4.4 Consistencia

Las medidas de consistencia se basan en que los valores de los atributos de dos casos deben ser distintos en caso de que pertenezcan a clases distintas. En caso de que existan, al menos, dos casos con los mismos valores de atributos, pero con distinta clase, se produce una inconsistencia. Los algoritmos de selección de variables se basan en la búsqueda del menor subconjunto de atributos que siga siendo consistente con el conjunto de aprendizaje o que dicha consistencia sea para el mayor número de casos posible.

#### Completo

FOCUS [3] es un algoritmo de gran impacto en el problema de la selección de variables que hace una búsqueda exhaustiva en anchura partiendo de un subconjunto inicial vacío para encontrar el mínimo subconjunto consistente para la clasificación. En problemas donde $d$ tenga un valor elevado, el tiempo de cálculo es prohibitivo.

Dos variantes de este algoritmo son la estrategia de Schimmer [129], que hace uso de una función heurística que hace que la búsqueda del subconjunto

de variables óptimo sea más eficiente, y el MIFES1 [114], que representa las instancias como una matriz. Un atributo se dice que cubre un elemento de la matriz si adopta valores opuestas para ejemplos de una clase positiva y otra negativa.

Una modificación del método de Ramificación y Acotación anteriormente expuesto es el denominado Ramificación y Acotación Automático (*Automatic Branch & Bound*, ABB) [93], donde la cota es determinada automáticamente por el propio algoritmo. Comienza con todo el conjunto de atributos y en cada iteración elimina un atributo mediante una búsqueda Primero en Profundidad hasta que la eliminación de un atributo implique que se deja de cumplir el criterio de consistencia.

### Heurístico (determinista)

SetCover [21] es un algoritmo que busca el mínimo subconjunto de variables que cubra pares de ejemplos de distinta clase con variables que varíen su valor. Se basa en la estrategia propuesta por Johnson [79].

El VCC [143] (*Vertical Compactness Criterion*) es un algoritmo que combina las heurísticas Primero en Profundidad y Primero en Anchura para buscar el mínimo subconjunto de variables que minimice la inconsistencia.

### Heurístico (no determinista)

En esta categoría el algoritmo más representativo es el algoritmo filtro de Las Vegas [95] (*Las Vegas Filter*, $LVF$), que hace uso de una estrategia aleatoria [13] para guiar la búsqueda. Este algoritmo es poco eficiente, por lo que no puede usarse en problemas con un gran número de variables. El algoritmo escalable de Las Vegas (*Scalable Las Vegas*, $SLV$) [96] es una variante que mejora la eficiencia reduciendo los casos sobre los que aplicar el criterio de inconsistencia.

El $QBB$ [23] (*Quick Branch & Bound*) es otra variante del $LVF$ que resuelve el problema de la lenta mejora que lleva a cabo a lo largo de las etapas. Para ello combina el $LVF$ con el $ABB$.

### 3.4.5 Error

El subconjunto a encontrar es aquel que tenga menor tasa de error en el aprendizaje. Sin embargo dicha medida es una estimación con lo que añade incertidumbre al proceso de búsqueda. Uno de los inconvenientes es que los resultados obtenidos restan generalidad por ser dependientes del clasificador considerado.

#### Completo

La estrategia AMB&B [44] (*Approximate Monotonic Branch & Bound*) es una estrategia de Ramificación y Acotación que permite un criterio de evaluación no monótona.

El BS [29] (*Beam Search*) es una estrategia Primero Mejor que considera una cola para limitar la profundidad de la búsqueda. En la cola se ordenan los subconjuntos de mejor a peor. El proceso de generación de subconjuntos consiste en seleccionar el subconjunto al frente de la cola, y explorar todos los subconjuntos que se forman añadiéndole un atributo. En caso de que la cola no tenga límite la estrategia hace una búsqueda exhaustiva y si se limita a uno, equivale a una búsqueda secuencial.

Hay dos estrategias que se basan en el clasificador LC (*Linear Classifier*) [74] y BC (*Box Classifier*) [75] desarrolladas por Ichino y Sklanksy. Se caracterizan porque aplican programación entera para seleccionar el subconjunto de atributos.

#### Heurístico (determinista)

Dos estrategias muy populares son la SFS (*Sequential Forward Selection*) y la SBE (*Sequential Backward Elimination*) [27]. La primera de ellas parte de un subconjunto inicial vacío y en cada etapa se añade la variable que mejore en mayor medida la evaluación. La estrategia SBE parte del subconjunto de todas las variables y elimina, en cada etapa, la variable sin la cual mejora en mayor medida la evaluación. Ambos paran en caso de que el movimento realizado no consiga mejorar la evaluación de la mejor solución.

El SBE-SLASH [18] es una modificación de la Búsqueda Secuencial hacia Atrás que se basa en eliminar en cada iteración, los atributos no usados en un clasificador del tipo de árbol de decisión como el ID3 [122].

La estrategia SFFS [116] (*Sequential Forward Floating Selection*), parte de un subconjunto de atributos vacío que, en cada iteración, aplica alternativamente las estrategias SFS y SBE hasta que no haya ninguna mejora. Tras incluir los atributos con los que se obtiene una mejor tasa de error se pasa a eliminar aquellos sin los cuales dicha tasa de error mejora. Puede suceder que la etapa SBE no elimine ningún atributo, teniendo una estrategia SFS.

La estrategia Búsqueda Bidireccional BDS [29] (*Bidirectional Search*) es una estrategia que lleva a cabo la búsqueda desde ambos extremos; un extremos partiendo de todos los atributos y otro del subconjunto vacío. El PQSS (*(p,q) Sequential Search*) es una búsqueda bidireccional que permite añadir y eliminar atributos. Si comienza con el conjunto vacío, favorecerá la inclusión de atributos, y si comienza con el conjunto de todos los atributos, favorecerá su eliminación.

Una variante del $SFS$ es el algoritmo de Queiros y Gelsema [118], el cual aplica, en cada iteración y atributo, diferentes interacciones con el conjunto de atributos previamente seleccionado.

Otra estrategia es el RC [30] (*Relevance in Context*), que parte de la suposición de que algunas variables son relevantes sólo en ciertas partes del espacio. Para ello considera información local a cada ejemplo del conjunto de entrenamiento para tomar la decisión de si una variable ha o no de ser eliminada. Las variables con valores distintos en ejemplos pertenecientes a una misma clase son eliminados si la tasa de error del clasificador disminuye. En caso contrario se pasa a seleccionar otro ejemplo.

En el algoritmo RACE [107] varios subconjuntos de variables compiten para alcanzar el óptimo. La validación cruzada se estima con un conjunto reducido de ejemplos que progresivamente va aumentando. Los conjuntos con menor probabilidad se van eliminando a lo largo de la ejecución.

La estrategia Oblivion [90] es una estrategia tipo SBE que utiliza el clasificador Oblivious y su tasa de error en el conjunto de entrenamiento para guiar la búsqueda. Inicialmente genera un árbol de decisión con todos

los atributos y, en cada iteración, genera todos los posibles árboles que puedan surgir de podar un atributo al árbol de partida en la iteración actual. En caso de que la tasa de error mejore o iguale, el árbol es actualizado y se pasa a la siguiente iteración. En cualquier otro caso la búsqueda finaliza.

El algoritmo IS [137] (*Importance Score*) ordena los atributos basándose en una medida de dependencia, a partir de la cual obtiene los atributos que inducen el conjunto de reglas con menor error.

La propuesta de Holte llamada 1-R [73] propone ordenar los atributos en función de su tasa de error individual de las reglas generadas y posteriormente seleccionarlos en función de este orden.

Una estrategia basada en la Búsqueda Secuencial hacia Atrás es la llamada RFE [66] (*Recursive Feature Elimination*), que pretende seleccionar el subconjunto de atributos con el que se consiga un margen de separación entre clases mayor usando como clasificador las Máquinas de Vectores de Soporte [138]. Hay que seleccionar el numero de atributos con el que nos queramos quedar.

Basada en la anterior estrategia encontramos el método RFE+ADJ [5] que parte del RFE para ordenar los atributos, y posteriormente selecciona un subconjunto de ellos empleando la métrica ADJ [130] (ADJusted distance estimate). Este método sólo puede aplicarse en caso de que se tengan más ejemplos que atributos.

### Heurístico (no determinista)

El método envolvente de Las Vegas LVW (*Las Vegas Wrapper*) [94] es una variante del LVF. Sólo se diferencian en el método de evaluación.

Los Algoritmos Genéticos [134, 137] (ver 2.5.3) han sido muy utilizados para este tipo de problemas. En [134], por ejemplo, se usa una función de ajuste para buscar el subconjunto de atributos que tenga menor cardinalidad y una tasa de error inferior a un umbral que se fija.

El Recocido Simulado (*Simulated Annealing*, SA) [29] (ver 2.5.2) sigue un proceso de búsqueda simulando un proceso de enfriamiento.

Los Algoritmos de Estimación de Distribuciones EDA (ver 2.5.3) (*Es-*

*timation Distribution Agorithms*) también han sido aplicados considerando diversas Redes Bayesianas, como el FSSEBNA [76, 77], que es un algoritmo evolutivo que considera la Red Bayesiana EBNA [38] (*Estimation of Bayesian Network Algorithm*) para inducir el modelo probabilístico.

### 3.4.6 Tabla resumen

En la Tabla 3.1 resumimos las estrategias de selección de atributos anteriormente comentadas con los nombres y las referencias de dichos trabajos. En las columnas la división de los trabajos se hace en función del tipo de búsqueda mientras que en las filas se hace en función del tipo de evaluación.

## 3.5 Formulación

En esta sección vamos a definir formalmente el problema de la selección de atributos en tareas de clasificación.

Sea $\mathcal{E}$ un conjunto formado por $n$ objetos $\{e(i) : i = 1, \ldots, n\}$. Cada objeto viene descrito por $d$ atributos $X = \{X_j : j = 1, \ldots, d\}$ de carácter cualitativo o cuantitativo. Se dispone de un atributo adicional $\mathcal{C}$ que describe la clase a la que pertenece el objeto. Sea $\mathbf{e_i} = (e_{i1}, \ldots, e_{id})$ el vector que representa los valores de las variables que describen un objeto de modo que $e_{ij}$ representa el valor del atributo $X_j$ del vector $\mathbf{e_i}$ dentro del dominio $D_j$ y sea $c_i \in \mathcal{C}$ la clase a la que pertenece. Entonces podemos representar el objeto por la dupla $e(i) = \langle \mathbf{e_i}, c_i \rangle$. El dominio de objetos es:

$$D = D_1 \times D_2 \times \cdots \times D_d = \prod_{j=1}^{d} D_j.$$

Sea $\mathcal{E}_\mathcal{T}$ un conjunto formado por $m$ objetos $\{\widehat{e}(i) : i = 1, \ldots, m\}$ descritos por el mismo conjunto de atributos $\{X_j : j = 1, \ldots, d\}$ que describían los objetos de $\mathcal{E}$ pero a diferencia de éstos el atributo $C$ es desconocido para todo $\widehat{e} \in \mathcal{E}_\mathcal{T}$. La dupla que representa cada objeto es de la forma $\widehat{e}(i) = \langle \widehat{\mathbf{e}}_\mathbf{i}, \widehat{c}_i \rangle$ con $\widehat{\mathbf{e}}_\mathbf{i} = (\widehat{e}_{i1}, \ldots, \widehat{e}_{id})$ el vector de atributos y $\widehat{c}_i$ la variable clase por estimar. A

partir de este momento diferenciaremos a los objetos con clase conocida de aquellos con clase desconocida llamándolos *ejemplos* e *instancias* respectivamente.

En el problema de selección de variables aplicado a la clasificación existen diversos objetivos a perseguir. En nuestro caso será encontrar el subconjunto de atributos $S \subseteq X$ mínimo tal que la precisión del clasificador sea máxima. El dominio $D_S$ viene dado por

$$D_S = D_{(1)} \times D_{(2)} \times \cdots \times D_{(s)} = \prod_{j=1}^{s} D_{(j)}.$$

con $s$ el número de atributos del subconjunto $S$ y $D_{(j)}$ el dominio del $j$-ésimo atributo del subconjunto $S$, que no tiene por qué coincidir con $D_j$.

Sea $f_S(.)$ la función encargada de evaluar la calidad del subconjunto de atributos $S$, entonces

$$\begin{aligned} f_S : D &\to S \\ \mathbf{e_i} &\to \mathbf{e_i^S} = f_S(\mathbf{e_i}) \ : \ i = 1, \ldots, m \end{aligned}$$

con $\mathbf{e_i^S} = (e_{i1}^S, \ldots, e_{is}^S) = (e_{i(1)}, \ldots, e_{i(s)})$. De esta forma los ejemplos vendrán dados por la dupla $e^S(i) = \langle \mathbf{e_i^S}, c_i \rangle : i = i, \ldots, n$ y las instancias por la dupla $\widehat{e}^S(i) = \langle \widehat{\mathbf{e}}_\mathbf{i}^\mathbf{S}, \widehat{c}_i \rangle : i = 1, \ldots, m$.

Dados los conjuntos $\mathcal{E}$ y $\mathcal{E}_\mathcal{T}$ la clasificación de cada instancia $\widehat{e}^S \in \mathcal{E}_\mathcal{T}$ mediante el conjunto de ejemplos $e^S \in \mathcal{E}$ es una aplicación del dominio $D_S \subseteq D$ que asocia a cada instancia una clase de $\mathcal{C}$ en función de los ejemplos. La función encargada de llevar a cabo dicha tarea es el clasificador $\mathcal{L}$ de modo que

$$\begin{aligned} \mathcal{L} : D_S &\to \mathcal{C} \\ \widehat{\mathbf{e}}_\mathbf{i}^\mathbf{S} &\to \mathcal{L}(\widehat{\mathbf{e}}_\mathbf{i}^\mathbf{S}) \ : \ i = 1, \ldots, m \end{aligned}$$

De esta forma se tiene que $\widehat{c}_i = \mathcal{L}(\widehat{\mathbf{e}}_\mathbf{i}) = \mathcal{L}(\widehat{\mathbf{e}}_\mathbf{i}^\mathbf{S})$ y la dupla de la instancia puede expresarse como $\widehat{e}^S(i) = \langle \widehat{\mathbf{e}}_\mathbf{i}^\mathbf{S}, \mathcal{L}(\widehat{\mathbf{e}}_\mathbf{i}^\mathbf{S}) \rangle$.

Un inductor $\mathcal{I}$ es el algoritmo encargado de generar un modelo $\mathcal{M}$ que generaliza el conjunto de ejemplos $\mathcal{E}$. Dicho modelo inducido es usado como clasificador $\mathcal{L} = \mathcal{I}(\mathcal{E})$ para etiquetar nuevas instancias.

Tabla 3.1: Estrategias de selección de atributos en función del tipo de evaluación.

| Búsqueda | Completa | Heurística | |
|---|---|---|---|
| | | determinista | no determinista |
| **Evaluación** | | | |
| Distancia | *B&B* [111] | *Relief* [81] | |
| | *BFF* [147] | *ReliefF* [86] | |
| | *Seg*84 [131] | | |
| | *EUBAFES* [128] | | |
| Información | *MDLM* [132] | *SFG* [17] | |
| | | *DT − CBL* [16] | |
| | | *DTM* [17] | |
| | | *KS*96 [85] | |
| | | *FCBF* [150] | |
| | | *MIFS* [7] | |
| | | *CR* [142] | |
| Dependencia | | *POE + ACC* [109] | |
| | | *PRESET* [106] | |
| Consistencia | *FOCUS* [3] | *SetCover* [21] | *LVF* [95] |
| | *Sch*93 [129] | *VCC* [143] | *SLV* [96] |
| | *MIFES*1 [114] | | *QBB* [23] |
| | *ABB* [93] | | |
| Error | *AMB&B* [44] | *SFS* [27] | *LVW* [94] |
| | *BS* [29] | *SBE* [27] | *GA* [134] |
| | *LC* [74] | *SBE − SLASH* [18] | *SA* [29] |
| | *BC* [75] | *SFFS* [116] | *FSSEBNA* [76] |
| | *PQSS* [29] | *BDS* [29] | |
| | | *RACE* [107] | |
| | | *QG*84 [118] | |
| | | *RC* [30] | |
| | | *RACE* [107] | |
| | | *Oblivion* [90] | |
| | | *IS* [137] | |
| | | 1 − *R* [73] | |
| | | *RFE* [66] | |
| | | *RFE + ADJ* [5] | |

# Capítulo 4

# Búsqueda Dispersa

## 4.1 Introducción

El propósito de esta sección es describir con mayor detalle la metaheurística Búsqueda Dispersa [87], así como explicar la aplicación de la misma al problema de la selección de atributos.

La Búsqueda Dispersa es un algoritmo evolutivo que considera un subconjunto de soluciones de tamaño moderado que evoluciona a lo largo del proceso de búsqueda mediante mecanismos inteligentes de combinación entre soluciones. A diferencia de otras estrategias de combinación existentes en la literatura, como los operadores de cruce y mutación en los Algoritmos Genéticos, donde buena parte del proceso hace uso de cambios aleatorios, la búsqueda de un óptimo local en la Búsqueda Dispersa es una tarea guiada. Este conjunto de soluciones que evoluciona se denomina Conjunto de Referencia ($ConjRef$) y se obtiene a partir de una población de soluciones de mayor tamaño. Las soluciones de la población que se usarán para construir el Conjunto de Referencia se obtienen atendiendo a criterios de intensificación y diversificación del proceso de búsqueda. De esta manera, en el Conjunto de Referencia generado habrá tanto soluciones de calidad según los valores de la función objetivo como soluciones diversas que permiten explorar distintas

regiones del espacio de búsqueda.

Los principios de la metaheurística Búsqueda Dispersa fueron introducidos en la década de los 70 como una extensión a la formulación para combinar reglas de decisión y restricciones de problemas. La propuesta inicial generaba soluciones teniendo en cuenta las características de las distintas regiones del espacio de búsqueda [56]. Una característica importante de la Búsqueda Dispersa es su asociación con la metaheurística Búsqueda Tabú y el hecho de que la búsqueda puede ser mejorada mediante la inclusión de formas particulares de memoria adaptativa y mecanismos asociados de explotación de la misma [58]. En cualquier caso, la Búsqueda Dispersa tiene un tipo de memoria implícita, que puede considerarse como una memoria hereditaria, dado que almacena las mejores soluciones encontradas durante la búsqueda y selecciona sus *buenas* características para crear nuevas soluciones prometedoras. Debido a que esta memoria puede no ser suficiente para alcanzar un óptimo global del problema o una solución muy cercana, los principios de memoria adaptativa de la Búsqueda Tabú pueden mejorar la eficiencia de la Búsqueda Dispersa. Consecuentemente, diversos algoritmos híbridos, resultado de la combinación de estas dos metaheurísticas, han sido propuestos para resolver problemas de optimización. El esquema de la Búsqueda Dispersa, propuesto por Fred Glover en 1998 [60], resume la descripción general del método dada en [58].

La Búsqueda Dispersa está formada por cinco procedimientos: *Método de Generación de Diversidad*, *Método de Mejora*, *Método de Actualización del Conjunto de Referencia*, *Método de Generación de Subconjuntos*, y *Método de Combinación de Soluciones*. A continuación se describen estos cinco métodos esenciales de la metaheurística búsqueda dispersa, cuyo esquema básico resumimos en la Figura 4.1.

1. *Método de Generación de Diversidad*. Con este método se genera una población, *Pob*, de soluciones diversas. El tamaño de este conjunto suele estar en torno a 100 aunque depende de muchas variantes. Generalmente tiene un orden de magnitud superior al Conjunto de Referencia.

2. *Método de Actualización del Conjunto de Referencia*. Este método tiene

1: **Procedimiento** *Búsqueda Dispersa*
2: {
3: $Pob \leftarrow$*GeneraPoblación*();
4: $ConjRef \leftarrow$*Método de Actualización del Conjunto de Referencia*($Pob$);
5: **repetir**
6: $ConjRef' \leftarrow \emptyset$
7: **repetir**
8: $SubConj \leftarrow$*Generación de Subconjuntos*($ConjRef$);
9: $SubConj_n \leftarrow$*Combinación*($SubConj$);
10: $SubConj_m \leftarrow$*Mejora*($SubConj_n$);
11: $ConjRef' \leftarrow ConjRef' \cup SubConj_m$
12: **hasta** ($CriterioDeParada_1$)
13: $ConjRef \leftarrow$ *Actualización*($ConjRef$,$ConjRef'$);
14: **hasta** ($CriterioDeParada_2$)
15: }

Figura 4.1: Pseudocódigo de la Búsqueda Dispersa.

las funciones de construir y actualizar el conjunto de soluciones de referencia. De entre el conjunto de soluciones diversas generado con el método anterior y las soluciones obtenidas tras aplicar el método de mejora, se selecciona el conjunto de referencia. La mitad de sus soluciones, $|ConjRef1|$, serán aquellas con mayor valor de la función objetivo de la población y la otra mitad, $|ConjRef2|$, se seleccionan siguiendo el criterio de diversidad, es decir, seleccionando aquellas que disten más (según la medida de diversidad considerada en el problema) respecto a las ya incluidas en el Conjunto de Referencia. Las soluciones seleccionadas se ordenan según su calidad (valor de la función objetivo), de mayor a menor calidad.

3. *Método de Generación de Subconjuntos*. A través de este método se generan subconjuntos de soluciones del conjunto de referencia, que son posteriormente combinadas para dar origen a otras soluciones. En un esquema básico de Búsqueda Dispersa, un criterio usado generalmente para obtener los subconjuntos consiste en considerar todos los pares de soluciones del conjunto de referencia.

4. *Método de Combinación de Soluciones*. Con este método se combinan las soluciones de cada subconjunto obtenido con el método anterior de generación de subconjuntos.

5. *Método de Mejora*. Dada una solución, este método aplica un procedimiento, habitualmente de búsqueda local, con objeto de mejorar dicha solución.

Una vez introducidos de forma breve y concisa los cinco componentes del proceso básico de la búsqueda dispersa, podemos explicar el proceso de la búsqueda. El proceso comienza generando una población inicial de soluciones diversas, mediante el *Método de Generación de la Diversidad*. Este método permite obtener una población formada por soluciones pertenecientes a distintas regiones del espacio de búsqueda. Para generar esta población se pueden aplicar diversas estrategias, aunque siempre es necesario aplicar procedimientos de generación con algún tipo de aleatoriedad para alcanzar

cierto grado de diversidad. Una vez generada la población, el *Método de Actualización del Conjunto de Referencia* selecciona un conjunto de buenas soluciones representativas para formar el *Conjunto de Referencia*. Por buenas soluciones no nos referimos únicamente a aquellas con los mejores valores de la función objetivo, sino también a soluciones que sean diversas respecto a las anteriores. Se consideran las $|ConjRef1|$ soluciones de la población con mejor valor de la función objetivo y se añade $|ConjRef2|$ veces la solución de la población más dispersa con respecto a las incluidas en el conjunto de referencia hasta el momento. A continuación, el *Método de Generación de Subconjuntos* selecciona varios subconjuntos de soluciones de $ConjRef$, que son luego combinadas por el *Método de Combinación de Soluciones* teniendo en cuenta sus buenas características. El *Método de Mejora* se aplica entonces a las nuevas soluciones surgidas de la combinación. Finalmente, el *Método de Actualización del Conjunto de Referencia* usa las soluciones obtenidas para actualizar el Conjunto de Referencia, $ConjRef$.

## 4.2 Aplicación al problema de selección de atributos

En esta sección explicamos la forma en la que se implementa el procedimiento de Búsqueda Dispersa para resolver el problema de selección de atributos objeto de este trabajo. Para ello, describiremos cada uno de los métodos componentes de la búsqueda.

### 4.2.1 Creación de la población

Inicialmente se crea una población inicial ($Pob$), la cual debe estar formada por soluciones de calidad dispersas que pertenezcan a distintas regiones del espacio de búsqueda. Esta población puede ser generada mediante la aplicación de diversas estrategias; por ejemplo, mediante procedimientos aleatorios para lograr un cierto nivel de diversidad. En el caso del problema de selección de atributos que nos ocupa, el espacio de soluciones depende del número

de atributos $d$ del problema, con lo que habrá que tenerlo en cuenta en el momento de fijar su tamaño.

Para generar una solución, usamos el método de generación de diversidad de la Figura 4.2. Generamos un vector de pesos de atributos $P(X) = (P(X_1), \ldots, P(X_d))$, dado por $P(X_j) = T(\{X_j\})$. Estos pesos indican la calidad de las características para clasificar por sí solas. Sea $L$ el conjunto de atributos $X_j$ con los mayores pesos $P(X_j)$. La estrategia propuesta consiste en seleccionar aleatoriamente una de los $|L|$ mejores atributos mientras su inclusión mejore la solución de forma iterativa.

1. $S \leftarrow \emptyset$.
2. Repetir
   (a) Seleccionar al azar una característica de $L$. Sea $X_{j*}$ la característica seleccionada.
   (b) Si $f_T(\{X_{j*}\} \cup S) \geq f_T(S)$ entonces
   $S \leftarrow S \cup \{X_{j*}\}$ y
   Sea $X_j \notin L$ la característica con el mayor $P(X_j)$, entonces
   $L \leftarrow (L \setminus \{X_{j*}\}) \cup \{X_j\}$

hasta que no se alcance ninguna mejora

Figura 4.2: Método de Generación de Diversidad.

### 4.2.2 Generación del Conjunto de Referencia

Para generar el Conjunto de Referencia, $ConjRef$, seleccionamos un conjunto de soluciones representativas de la población. Tal como indicamos en el esquema básico de la Búsqueda Dispersa, las soluciones a incluir no se limitan únicamente a aquellas con mejor valor de la función objetivo $f$; sino que constituyen las $|ConjRef1|$ soluciones con mejor valor de $f$ y las $|ConjRef2|$ soluciones que proporcionen mayor diversidad. Por tanto, el Conjunto de Referencia generado cuenta con $|ConjRef| = |ConjRef1| + |ConjRef2|$ solu-

ciones. Sea $C$ el conjunto de atributos que pertenecen a las soluciones del Conjunto de Referencia $ConjRef$:

$$C = \bigcup_{S \in ConjRef} S.$$

Definimos la diversidad de una solución dada, $S$, con respecto al Conjunto de Referencia $ConjRef$ mediante la diferencia simétrica entre $S$ y $C$, dada por:

$$Div(S) = Diff(S, C) = |(S \cup C) \setminus (S \cap C)|.$$

El algoritmo propuesto para generar el Conjunto de Referencia es descrito en la Figura 4.3.

---

1. Inicializar:
   (a) Sea $ConjRef \leftarrow \emptyset$.
   (b) Añadir a $ConjRef$ las $|ConjRef1|$ mejores soluciones en $Pob$.
   (c) Obtener el conjunto inicial de atributos: $C = \cup_{S \in ConjRef} S$.
2. Repetir
   (a) Para cada solución $S \notin ConjRef$, calcular $Div(S, C)$.
   (b) $S^* \leftarrow \arg\max Div(S, C) : S \notin ConjRef$.
   (c) $ConjRef \leftarrow ConjRef \cup S^*$.
   (d) $|ConjRef| \leftarrow |ConjRef| + 1$.
   (e) Actualizar $C$

   hasta $|ConjRef| = |ConjRef1| + |ConjRef2|$.

---

Figura 4.3: Método de Actualización del Conjunto de Referencia.

### 4.2.3 Selección de subconjuntos

Consideramos, como es habitual en la mayoría de las implementaciones de la Búsqueda Dispersa que aparecen en la literatura, todos los subconjuntos de dos soluciones del Conjunto de Referencia. En la aplicación que hemos realizado en este trabajo para resolver el problema de selección de atributos, las soluciones combinadas generan dos nuevas soluciones y no sólo una como es común en las implementaciones de Búsqueda Dispersa.

### 4.2.4 Combinación de soluciones

El método de combinación pretende que las buenas características de las soluciones seleccionadas se combinen y formen parte de las nuevas soluciones. El objetivo es obtener buenas soluciones diferentes a las ya presentes en $ConjRef$.

Consideramos dos métodos de combinación, ambos de tipo voraz (*greedy*); la Combinación Voraz o GC (*Greedy Combination*) y la Combinación Voraz Reducida o RGC (*Reduced Greedy Combination*). Cada combinación genera dos nuevas soluciones, $S_1'$ y $S_2'$ a partir de dos soluciones de partida $S_1$ y $S_2$. Ambas comienzan añadiendo a $S_1'$ y $S_2'$ los atributos comunes a $S_1$ y $S_2$ (y eliminando estos atributos de las soluciones de partida). Entonces, en cada iteración, uno de los atributos aún presentes en $S_1$ o $S_2$ es añadido a $S_1'$ o $S_2'$ y se actualiza $S_1$ o $S_2$, dependiendo de la solución en la que se encontraba el atributo. En la Figura 4.4 se describe la Combinación Voraz en mayor detalle.

La estrategia Combinación Voraz Reducida o RGC difiere de la estrategia anterior en que en vez de considerar todo el conjunto de atributos en $C = (S_1 \cup S_2) \setminus (S_1 \cap S_2)$, usa únicamente aquellos atributos presentes en las soluciones con mejor valor de la función objetivo $f$. El conjunto inicial $C$ es reducido aplicando la siguiente estrategia. Sea $Q$ un vector de pesos definido del siguiente modo. Para cada característica $X_j \in C$, $Q(X_j)$ es el promedio estimado del porcentaje de acierto de todas las soluciones en las que estaba presente la característica $X_j$. Entonces, $Q(X_j)$ viene dado del siguiente modo:

1. Inicializar las nuevas soluciones:
   $S_1' \leftarrow S_1 \cap S_2$.
   $S_2' \leftarrow S_1 \cap S_2$.
   Sea $C = (S_1 \cup S_2) \setminus (S_1 \cap S_2)$.

2. Repetir

   (a) Para cada característica $X_j \in C$, evaluar $f_T(S_1' \cup \{X_j\})$ y $f_T(S_2' \cup \{X_j\})$.

   (b) Sean $j_1^*$ y $j_2^*$ los atributos tales que

   $$f_T(S_1' \cup \{X_{j_1^*}\}) = \max_j\{f_T(S_1' \cup \{X_j\})\}$$

   y

   $$f_T(S_2' \cup \{X_{j_2^*}\}) = \max_j\{f_T(S_2' \cup \{X_j\})\}$$

   , respectivamente.

   (c) Si $f_T(S_1' \cup \{X_{j_1^*}\}) > f_T(S_1')$ o $f_T(S_2' \cup \{X_{j_2^*}\}) > f_T(S_2')$ entonces:

      i. Si $f_T(S_1' \cup \{X_{j_1^*}\}) > f_T(S_1')$ y $f_T(S_2' \cup \{X_{j_2^*}\}) \leq f_T(S_2')$, hacer $k \leftarrow 1$.

      ii. Si $f_T(S_1' \cup \{X_{j_1^*}\}) \leq f_T(S_1')$ y $f_T(S_2' \cup \{X_{j_2^*}\}) > f_T(S_2')$, hacer $k \leftarrow 2$.

      iii. Si $f_T(S_1' \cup \{X_{j_1^*}\}) > f_T(S_1')$ y $f_T(S_2' \cup \{X_{j_2^*}\}) > f_T(S_2')$, hacer $k \leftarrow \arg\max_{\{} f_T(S_1' \cup \{X_{j_1^*}\}), f_T(S_2' \cup \{X_{j_2^*}\})\}$. Si $f_T(S_1' \cup \{X_{j_1^*}\}) = f_T(S_2' \cup \{X_{j_2^*}\})$, entonces $k$ es el índice 1 ó 2, correspondiendo a la solución con el menor número de características. Si ambas soluciones tienen el mismo número de atributos, elegimos $k$ al azar.

   Añadir $X_{j_k^*}$ a la solución $S_k'$, hacer $C \leftarrow C \setminus X_{j_k^*}$ e ir al paso 2.

hasta que no haya mejora; es decir, $f_T(S_1' \cup \{X_{j_1^*}\}) \leq f_T(S_1')$ y $f_T(S_2' \cup \{X_{j_2^*}\}) \leq f_T(S_2')$

Figura 4.4: Método de Combinación Voraz.

$$Q(X_j) = \frac{1}{|\{i : X_j \in S_i\}|} \sum_{\{i:X_j \in S_i\}} f(S_i).$$

Sea $\overline{Q}$ la media de los valores $Q(X_j)$ tal que $X_j \in C$,

$$\overline{Q} = \frac{1}{|C|} \sum_{j=0}^{|C|} Q(X_j)$$

La estrategia $CVR$ usa los atributos $X_j \in C$ que cumplen la condición $Q(X_j) \geq \overline{Q}$.

### 4.2.5 Método de mejora

El método de mejora se aplica a todas las soluciones, $S$, generadas tras la aplicación del método de combinación explicado anteriormente. Sea $CA$ el conjunto de atributos que no pertenecen a la solución $S$, entonces se ordenan los atributos $X_j \in CA$ de acuerdo al valor de los pesos $P(X_j)$. El método de mejora se describe en la Figura 4.5.

1. Sean $X_{(1)}, \ldots, X_{(|CA|)}$ los atributos ordenados tales que

$$P(X_{(j)}) \geq P(X_{(j+1)}).$$

2. $j \leftarrow 0$.

3. Repetir

   (a) $j \leftarrow j + 1$

   (b) Si $f_T(S \cup \{X_{(j)}\}) \geq f_T(S)$, entonces $S \leftarrow S \cup \{X_{(j)}\}$

   hasta $(j \leftarrow |CA|)$

Figura 4.5: Método de mejora.

Todas las soluciones obtenidas a partir de la aplicación del método de mejora son almacenadas en el conjunto, $ConjRef'$, que se usará para actualizar el Conjunto de Referencia.

### 4.2.6 Actualización del Conjunto de Referencia

Finalmente, tras obtener todas las soluciones mejoradas, el $ConjRef$ es actualizado en función de los criterios de intensidad y diversidad. Primero, seleccionamos las $|ConjRef|/2$ mejores soluciones de $ConjRef \cup ConjRef'$. Entonces, $ConjRef$ es actualizado atendiendo a la diversidad aplicando el criterio explicado en la Figura 4.3.

## 4.3 Búsqueda Dispersa paralela

Aunque las metaheurísticas proporcionan estrategias efectivas para encontrar soluciones de buena calidad en problemas combinatorios, los tiempos computacionales asociados a la exploración del espacio de soluciones pueden ser muy costosos; como es el caso del problema que nos ocupa. Con la proliferación de computadores paralelos, la implementación de metaheurísticas paralelas surge como una alternativa natural para incrementar tanto la velocidad de búsqueda en el espacio de soluciones como la exploración en el mismo. Esto nos da la posibilidad de resolver problemas de mayor tamaño, así como de encontrar mejores soluciones, respecto a las soluciones encontradas con la versión secuencial.

Por tanto, el paralelismo es una vía posible, no sólo para reducir el tiempo de ejecución del algoritmo de búsqueda, sino también para mejorar su efectividad. La primera paralelización de la Búsqueda Dispersa fue propuesta por García et al. [47]. Dichos autores consideran tres estrategias de paralelización para reducir el tiempo de ejecución y aumentar la capacidad exploratoria en el espacio de soluciones. Además, algunas de estas nuevas estrategias mejoran la calidad de la solución con respecto a la versión secuencial.

En este trabajo, consideramos una estrategia paralela directa para mejorar la precisión de la metaheurística Búsqueda Dispersa sin que aumente el tiempo de ejecución. Podemos obtener implementaciones más precisas de la búsqueda dispersa haciendo uso de diferentes métodos de combinación y ajustes de parámetros en cada procesador, obteniendo así soluciones de alta calidad para diferentes clases de ejemplos del mismo problema.

La Figura 4.6 muestra el pseudocódigo de la implementación paralela que denominamos Búsqueda Dispersa con Combinación Múltiple o *MCSS* (*Multiple Combinations Scatter Search*). Esta implementación aplica diferentes métodos de combinación (definidos por $CombinaSoluciones_r$) en cada procesador ($r = 1, \ldots, n_{pr}$). Tal como explicamos en la sección anterior, hemos diseñado dos métodos de combinación diferentes, que serán usados en la experiencia computacional presentada en la sección 4.4, donde el número de procesadores usados es $n_{pr} = 2$.

La idea de diseñar varios métodos de combinación en un proceso de Búsqueda Dispersa ya había aparecido en algunos trabajos previos de la literatura. Por ejemplo, Campos et al. [14] diseñaron diferentes métodos de combinación para una implementación secuencial de la Búsqueda Dispersa para el problema de ordenación lineal. Estos autores estimaron la contribución relativa de cada método en la calidad final de la solución. Basándose en los resultados obtenidos, usaron finalmente el método de combinación que presentó el mejor comportamiento. Sin embargo, en nuestro trabajo ejecutamos en paralelo los dos métodos de combinación simultáneamente haciendo uso de dos procesadores.

La Búsqueda Dispersa requiere tiempos de computación altos para el problema de selección de atributos, lo cual complica la ejecución secuencial de varios métodos de combinación. El objetivo de la paralelización propuesta en este trabajo es alcanzar una mejora en la calidad de la solución, usando el mismo tiempo computacional que el usado por el algoritmo secuencial.

## 4.4 Resultados Computacionales

### 4.4.1 Motivación y objetivos

En este apartado se analizará y estudiará el rendimiento de las distintas propuestas de la estrategia Búsqueda Dispersa presentadas en este trabajo. El estudio se llevó a cabo considerando las medidas del porcentaje de acierto, tamaño de los subconjuntos de atributos encontrados y tiempo de ejecución empleado.

**procedimiento** Búsqueda Dispersa con Varias Combinaciones
**begin**
    Crear Población ($Pob$, $|ConjRef|$);
    Generar el Conjunto de Referencia ($ConjRef$, $|ConjRef|$);
    **repeat**
        **repeat**
            Selección de Subconjuntos ($SubConj$);
            **para** cada procesador $r = 1, \ldots, n_{pr}$ **hacer** en paralelo
            **comenzar**
                $CombinaSoluciones_r(SubConj, CurSol_r)$;
                MejoraSoluciones($CurSol_r$, $ImpSol_r$);
            **fin;**
        **hasta** ($CriterioParada1$);
        Actualizar Conjunto de Referencia($ConjRef$);
    **hasta** ($CriterioParada2$);
**fin.**

Figura 4.6: Pseudocógido de la Búsqueda Dispersa Paralela.

El análisis de la Búsqueda Dispersa se llevó a cabo en cuatro etapas. En la primera de ellas se estudió el rendimiento de las dos versiones de la Búsqueda Dispersa con distintos valores en los parámetros de entrada. Dichos parámetros de entrada son el tamaño de la población, del Conjunto de Referencia y del conjunto $L$ del método que genera la población. Una vez fijados los valores en los parámetros de entrada se pasó a comparar, en una segunda etapa, los resultados obtenidos con la Búsqueda Dispersa considerando los dos métodos de combinación presentados y con una versión paralela que combinaba, en la búsqueda, ambos métodos. En una tercera etapa se estudió la aportación de la estrategia analizando los resultados obtenidos con los clasificadores con y sin selección de atributos. Por último se pasó a comparar los resultados de la Búsqueda Dispersa con otras estrategias evolutivas, el Algoritmo Genético [64] y el FSSEBNA [76, 77].

De esta forma los objetivos podemos resumirlos en los siguientes puntos:

- Ajustar los parámetros de entrada de la Búsqueda Dispersa (BD).
- Analizar el rendimiento de las dos versiones de la Búsqueda Dispersa.
- Estudiar si la versión paralela que usa ambos métodos de combinación mejora los resultados obtenidos por las versiones secuenciales.
- Examinar la aportación de la Búsqueda Dispersa comparando el rendimiento de los clasificadores con y sin selección de atributos.
- Comparar el rendimiento de la Búsqueda Dispersa con el Algoritmo Genético (AG) y el *Feature Subset Selection by Estimation of Bayesian Network Algorithm* (FSSEBNA).

### 4.4.2 Descripción de los experimentos

Los experimentos se llevaron a cabo considerando los clasificadores Naïve Bayes, IB1 y J48 de la librería weka [146]. Como método de evaluación se consideró la validación cruzada 5x2cv [28] con el test F [4] para estimar si las diferencias obtenidas en un conjunto de datos eran o no significativas.

Siguiendo las recomendaciones de Demšar [26], se aplicó el test de Friedman [45, 46] para estimar si el rendimiento de las distintas estrategias estudiadas sobre todos los conjuntos de datos en estudio son o no diferentes de forma significativa. En caso de encontrar diferencias se consideraron comparaciones dos a dos aplicando el test de Nemenyi [112] con el procedimiento de Hochberg [70].

El test F aplicado sobre los resultados obtenidos en un mismo conjunto de datos tiene errores de tipo I que pueden llegar a ser elevados (falsos positivos, es decir que erróneamente se rechaza la hipótesis nula), con lo que se adoptó una postura conservadora y se consideraron intervalos de confianza al 95 % y 99 %. Este tipo de errores no son significativos en el test de Friedman aplicado sobre los promedios obtenidos con todos los conjuntos de datos, con lo que se consideraron intervalos de confianza al 90 % y 95 %.

Se consideraron conjuntos de datos reales del repositorio de la UCI [113] (ver descripción en B.2) para llevar a cabo las distintas pruebas. Para la comparativa con las otras estrategias evolutivas también se consideraron problemas generados artificialmente (ver descripción en B.1).

En el estudio de los parámetros de la Búsqueda Dispersa, así como en el que se lleva a cabo la comparación entre las distintas versiones de BD sólo se consideró el clasificador IB1. Aunque sería conveniente extender este estudio a otros clasificadores, la similitud en el procedimiento de búsqueda de las distintas versiones presentadas influirían del mismo modo en todas ellas.

Con objeto de comparar el rendimiento de la estrategia Búsqueda Dispersa con el AG y FSSEBNA, impusimos la condición de que todas las estrategias examinaran un mismo número de soluciones. Se consideró que cada estrategia examinara un total de 4000 soluciones y una población inicial de 1000 individuos. Siguiendo las recomendaciones encontradas en [76, 77], el número de individuos seleccionados por iteración se fijó a la mitad de la población inicial (500) y el tamaño de la nueva población se fijó en 999 individuos. En realidad esta estrategia examina, con los parámetros considerados, un total de 3996 soluciones. En la selección de los individuos se consideró elitismo y se consideraron 3 generaciones. Igualmente para el AG se consideró una población inicial de 1000 individuos y un total de 3 generaciones. Siguiendo las

recomendaciones de Yang y Honavar [149] la probabilidad de cruce se fijó en 0,6 y la de mutación en 0,001.

### 4.4.3 Estudio de parámetros de la Búsqueda Dispersa.

El primer estudio se centró en seleccionar la mejor combinación en los valores de los parámetros de entrada de la Búsqueda Dispersa. Por mejor combinación entendemos que serán aquellos valores con el que el clasificador obtenga la menor tasa de error con el subconjunto de atributos encontrados en un conjunto de datos seleccionados. Para ello se consideraron conjuntos de datos reales del repositorio de la UCI [113] (ver la descripción de los conjuntos en los apéndices B.2).

La Tabla 4.1 muestra los resultados obtenidos por la BD al comparar los dos métodos de combinación con distintos valores para los parámetros de entrada. Se usaron distintos tamaños del conjunto $L$, $|L| = 3, d/2$, donde $d$ es el número de atributos del conjunto de datos. Para ambos casos se consideraron los tamaños del Conjunto de Referencia $|ConjRef| = 5, 10$ y además $|ConjRef| = d^2$ sólo para el caso $|L| = d/2$. Como criterio de parada se consideró la convergencia, de la búsqueda, a la mejor solución.

En las pruebas realizadas, las diferencias en el acierto y en el tamaño de los subconjuntos encontrados son pequeñas con ambos métodos de combinación no siendo estadísticamente significativas. Sin embargo, para ambos métodos de combinación, la Búsqueda Dispersa con $L = d/2$ y $|ConjRef| = d^2$ es la que encuentra, en promedio, el subconjunto de atributos de menor tamaño. Debido a esto, éstos serán los parámetros que consideramos para comprobar si existen diferencias entre ambos métodos de combinación y la propuesta paralela de la Búsqueda Dispersa con ambos métodos de combinación. Sin embargo, para las pruebas posteriores con las que que comparamos la Búsqueda Dispersa con los algoritmos evolutivos Algoritmo Genético y FSSEBNA, tendremos que estas estrategias llevan a cabo un número fijo de evaluaciones, con lo que en estas pruebas se modificó el criterio de parada para que la estrategia llevara a cabo un número de evaluaciones fijo. Si consideramos los mismos valores de parámetros de entrada puede suceder que la estrategia

no pueda completar ni la primera iteración debido al alto número de evaluaciones por iteración llevado a cabo. De esta forma se consideraron los valores $L = 3$ y $|ConjRef| = 5$. La robustez mostrada por la estrategia ante los distintos valores permite que no haya preferencias para considerar uno u otro valor.

### 4.4.4 Comparativa del rendimiento de la versión secuencial de la Búsqueda Dispersa, con ambos métodos de combinación, y la versión paralela

En las Tablas 4.2 y 4.3 podemos ver los resultados del porcentaje de acierto y número de atributos respectivamente obtenidos sobre conjuntos de datos reales con el clasificador IB1. La linea horizontal separa los resultados obtenidos en problemas de tamaño pequeño con los de tamaño medio. La primera columna muestra los valores con el clasificador base sin selección de atributos. En la segunda columna están los resultados con la estrategia Búsqueda Dispersa considerando la Combinación Voraz mientras que en la tercera se muestran los de la estrategia con Combinación Voraz Reducida. Por último los resultados de la versión paralela están en la última columna.

Como puede verse en la Tabla 4.2 referida al acierto, los valores promedios alcanzados con las distintas estrategias son próximos entre sí. Las diferencias no son estadísticamente significativas en ninguno de los casos. Los resultados referidos al tamaño del subconjunto de atributos muestran que las versiones secuenciales tienen una capacidad de reducción similar aunque la BD-CVR encuentra, en la mayoría de los casos, subconjuntos de atributos de menor tamaño excepto en los problemas *cra* y *soy*, que encuentra subconjuntos de mayor tamaño, y *vow* y *tse*, que encuentra subconjuntos del mismo tamaño. La versión paralela tiene una capacidad de reducción mayor que las versiones secuenciales, sobre todo en los problemas de mayor tamaño donde en todos los casos la reducción es mayor. En los problemas pequeños, sólo en los conjuntos *hcl* y *cra* consigue reducir en mayor medida. En promedio, la reducción obtenida con las versiones secuenciales fueron del 70,47 % y 71,10 %

Tabla 4.1: Comparativa de las versiones secuenciales de la Búsqueda Dispersa con distintos valores de los parámetros. Se presentan los valores promedios del porcentaje de aciertos y número de atributos

| $\|L\|$ | 3 | | $d/2$ | | |
|---|---|---|---|---|---|
| $\|ConjRef\|$ | 5 | 10 | 5 | 10 | $d^2$ |
| | Búsqueda Dispersa (CV) | | | | |
| | % acierto | | | | |
| cra | 78,64 ± 2,46 | 79,82 ± 3,33 | 80,67 ± 2,10 | 79,71 ± 2,82 | 83,28 ± 3,12 |
| hco | 81,03 ± 3,66 | 81,85 ± 3,24 | 81,63 ± 3,26 | 80,50 ± 4,04 | 76,69 ± 3,49 |
| ion | 89,17 ± 2,60 | 87,18 ± 2,12 | 88,88 ± 3,30 | 87,86 ± 1,86 | 87,75 ± 1,37 |
| ann | 94,41 ± 4,88 | 94,68 ± 4,75 | 95,72 ± 1,71 | 95,63 ± 1,91 | 94,14 ± 3,16 |
| | 85,80 | 85,88 | 86,82 | 85,91 | 85,46 |
| | # atributos | | | | |
| cra | 6,20 ± 0,79 | 6,60 ± 1,07 | 6,60 ± 1,43 | 6,50 ± 1,65 | 3,40 ± 1,43 |
| hco | 7,30 ± 1,49 | 8,80 ± 2,20 | 7,20 ± 2,25 | 8,30 ± 2,06 | 7,40 ± 2,41 |
| ion | 6,90 ± 2,28 | 7,00 ± 2,36 | 6,50 ± 2,92 | 7,10 ± 1,80 | 6,10 ± 1,37 |
| ann | 9,90 ± 2,47 | 10,50 ± 2,17 | 11,40 ± 1,78 | 12,70 ± 2,21 | 8,90 ± 2,89 |
| | 7,57 | 8,22 | 7,92 | 8,65 | 6,45 |
| | Búsqueda Dispersa (CVR) | | | | |
| | % acierto | | | | |
| cra | 80,46 ± 2,83 | 79,51 ± 3,30 | 80,35 ± 2,23 | 79,68 ± 3,44 | 83,91 ± 3,27 |
| hco | 80,71 ± 3,50 | 80,71 ± 3,53 | 78,59 ± 6,16 | 79,51 ± 3,94 | 77,94 ± 2,96 |
| ion | 87,98 ± 2,78 | 88,03 ± 2,47 | 87,23 ± 2,12 | 87,24 ± 2,37 | 87,12 ± 1,24 |
| ann | 94,25 ± 4,84 | 95,14 ± 3,05 | 88,78 ± 16,63 | 95,86 ± 2,37 | 92,98 ± 2,68 |
| | 85,85 | 85,84 | 83,73 | 85,57 | 85,48 |
| | # atributos | | | | |
| cra | 6,80 ± 1,32 | 7,10 ± 1,29 | 6,60 ± 3,12 | 6,80 ± 1,40 | 4,50 ± 2,27 |
| hco | 8,30 ± 1,34 | 8,40 ± 1,71 | 7,30 ± 1,89 | 7,90 ± 1,73 | 6,30 ± 2,16 |
| ion | 6,00 ± 1,83 | 6,90 ± 2,47 | 5,40 ± 2,17 | 7,20 ± 1,14 | 5,70 ± 1,06 |
| ann | 9,90 ± 2,51 | 10,80 ± 2,39 | 10,20 ± 3,16 | 11,60 ± 1,40 | 8,20 ± 2,66 |
| | 7,75 | 8,30 | 7,37 | 8,37 | 6,17 |

CV-Combinación Voraz; CVR-Combinación Voraz Reducida; $|ConjRef|$-tamaño del Conjunto de Referencia; $|L|$-tamaño de la Lista Restringida de Candidatos.

para las estrategias BD-CV y BD-CVR respectivamente y del 76,27 % para la versión paralela.

El test de Friedman aplicado sobre todos los problemas pequeños no muestra diferencias estadísticamente significativas, sin embargo en los problemas medianos sí encuentra diferencias estadísticamente significativas entre la versión paralela y las secuenciales. El test de Nemenyi indica que estas diferencias son significativas con la BD-CV al nivel $\alpha = 0{,}05$ y con la BD-CVR al nivel $\alpha = 0{,}1$.

Tabla 4.2: Comparativa del porcentaje de acierto entre las versiones secuenciales y paralela de la BD. Los resultados mostrados son los valores promedios y su desviación típica.

| cdd | base | BD-CV | BD-CVR | BDPll |
|---|---|---|---|---|
| pdi | $69{,}71 \pm 2{,}68$ | $67{,}92 \pm 2{,}35$ | $67{,}66 \pm 2{,}42$ | $68{,}10 \pm 2{,}43$ |
| bca | $95{,}48 \pm 0{,}70$ | $95{,}22 \pm 1{,}07$ | $94{,}88 \pm 1{,}45$ | $95{,}11 \pm 0{,}90$ |
| bwi | $95{,}61 \pm 0{,}82$ | $94{,}66 \pm 1{,}51$ | $93{,}57 \pm 2{,}23$ | $93{,}67 \pm 2{,}36$ |
| hcl | $75{,}98 \pm 3{,}10$ | $74{,}99 \pm 5{,}31$ | $74{,}99 \pm 5{,}68$ | $74{,}91 \pm 2{,}85$ |
| vow | $95{,}29 \pm 1{,}66$ | $93{,}58 \pm 1{,}39$ | $93{,}58 \pm 1{,}39$ | $93{,}64 \pm 1{,}34$ |
| cra | $81{,}54 \pm 2{,}31$ | $83{,}28 \pm 3{,}12$ | $83{,}91 \pm 3{,}27$ | $83{,}39 \pm 2{,}74$ |
| hco | $75{,}60 \pm 1{,}99$ | $76{,}69 \pm 3{,}49$ | $77{,}94 \pm 2{,}96$ | $76{,}96 \pm 3{,}79$ |
| abp | $95{,}86 \pm 0{,}24$ | $95{,}53 \pm 0{,}33$ | $95{,}44 \pm 0{,}44$ | $95{,}44 \pm 0{,}43$ |
| ion | $85{,}75 \pm 1{,}30$ | $87{,}75 \pm 1{,}37$ | $87{,}12 \pm 1{,}24$ | $87{,}35 \pm 1{,}56$ |
| soy | $85{,}02 \pm 4{,}18$ | $82{,}41 \pm 3{,}49$ | $83{,}65 \pm 3{,}65$ | $80{,}53 \pm 1{,}92$ |
| ann | $93{,}59 \pm 1{,}09$ | $94{,}14 \pm 3{,}16$ | $92{,}98 \pm 2{,}68$ | $91{,}49 \pm 2{,}19$ |
| tse | $92{,}67 \pm 0{,}68$ | $95{,}09 \pm 2{,}76$ | $95{,}12 \pm 2{,}78$ | $93{,}58 \pm 2{,}20$ |
| *Media* | $86{,}84 \pm 1{,}73$ | $86{,}77 \pm 2{,}45$ | $86{,}74 \pm 2{,}52$ | $86{,}18 \pm 2{,}06$ |

CV-Combinación Voraz; CVR-Combinación Voraz Reducida; BDPll-Búsqueda Dispersa Paralela.

### 4.4.5 Comparativa del rendimiento de los clasificadores con y sin estrategia Búsqueda Dispersa de selección de atributos

Esta comparativa pretende, como dijimos anteriormente, estudiar la aportación de la estrategia Búsqueda Dispersa, y para ello se analizaron los resultados obtenidos por los clasificadores con y sin selección de atributos. En el

Tabla 4.3: Comparativa de la reducción entre las versiones secuenciales y paralela de la BD. Los resultados mostrados son los valores promedios y su desviación típica.

| cdd | base | BD-CV | BD-CVR | BD-Pll |
|---|---|---|---|---|
| pdi | 8 | $4{,}10 \pm 0{,}99$ | $4{,}00 \pm 0{,}94$ | $4{,}20 \pm 1{,}14$ |
| bca | 9 | $5{,}20 \pm 1{,}62$ | $4{,}78 \pm 1{,}48$ | $5{,}40 \pm 1{,}71$ |
| bwi | 30 | $6{,}80 \pm 2{,}53$ | $5{,}50 \pm 1{,}43$ | $6{,}00 \pm 2{,}63$ |
| hcl | 13 | $6{,}30 \pm 1{,}64$ | $6{,}20 \pm 2{,}10$ | $5{,}56 \pm 1{,}60$ |
| vow | 10 | $7{,}70 \pm 0{,}68$ | $7{,}70 \pm 0{,}68$ | $8{,}00 \pm 0{,}94$ |
| cra | 15 | $3{,}40 \pm 1{,}43$ | $4{,}50 \pm 2{,}27$ | $2{,}80 \pm 2{,}62$ |
| hco | 21 | $7{,}40 \pm 2{,}41$ | $6{,}30 \pm 2{,}16$ | $4{,}50 \pm 1{,}51$ |
| abp | 29 | $2{,}80 \pm 1{,}48$ | $2{,}70 \pm 1{,}83$ | $2{,}00 \pm 1{,}05$ |
| ion | 34 | $6{,}10 \pm 1{,}37$ | $5{,}70 \pm 1{,}06$ | $3{,}90 \pm 0{,}88$ |
| soy | 35 | $15{,}0 \pm 2{,}71$ | $16{,}5 \pm 2{,}22$ | $12{,}80 \pm 1{,}81$ |
| ann | 38 | $8{,}90 \pm 2{,}89$ | $8{,}20 \pm 2{,}66$ | $6{,}30 \pm 2{,}06$ |
| tse | 25 | $5{,}10 \pm 2{,}47$ | $5{,}10 \pm 2{,}42$ | $1{,}90 \pm 1{,}20$ |
| *Media* | 22,25 | $6{,}57 \pm 1{,}85$ | $6{,}43 \pm 1{,}77$ | $5{,}28 \pm 1{,}60$ |
| *Reducción* | 100,00 % | 70,47 % | 71,10 % | 76,27 % |

CV-Combinación Voraz; CVR-Combinación Voraz Reducida; BDPll-Búsqueda Dispersa Paralela.

apartado anterior se vió que las dos propuestas del método de combinación obtienen resultados similares, con lo que a partir de ahora sólo se considerará la BD-CV y que pasará a denominarse Búsqueda Dispersa o BD.

### Problemas artificiales

En la Tabla 4.4 están los resultados obtenidos con el clasificador Naïve Bayes. Las cuatro primeras columnas describen las características de los conjuntos de datos. Los resultados de la Búsqueda Dispersa en el acierto están en la quinta columna y el tamaño del subconjunto de atributos encontrado en la novena. Análogamente, los correspondientes al clasificador base se encuentran en las columnas octava y duodécima.

En todos los casos la reducción de la Búsqueda Dispersa es significativa, permitiendo mejorar, en promedio, los resultados en el porcentaje de aciertos en todos los casos menos en uno. Dicha mejora en el acierto es estadísticamente significativa en la mayor parte de los casos. El test de Nemenyi aplica-

do sobre todos los resultados indica que las diferencias son estadísticamente significativas al nivel $\alpha = 0{,}05$. Analizando los resultados en función de la regla de clasificación tenemos que para el caso lineal y no lineal las diferencias son estadísticamente significativas al nivel $\alpha = 0{,}05$. Igualmente haciendo el estudio en función del número de atributos relevantes obtenemos que en ambos casos (con 5 y 10 atributos relevantes) las diferencias son significativas al nivel $\alpha = 0{,}05$.

Los porcentajes de acierto y tamaños de los subconjuntos usando el clasificador IB1 están en la Tabla 4.5. Como en el caso anterior, en promedio la reducción es significativa en todos los casos; sin embargo en el acierto la BD mejora los resultados en la mitad de los problemas considerados. Estas diferencias son estadísticamente significativas en 6 casos, de los cuales en 5 de ellos la BD mejora y sólo en uno empeora.

El test de Nemenyi aplicado sobre todos los problemas obtiene que en el acierto las diferencias son estadísticamente significativas al nivel $\alpha = 0{,}1$ y en el tamaño de los subconjuntos al nivel $\alpha = 0{,}05$. Estudiando en detalle los resultados en función de la regla de clasificación tenemos que no hay diferencias significativas; sin embargo al hacer el estudio en función del número de atributos relevantes encontramos que en ambos casos las diferencias son significativas con $\alpha = 0{,}05$. En el caso de que haya 5 atributos relevantes encontramos que la BD mejora los resultados; no así en el otro caso en el que los resultados empeoran.

Los resultados con el clasificador J48 aparecen en la Tabla 4.6. En este caso los resultados obtenidos con el clasificador usando el subconjunto encontrado por la BD mejora los del clasificador base en casi todos los conjuntos de datos. Sólo en dos de ellos los resultados son peores. Las diferencias en el acierto, según el test F, son estadísticamente significativas en 5 casos y en todas ellas la aplicación de la BD mejora los resultados. Como en los casos anteriores, la reducción es estadísticamente significativa en todos los casos.

Aplicando el test de Nemenyi sobre todos los problemas obtenemos que existen diferencias significativas al nivel $\alpha = 0{,}05$ tanto en el acierto como en la reducción. Estudiando los resultados en función de la regla de clasificación encontramos que la mejora de la BD es significativa al nivel $\alpha = 0{,}05$. Anali-

zando los resultados en función del número de atributos relevantes presentes en los datos, tenemos que la aplicación de la BD mejora significativamente al clasificador base al nivel $\alpha = 0{,}05$ en ambos casos.

Tabla 4.4: Comparativa entre la metaheurística BD y las estrategias AG y FSSEBNA con el clasificador Naïve-Bayes. Porcentaje de acierto y reducción en conjuntos de datos artificiales.

| conjuntos de datos | | | | Naïve Bayes | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | % aciertos | | | | # atributos | | | |
| l | r | $\sigma$ | d | BD | AG | FSSEBNA | base | BD | AG | FSSEBNA | base |
| 1 | 5 | 0 | 25 | $89{,}0 \pm 2{,}8$ | $89{,}2 \pm 3{,}3$ | $86{,}8 \pm 3{,}2$ | $83{,}9 \pm 2{,}2$ | $7{,}7 \pm 1{,}9$ | $7{,}6 \pm 2{,}5$ | $11{,}5 \pm 2{,}8$ | 25 $\ll$ |
| | | | 50 | $88{,}5 \pm 3{,}5$ | $86{,}7 \pm 3{,}8$ | $83{,}5 \pm 2{,}2$ | $77{,}6 \pm 2{,}5$ $<$ | $7{,}6 \pm 2{,}0$ | $13{,}4 \pm 6{,}9$ | $20{,}6 \pm 3{,}3$ $\ll$ | 50 $\ll$ |
| | | | 100 | $88{,}9 \pm 2{,}6$ | $84{,}5 \pm 4{,}6$ | $79{,}0 \pm 1{,}5$ $<$ | $72{,}5 \pm 1{,}6$ $\ll$ | $7{,}6 \pm 2{,}2$ | $21{,}2 \pm 13{,}2$ | $43{,}5 \pm 3{,}2$ $\ll$ | 100 $\ll$ |
| | | | 200 | $87{,}7 \pm 4{,}7$ | $79{,}2 \pm 3{,}1$ | $72{,}6 \pm 3{,}5$ $\ll$ | $68{,}1 \pm 3{,}1$ $\ll$ | $7{,}6 \pm 2{,}1$ | $26{,}1 \pm 19{,}1$ | $96{,}4 \pm 7{,}2$ $\ll$ | 200 $\ll$ |
| | | 0,1 | 25 | $89{,}2 \pm 2{,}6$ | $89{,}3 \pm 1{,}3$ | $87{,}8 \pm 2{,}2$ | $83{,}2 \pm 2{,}2$ | $7{,}3 \pm 2{,}2$ | $8{,}3 \pm 2{,}3$ | $11{,}1 \pm 2{,}5$ | 25 $\ll$ |
| | | | 50 | $89{,}1 \pm 3{,}0$ | $86{,}2 \pm 3{,}2$ | $84{,}0 \pm 2{,}8$ | $77{,}4 \pm 2{,}5$ $\ll$ | $7{,}0 \pm 1{,}9$ | $16{,}2 \pm 5{,}6$ | $20{,}8 \pm 4{,}1$ $\ll$ | 50 $\ll$ |
| | | | 100 | $88{,}9 \pm 2{,}4$ | $83{,}5 \pm 2{,}9$ | $79{,}7 \pm 2{,}5$ $\ll$ | $72{,}9 \pm 2{,}7$ $\ll$ | $7{,}1 \pm 2{,}1$ | $22{,}0 \pm 7{,}1$ | $42{,}6 \pm 6{,}9$ $\ll$ | 100 $\ll$ |
| | | | 200 | $88{,}4 \pm 2{,}9$ | $80{,}2 \pm 4{,}8$ | $73{,}4 \pm 3{,}1$ $\ll$ | $68{,}7 \pm 3{,}3$ $\ll$ | $6{,}9 \pm 2{,}0$ | $24{,}1 \pm 17{,}3$ | $97{,}4 \pm 7{,}7$ $\ll$ | 200 $\ll$ |
| | 10 | 0 | 25 | $82{,}4 \pm 5{,}2$ | $83{,}7 \pm 3{,}5$ | $82{,}8 \pm 4{,}2$ | $81{,}1 \pm 3{,}2$ | $10{,}3 \pm 1{,}3$ | $14{,}4 \pm 3{,}1$ | $13{,}5 \pm 2{,}7$ | 25 $\ll$ |
| | | | 50 | $81{,}1 \pm 3{,}8$ | $82{,}0 \pm 2{,}6$ | $81{,}3 \pm 3{,}2$ | $77{,}2 \pm 2{,}2$ | $11{,}1 \pm 2{,}3$ | $19{,}9 \pm 6{,}6$ | $23{,}8 \pm 2{,}2$ $\ll$ | 50 $\ll$ |
| | | | 100 | $80{,}6 \pm 4{,}6$ | $79{,}2 \pm 4{,}3$ | $74{,}7 \pm 3{,}7$ | $69{,}4 \pm 3{,}0$ $<$ | $10{,}2 \pm 1{,}3$ | $22{,}0 \pm 8{,}0$ | $46{,}8 \pm 3{,}1$ $\ll$ | 100 $\ll$ |
| | | | 200 | $79{,}6 \pm 4{,}8$ | $72{,}4 \pm 3{,}0$ | $67{,}6 \pm 2{,}4$ $<$ | $64{,}9 \pm 3{,}0$ $\ll$ | $10{,}0 \pm 2{,}6$ | $31{,}5 \pm 18{,}3$ | $99{,}3 \pm 7{,}5$ $\ll$ | 200 $\ll$ |
| | | 0,1 | 25 | $82{,}9 \pm 4{,}4$ | $83{,}3 \pm 2{,}6$ | $83{,}8 \pm 2{,}8$ | $81{,}1 \pm 3{,}8$ | $11{,}7 \pm 2{,}4$ | $15{,}8 \pm 1{,}8$ | $13{,}8 \pm 1{,}6$ | 25 $\ll$ |
| | | | 50 | $82{,}6 \pm 4{,}9$ | $80{,}2 \pm 3{,}6$ | $80{,}4 \pm 3{,}2$ | $77{,}3 \pm 2{,}0$ | $10{,}9 \pm 2{,}0$ | $19{,}3 \pm 7{,}1$ | $25{,}8 \pm 2{,}4$ $\ll$ | 50 $\ll$ |
| | | | 100 | $80{,}0 \pm 5{,}3$ | $76{,}6 \pm 4{,}6$ | $74{,}8 \pm 3{,}0$ | $70{,}0 \pm 3{,}8$ $\ll$ | $10{,}8 \pm 2{,}7$ | $24{,}8 \pm 5{,}5$ $<$ | $45{,}5 \pm 4{,}7$ $\ll$ | 100 $\ll$ |
| | | | 200 | $78{,}5 \pm 5{,}5$ | $76{,}2 \pm 4{,}7$ | $67{,}4 \pm 2{,}5$ $<$ | $64{,}8 \pm 2{,}1$ $<$ | $10{,}1 \pm 2{,}3$ | $28{,}3 \pm 11{,}8$ | $97{,}9 \pm 8{,}5$ $\ll$ | 200 $\ll$ |
| 2 | 5 | 0 | 25 | $89{,}4 \pm 2{,}8$ | $89{,}8 \pm 1{,}9$ | $87{,}7 \pm 2{,}6$ | $84{,}3 \pm 2{,}2$ $<$ | $6{,}5 \pm 1{,}8$ | $7{,}5 \pm 2{,}2$ | $12{,}3 \pm 2{,}2$ $<$ | 25 $\ll$ |
| | | | 50 | $88{,}0 \pm 3{,}2$ | $85{,}9 \pm 2{,}1$ | $85{,}6 \pm 1{,}4$ | $79{,}8 \pm 1{,}4$ $\ll$ | $7{,}1 \pm 2{,}3$ | $11{,}8 \pm 4{,}6$ | $21{,}3 \pm 3{,}8$ $<$ | 50 $\ll$ |
| | | | 100 | $87{,}4 \pm 3{,}2$ | $83{,}9 \pm 3{,}8$ | $78{,}1 \pm 1{,}1$ $<$ | $73{,}3 \pm 2{,}9$ $\ll$ | $7{,}8 \pm 1{,}2$ | $12{,}8 \pm 7{,}3$ | $45{,}8 \pm 2{,}5$ $\ll$ | 100 $\ll$ |
| | | | 200 | $85{,}7 \pm 3{,}0$ | $80{,}4 \pm 3{,}5$ | $73{,}2 \pm 3{,}4$ $\ll$ | $69{,}7 \pm 1{,}7$ $\ll$ | $8{,}2 \pm 2{,}4$ | $34{,}0 \pm 18{,}1$ | $92{,}3 \pm 6{,}3$ $\ll$ | 200 $\ll$ |
| | | 0,1 | 25 | $90{,}1 \pm 2{,}6$ | $89{,}8 \pm 2{,}6$ | $87{,}6 \pm 2{,}6$ | $84{,}2 \pm 2{,}8$ $\ll$ | $6{,}7 \pm 1{,}7$ | $8{,}2 \pm 3{,}2$ | $11{,}7 \pm 2{,}2$ $\ll$ | 25 $\ll$ |
| | | | 50 | $88{,}9 \pm 2{,}6$ | $86{,}5 \pm 2{,}8$ | $84{,}1 \pm 2{,}0$ $<$ | $79{,}2 \pm 1{,}6$ $\ll$ | $6{,}4 \pm 1{,}3$ | $15{,}1 \pm 4{,}8$ $\ll$ | $20{,}8 \pm 3{,}3$ $\ll$ | 50 $\ll$ |
| | | | 100 | $87{,}0 \pm 1{,}8$ | $84{,}9 \pm 4{,}2$ | $79{,}0 \pm 2{,}9$ | $73{,}2 \pm 2{,}2$ $\ll$ | $7{,}5 \pm 1{,}4$ | $15{,}3 \pm 6{,}3$ | $47{,}4 \pm 4{,}9$ $\ll$ | 100 $\ll$ |
| | | | 200 | $85{,}8 \pm 3{,}9$ | $80{,}4 \pm 2{,}3$ | $72{,}5 \pm 2{,}5$ $<$ | $69{,}6 \pm 1{,}6$ $\ll$ | $7{,}6 \pm 2{,}7$ | $24{,}9 \pm 8{,}8$ $<$ | $96{,}9 \pm 9{,}1$ $\ll$ | 200 $\ll$ |
| | 10 | 0 | 25 | $83{,}4 \pm 2{,}0$ | $81{,}9 \pm 2{,}9$ | $81{,}5 \pm 3{,}1$ | $80{,}3 \pm 2{,}9$ $<$ | $9{,}9 \pm 1{,}4$ | $12{,}1 \pm 2{,}4$ | $13{,}7 \pm 1{,}9$ | 25 $\ll$ |
| | | | 50 | $83{,}7 \pm 2{,}8$ | $80{,}4 \pm 3{,}1$ | $79{,}9 \pm 1{,}7$ | $74{,}8 \pm 1{,}8$ $<$ | $10{,}5 \pm 1{,}8$ | $13{,}0 \pm 5{,}5$ | $22{,}4 \pm 3{,}0$ $<$ | 50 $\ll$ |
| | | | 100 | $80{,}3 \pm 3{,}1$ | $77{,}2 \pm 4{,}9$ $\ll$ | $74{,}2 \pm 2{,}4$ $<$ | $69{,}0 \pm 2{,}7$ $\ll$ | $10{,}9 \pm 0{,}9$ | $17{,}2 \pm 4{,}4$ | $47{,}3 \pm 4{,}5$ $\ll$ | 100 $\ll$ |
| | | | 200 | $78{,}2 \pm 5{,}4$ | $73{,}5 \pm 5{,}6$ | $67{,}9 \pm 2{,}7$ $\ll$ | $64{,}9 \pm 3{,}3$ $\ll$ | $11{,}1 \pm 2{,}0$ | $36{,}2 \pm 25{,}3$ | $95{,}9 \pm 5{,}1$ $\ll$ | 200 $\ll$ |
| | | 0,1 | 25 | $82{,}8 \pm 3{,}5$ | $81{,}4 \pm 3{,}3$ | $81{,}9 \pm 2{,}6$ | $80{,}7 \pm 2{,}2$ | $10{,}3 \pm 1{,}3$ | $12{,}1 \pm 3{,}1$ | $14{,}0 \pm 1{,}7$ | 25 $\ll$ |
| | | | 50 | $80{,}1 \pm 3{,}7$ | $78{,}9 \pm 3{,}2$ | $77{,}5 \pm 3{,}4$ | $74{,}4 \pm 2{,}1$ | $9{,}3 \pm 1{,}8$ | $17{,}3 \pm 6{,}2$ | $23{,}9 \pm 3{,}0$ $\ll$ | 50 $\ll$ |
| | | | 100 | $80{,}2 \pm 4{,}9$ | $78{,}3 \pm 2{,}8$ | $72{,}8 \pm 3{,}3$ | $69{,}0 \pm 2{,}6$ $<$ | $9{,}9 \pm 1{,}3$ | $19{,}5 \pm 9{,}7$ | $49{,}1 \pm 5{,}1$ $\ll$ | 100 $\ll$ |
| | | | 200 | $79{,}9 \pm 3{,}5$ | $74{,}0 \pm 4{,}6$ $<$ | $65{,}8 \pm 2{,}7$ $\ll$ | $65{,}5 \pm 3{,}0$ $\ll$ | $10{,}5 \pm 1{,}0$ | $31{,}2 \pm 29{,}8$ | $98{,}0 \pm 6{,}7$ $\ll$ | 200 $\ll$ |

$l$-regla de clasificación; $r$-número de atributos relevantes; $\sigma$-ruido Gaussiano;BD-Búsqueda Dispersa; AG-Algoritmo Genético; FSSEBNA-Feature Subset Selection by Estimation of Bayesian Network Algorithm; base-clasificador sin selección de atributos; $>$ y $\gg$ mejora significativa en un 95 y 99 %; $<$ y $\ll$ empeora significativamente.

Tabla 4.5: Comparativa entre la metaheurística BD y las estrategias AG y FSSEBNA con el clasificador IB1. Porcentaje de acierto y reducción en conjuntos de datos artificiales.

| conjuntos de datos | | | | IB1 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | % aciertos | | | | # atributos | | | |
| l | r | $\sigma$ | d | BD | AG | FSSEBNA | base | BD | AG | FSSEBNA | base |
| 1 | 5 | 0 | 25 | $78{,}0 \pm 6{,}1$ | $79{,}8 \pm 6{,}0$ | $80{,}8 \pm 3{,}4$ | $69{,}7 \pm 2{,}2$ | $5{,}0 \pm 1{,}3$ | $7{,}5 \pm 1{,}8$ | $8{,}3 \pm 1{,}8$ | 25 $\ll$ |
| | | | 50 | $73{,}4 \pm 6{,}6$ | $77{,}3 \pm 5{,}7$ | $70{,}0 \pm 2{,}6$ | $61{,}9 \pm 3{,}3$ | $3{,}9 \pm 1{,}5$ | $8{,}3 \pm 4{,}1$ | $19{,}7 \pm 2{,}7$ $\ll$ | 50 $\ll$ |
| | | | 100 | $69{,}6 \pm 6{,}2$ | $71{,}8 \pm 6{,}2$ | $62{,}3 \pm 2{,}7$ | $55{,}4 \pm 1{,}6$ $<$ | $3{,}3 \pm 0{,}7$ | $10{,}9 \pm 8{,}5$ | $47{,}2 \pm 5{,}5$ $\ll$ | 100 $\ll$ |
| | | | 200 | $64{,}6 \pm 7{,}7$ | $64{,}2 \pm 4{,}8$ | $57{,}8 \pm 3{,}5$ | $54{,}6 \pm 2{,}2$ | $3{,}4 \pm 1{,}6$ | $31{,}1 \pm 28{,}3$ | $100{,}6 \pm 6{,}2$ $\ll$ | 200 $\ll$ |
| | | 0,1 | 25 | $81{,}2 \pm 8{,}2$ | $83{,}2 \pm 3{,}6$ | $80{,}4 \pm 3{,}5$ | $69{,}9 \pm 2{,}4$ | $5{,}3 \pm 0{,}9$ | $7{,}1 \pm 2{,}1$ | $7{,}8 \pm 1{,}6$ | 25 $\ll$ |
| | | | 50 | $73{,}8 \pm 7{,}1$ | $77{,}9 \pm 7{,}7$ | $67{,}2 \pm 3{,}1$ | $63{,}6 \pm 3{,}5$ | $4{,}7 \pm 1{,}1$ | $7{,}9 \pm 4{,}0$ | $22{,}2 \pm 2{,}5$ $\ll$ | 50 $\ll$ |
| | | | 100 | $69{,}4 \pm 8{,}2$ | $73{,}3 \pm 6{,}9$ | $61{,}9 \pm 3{,}3$ | $55{,}3 \pm 1{,}5$ | $3{,}9 \pm 1{,}7$ | $12{,}1 \pm 7{,}5$ | $44{,}3 \pm 5{,}8$ $\ll$ | 100 $\ll$ |
| | | | 200 | $69{,}0 \pm 5{,}0$ | $65{,}4 \pm 5{,}9$ | $56{,}3 \pm 2{,}6$ | $54{,}8 \pm 2{,}7$ $<$ | $4{,}0 \pm 1{,}2$ | $27{,}4 \pm 18{,}1$ | $97{,}4 \pm 5{,}7$ $\ll$ | 200 $\ll$ |
| | 10 | 0 | 25 | $60{,}5 \pm 6{,}3$ | $72{,}4 \pm 4{,}4$ | $72{,}8 \pm 2{,}8$ | $68{,}0 \pm 2{,}5$ | $4{,}9 \pm 2{,}0$ | $11{,}5 \pm 2{,}4$ $<$ | $13{,}2 \pm 2{,}3$ $\ll$ | 25 $\ll$ |
| | | | 50 | $58{,}7 \pm 8{,}4$ | $68{,}9 \pm 2{,}7$ | $66{,}5 \pm 2{,}9$ | $62{,}2 \pm 2{,}6$ | $3{,}8 \pm 1{,}9$ | $10{,}4 \pm 4{,}4$ | $24{,}9 \pm 4{,}0$ $\ll$ | 50 $\ll$ |
| | | | 100 | $54{,}5 \pm 5{,}9$ | $63{,}6 \pm 4{,}0$ | $60{,}0 \pm 2{,}9$ | $59{,}8 \pm 2{,}8$ | $2{,}2 \pm 1{,}1$ | $24{,}7 \pm 13{,}7$ $<$ | $52{,}1 \pm 4{,}7$ $\ll$ | 100 $\ll$ |
| | | | 200 | $52{,}2 \pm 4{,}3$ | $61{,}6 \pm 4{,}9$ | $56{,}8 \pm 5{,}2$ | $53{,}6 \pm 2{,}3$ | $2{,}2 \pm 1{,}5$ | $43{,}7 \pm 31{,}3$ | $96{,}4 \pm 6{,}9$ $\ll$ | 200 $\ll$ |
| | | 0,1 | 25 | $63{,}3 \pm 5{,}3$ | $72{,}2 \pm 3{,}3$ | $71{,}6 \pm 2{,}3$ | $68{,}9 \pm 3{,}4$ | $4{,}6 \pm 1{,}5$ | $13{,}3 \pm 2{,}4$ $<$ | $13{,}6 \pm 2{,}8$ $<$ | 25 $\ll$ |
| | | | 50 | $55{,}2 \pm 3{,}6$ | $66{,}4 \pm 5{,}5$ $>$ | $65{,}4 \pm 2{,}7$ $>$ | $61{,}0 \pm 1{,}6$ $>$ | $2{,}4 \pm 1{,}5$ | $11{,}9 \pm 5{,}0$ $\ll$ | $23{,}9 \pm 3{,}5$ $\ll$ | 50 $\ll$ |
| | | | 100 | $54{,}3 \pm 7{,}0$ | $65{,}7 \pm 4{,}1$ | $58{,}7 \pm 2{,}7$ | $58{,}5 \pm 2{,}2$ | $2{,}3 \pm 2{,}2$ | $17{,}2 \pm 11{,}1$ | $50{,}3 \pm 5{,}7$ $\ll$ | 100 $\ll$ |
| | | | 200 | $52{,}5 \pm 3{,}6$ | $61{,}8 \pm 5{,}9$ $\gg$ | $56{,}4 \pm 2{,}4$ | $53{,}5 \pm 2{,}7$ | $1{,}5 \pm 0{,}8$ | $38{,}7 \pm 35{,}8$ | $97{,}4 \pm 5{,}7$ $\ll$ | 200 $\ll$ |
| 2 | 5 | 0 | 25 | $77{,}1 \pm 6{,}9$ | $81{,}4 \pm 4{,}0$ | $79{,}4 \pm 4{,}5$ | $70{,}2 \pm 2{,}3$ | $4{,}4 \pm 1{,}5$ | $7{,}3 \pm 2{,}9$ | $8{,}6 \pm 2{,}1$ | 25 $\ll$ |
| | | | 50 | $75{,}4 \pm 5{,}8$ | $79{,}3 \pm 4{,}5$ | $70{,}3 \pm 3{,}9$ | $63{,}0 \pm 1{,}8$ $<$ | $3{,}0 \pm 0{,}7$ | $7{,}2 \pm 2{,}6$ | $20{,}1 \pm 4{,}7$ $\ll$ | 50 $\ll$ |
| | | | 100 | $74{,}6 \pm 6{,}4$ | $73{,}4 \pm 4{,}1$ | $63{,}4 \pm 2{,}5$ | $60{,}5 \pm 2{,}7$ | $4{,}3 \pm 1{,}4$ | $9{,}0 \pm 4{,}2$ | $50{,}2 \pm 6{,}1$ $\ll$ | 100 $\ll$ |
| | | | 200 | $70{,}9 \pm 6{,}3$ | $66{,}2 \pm 4{,}5$ | $59{,}0 \pm 3{,}6$ | $56{,}1 \pm 2{,}1$ | $4{,}4 \pm 2{,}2$ | $32{,}1 \pm 27{,}5$ | $95{,}7 \pm 7{,}3$ $\ll$ | 200 $\ll$ |
| | | 0,1 | 25 | $78{,}5 \pm 4{,}3$ | $82{,}6 \pm 5{,}0$ | $80{,}2 \pm 3{,}2$ | $70{,}4 \pm 1{,}8$ | $4{,}3 \pm 0{,}9$ | $7{,}4 \pm 2{,}2$ | $8{,}6 \pm 1{,}3$ $\ll$ | 25 $\ll$ |
| | | | 50 | $74{,}8 \pm 7{,}9$ | $79{,}8 \pm 3{,}7$ | $69{,}2 \pm 3{,}4$ | $62{,}4 \pm 2{,}6$ | $4{,}0 \pm 1{,}1$ | $8{,}2 \pm 2{,}7$ | $19{,}4 \pm 2{,}9$ $\ll$ | 50 $\ll$ |
| | | | 100 | $75{,}0 \pm 4{,}4$ | $74{,}5 \pm 8{,}1$ | $62{,}6 \pm 3{,}4$ $<$ | $59{,}2 \pm 3{,}0$ $<$ | $3{,}8 \pm 1{,}3$ | $11{,}9 \pm 10{,}8$ | $48{,}8 \pm 4{,}4$ $\ll$ | 100 $\ll$ |
| | | | 200 | $70{,}9 \pm 4{,}7$ | $67{,}8 \pm 6{,}0$ | $58{,}4 \pm 3{,}3$ $<$ | $55{,}8 \pm 2{,}0$ $<$ | $3{,}5 \pm 1{,}3$ | $28{,}6 \pm 29{,}3$ | $102{,}1 \pm 5{,}9$ $\ll$ | 200 $\ll$ |
| | 10 | 0 | 25 | $60{,}2 \pm 9{,}8$ | $69{,}0 \pm 3{,}3$ | $68{,}8 \pm 3{,}9$ | $65{,}1 \pm 2{,}7$ | $4{,}2 \pm 2{,}7$ | $11{,}2 \pm 2{,}7$ $<$ | $12{,}1 \pm 0{,}9$ $\ll$ | 25 $\ll$ |
| | | | 50 | $54{,}5 \pm 5{,}5$ | $66{,}4 \pm 4{,}0$ | $63{,}2 \pm 3{,}8$ | $58{,}8 \pm 2{,}8$ | $2{,}9 \pm 2{,}2$ | $13{,}1 \pm 7{,}1$ | $25{,}3 \pm 5{,}2$ $\ll$ | 50 $\ll$ |
| | | | 100 | $53{,}8 \pm 4{,}8$ | $63{,}8 \pm 6{,}0$ | $60{,}1 \pm 2{,}7$ | $59{,}0 \pm 2{,}7$ | $2{,}2 \pm 1{,}9$ | $23{,}6 \pm 11{,}5$ $<$ | $49{,}8 \pm 3{,}0$ $\ll$ | 100 $\ll$ |
| | | | 200 | $51{,}2 \pm 3{,}5$ | $57{,}8 \pm 4{,}8$ | $55{,}8 \pm 3{,}8$ | $55{,}1 \pm 2{,}7$ | $1{,}4 \pm 1{,}3$ | $60{,}5 \pm 36{,}2$ | $98{,}7 \pm 6{,}3$ $\ll$ | 200 $\ll$ |
| | | 0,1 | 25 | $60{,}6 \pm 4{,}3$ | $67{,}8 \pm 3{,}8$ | $69{,}8 \pm 3{,}4$ $>$ | $65{,}8 \pm 2{,}2$ | $2{,}8 \pm 1{,}9$ | $10{,}3 \pm 3{,}4$ $<$ | $11{,}8 \pm 2{,}3$ $\ll$ | 25 $\ll$ |
| | | | 50 | $56{,}6 \pm 3{,}9$ | $68{,}1 \pm 3{,}1$ | $62{,}6 \pm 5{,}5$ | $59{,}4 \pm 3{,}7$ | $2{,}5 \pm 1{,}4$ | $12{,}8 \pm 5{,}7$ | $25{,}6 \pm 3{,}1$ $\ll$ | 50 $\ll$ |
| | | | 100 | $52{,}5 \pm 7{,}2$ | $64{,}4 \pm 4{,}8$ | $58{,}5 \pm 2{,}7$ | $58{,}9 \pm 2{,}8$ | $2{,}2 \pm 1{,}4$ | $18{,}7 \pm 10{,}3$ $<$ | $49{,}5 \pm 5{,}3$ $\ll$ | 100 $\ll$ |
| | | | 200 | $53{,}1 \pm 6{,}1$ | $62{,}8 \pm 5{,}3$ | $55{,}8 \pm 2{,}2$ | $54{,}8 \pm 2{,}0$ | $1{,}8 \pm 1{,}1$ | $23{,}6 \pm 27{,}6$ $\ll$ | $97{,}7 \pm 4{,}9$ $\ll$ | 200 $\ll$ |

$l$-regla de clasificación; $r$-número de atributos relevantes; $\sigma$-ruido Gaussiano;BD-Búsqueda Dispersa; AG-Algoritmo Genético; FSSEBNA-Feature Subset Selection by Estimation of Bayesian Network Algorithm; base-clasificador sin selección de atributos; $>$ y $\gg$ mejora significativa en un 95 y 99 %; $<$ y $\ll$ empeora significativamente.

Tabla 4.6: Comparativa entre la metaheurística BD y las estrategias AG y FSSEBNA con el clasificador J48. Porcentaje de acierto y reducción en conjuntos de datos artificiales.

| conjuntos de datos | | | | J48 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | % aciertos | | | | # atributos | | | |
| l | r | $\sigma$ | d | BD | AG | FSSEBNA | base | BD | AG | FSSEBNA | base |
| 1 | 5 | 0 | 25 | $76{,}4 \pm 2{,}9$ | $77{,}2 \pm 3{,}8$ | $77{,}4 \pm 3{,}0$ | $79{,}1 \pm 2{,}8$ | $3{,}8 \pm 0{,}6$ | $8{,}5 \pm 2{,}4$ $<$ | $10{,}1 \pm 2{,}5$ $\ll$ | 25 $\ll$ |
| | | | 50 | $75{,}8 \pm 2{,}7$ | $74{,}0 \pm 3{,}3$ | $75{,}4 \pm 3{,}3$ | $75{,}8 \pm 2{,}1$ | $3{,}9 \pm 0{,}7$ | $14{,}6 \pm 5{,}6$ $\ll$ | $23{,}1 \pm 2{,}9$ $\ll$ | 50 $\ll$ |
| | | | 100 | $76{,}5 \pm 2{,}9$ | $74{,}4 \pm 6{,}9$ | $71{,}4 \pm 3{,}7$ | $72{,}9 \pm 2{,}9$ | $4{,}2 \pm 1{,}0$ | $30{,}3 \pm 13{,}3$ $<$ | $47{,}9 \pm 6{,}2$ $\ll$ | 100 $\ll$ |
| | | | 200 | $77{,}2 \pm 2{,}5$ | $74{,}1 \pm 4{,}1$ | $71{,}3 \pm 3{,}4$ | $70{,}6 \pm 2{,}4$ | $5{,}0 \pm 1{,}4$ | $41{,}4 \pm 23{,}4$ | $98{,}3 \pm 6{,}5$ $\ll$ | 200 $\ll$ |
| | | 0,1 | 25 | $77{,}5 \pm 3{,}5$ | $76{,}0 \pm 4{,}3$ | $75{,}0 \pm 3{,}7$ | $76{,}4 \pm 3{,}1$ | $4{,}0 \pm 0{,}8$ | $8{,}7 \pm 1{,}7$ $\ll$ | $10{,}4 \pm 2{,}8$ $\ll$ | 25 $\ll$ |
| | | | 50 | $78{,}6 \pm 3{,}8$ | $77{,}8 \pm 4{,}0$ | $74{,}6 \pm 5{,}0$ | $75{,}4 \pm 2{,}3$ | $5{,}5 \pm 1{,}7$ | $13{,}7 \pm 5{,}2$ $<$ | $23{,}0 \pm 2{,}1$ $\ll$ | 50 $\ll$ |
| | | | 100 | $77{,}6 \pm 2{,}2$ | $75{,}0 \pm 3{,}2$ | $71{,}9 \pm 4{,}7$ | $72{,}3 \pm 2{,}2$ $<$ | $4{,}5 \pm 0{,}8$ | $21{,}7 \pm 13{,}5$ | $49{,}8 \pm 5{,}1$ $\ll$ | 100 $\ll$ |
| | | | 200 | $78{,}3 \pm 3{,}4$ | $72{,}5 \pm 3{,}9$ | $70{,}1 \pm 3{,}4$ $<$ | $70{,}2 \pm 3{,}3$ $<$ | $5{,}3 \pm 1{,}6$ | $55{,}2 \pm 29{,}9$ | $98{,}3 \pm 7{,}3$ $\ll$ | 200 $\ll$ |
| | 10 | 0 | 25 | $67{,}2 \pm 5{,}1$ | $67{,}3 \pm 3{,}4$ | $67{,}6 \pm 3{,}6$ | $67{,}4 \pm 2{,}2$ | $6{,}2 \pm 2{,}1$ | $11{,}9 \pm 2{,}8$ | $12{,}4 \pm 1{,}8$ $<$ | 25 $\ll$ |
| | | | 50 | $66{,}5 \pm 5{,}4$ | $66{,}7 \pm 2{,}7$ | $67{,}5 \pm 4{,}4$ | $65{,}2 \pm 2{,}7$ | $5{,}9 \pm 1{,}4$ | $18{,}0 \pm 4{,}2$ $\ll$ | $23{,}5 \pm 3{,}2$ $\ll$ | 50 $\ll$ |
| | | | 100 | $68{,}2 \pm 4{,}9$ | $64{,}2 \pm 5{,}7$ | $64{,}2 \pm 2{,}3$ | $63{,}2 \pm 2{,}3$ | $6{,}3 \pm 1{,}3$ | $28{,}3 \pm 14{,}2$ | $51{,}6 \pm 3{,}7$ $\ll$ | 100 $\ll$ |
| | | | 200 | $69{,}2 \pm 3{,}0$ | $62{,}6 \pm 4{,}0$ | $61{,}0 \pm 3{,}3$ | $61{,}9 \pm 2{,}8$ | $6{,}0 \pm 1{,}5$ | $59{,}0 \pm 33{,}6$ | $96{,}0 \pm 6{,}7$ $\ll$ | 200 $\ll$ |
| | | 0,1 | 25 | $67{,}1 \pm 2{,}4$ | $66{,}4 \pm 2{,}6$ | $67{,}2 \pm 3{,}4$ | $66{,}5 \pm 2{,}2$ | $6{,}1 \pm 1{,}3$ | $10{,}5 \pm 2{,}2$ | $12{,}5 \pm 2{,}7$ $<$ | 25 $\ll$ |
| | | | 50 | $67{,}4 \pm 3{,}7$ | $63{,}7 \pm 4{,}3$ | $64{,}1 \pm 4{,}2$ | $63{,}0 \pm 4{,}0$ | $5{,}6 \pm 2{,}1$ | $18{,}9 \pm 7{,}8$ $<$ | $22{,}0 \pm 4{,}6$ $\ll$ | 50 $\ll$ |
| | | | 100 | $67{,}4 \pm 3{,}3$ | $63{,}0 \pm 3{,}0$ | $62{,}8 \pm 3{,}2$ | $63{,}2 \pm 2{,}8$ | $5{,}5 \pm 1{,}6$ | $28{,}9 \pm 17{,}9$ $<$ | $50{,}3 \pm 3{,}5$ $\ll$ | 100 $\ll$ |
| | | | 200 | $67{,}4 \pm 4{,}5$ | $62{,}4 \pm 4{,}7$ | $61{,}4 \pm 4{,}5$ | $60{,}9 \pm 3{,}4$ | $5{,}4 \pm 1{,}8$ | $58{,}9 \pm 30{,}3$ $\ll$ | $100{,}0 \pm 10{,}8$ $\ll$ | 200 $\ll$ |
| 2 | 5 | 0 | 25 | $81{,}1 \pm 1{,}9$ | $81{,}1 \pm 2{,}8$ | $79{,}0 \pm 4{,}1$ | $78{,}5 \pm 2{,}9$ | $4{,}5 \pm 1{,}4$ | $8{,}2 \pm 2{,}6$ | $11{,}8 \pm 2{,}0$ $<$ | 25 $\ll$ |
| | | | 50 | $81{,}5 \pm 2{,}1$ | $79{,}2 \pm 3{,}4$ | $79{,}0 \pm 2{,}5$ | $77{,}3 \pm 3{,}6$ | $4{,}4 \pm 0{,}8$ | $18{,}6 \pm 7{,}4$ | $24{,}0 \pm 4{,}5$ $\ll$ | 50 $\ll$ |
| | | | 100 | $81{,}7 \pm 2{,}1$ | $78{,}6 \pm 3{,}6$ | $75{,}6 \pm 3{,}7$ | $75{,}4 \pm 3{,}0$ $<$ | $5{,}5 \pm 1{,}6$ | $28{,}1 \pm 16{,}8$ | $49{,}8 \pm 4{,}9$ $\ll$ | 100 $\ll$ |
| | | | 200 | $81{,}8 \pm 2{,}4$ | $76{,}2 \pm 4{,}6$ $<$ | $74{,}4 \pm 4{,}4$ | $73{,}2 \pm 3{,}8$ $<$ | $4{,}8 \pm 1{,}0$ | $53{,}7 \pm 24{,}8$ | $98{,}4 \pm 7{,}2$ $\ll$ | 200 $\ll$ |
| | | 0,1 | 25 | $80{,}9 \pm 2{,}8$ | $79{,}2 \pm 3{,}7$ | $78{,}2 \pm 3{,}3$ $\ll$ | $76{,}8 \pm 4{,}5$ | $4{,}5 \pm 1{,}2$ | $9{,}1 \pm 4{,}5$ | $11{,}8 \pm 2{,}9$ $<$ | 25 $\ll$ |
| | | | 50 | $81{,}5 \pm 3{,}8$ | $78{,}9 \pm 2{,}4$ | $78{,}7 \pm 2{,}0$ | $78{,}9 \pm 3{,}3$ | $4{,}0 \pm 0{,}8$ | $17{,}1 \pm 6{,}0$ | $22{,}2 \pm 2{,}8$ $\ll$ | 50 $\ll$ |
| | | | 100 | $81{,}2 \pm 2{,}1$ | $77{,}1 \pm 4{,}0$ | $76{,}4 \pm 2{,}8$ | $75{,}5 \pm 3{,}5$ $\ll$ | $4{,}2 \pm 1{,}2$ | $31{,}4 \pm 17{,}4$ | $48{,}0 \pm 3{,}7$ $\ll$ | 100 $\ll$ |
| | | | 200 | $79{,}5 \pm 2{,}9$ | $75{,}6 \pm 2{,}3$ | $75{,}0 \pm 2{,}7$ $\ll$ | $72{,}8 \pm 3{,}8$ | $4{,}5 \pm 1{,}5$ | $48{,}5 \pm 29{,}7$ | $101{,}6 \pm 7{,}0$ $\ll$ | 200 $\ll$ |
| | 10 | 0 | 25 | $69{,}5 \pm 2{,}5$ | $67{,}7 \pm 3{,}5$ | $68{,}1 \pm 2{,}7$ | $67{,}4 \pm 3{,}8$ | $5{,}8 \pm 1{,}8$ | $11{,}6 \pm 2{,}5$ $<$ | $12{,}8 \pm 2{,}1$ $<$ | 25 $\ll$ |
| | | | 50 | $69{,}7 \pm 2{,}6$ | $67{,}7 \pm 4{,}5$ | $64{,}6 \pm 3{,}6$ | $66{,}3 \pm 3{,}5$ | $5{,}9 \pm 1{,}8$ | $15{,}0 \pm 5{,}3$ $<$ | $23{,}3 \pm 2{,}4$ $\ll$ | 50 $\ll$ |
| | | | 100 | $69{,}2 \pm 2{,}3$ | $67{,}7 \pm 5{,}5$ | $64{,}0 \pm 3{,}2$ | $64{,}4 \pm 3{,}2$ | $6{,}1 \pm 1{,}7$ | $25{,}5 \pm 17{,}3$ | $49{,}0 \pm 4{,}8$ $\ll$ | 100 $\ll$ |
| | | | 200 | $69{,}1 \pm 2{,}6$ | $63{,}9 \pm 2{,}8$ | $60{,}6 \pm 3{,}9$ | $60{,}8 \pm 2{,}1$ | $6{,}7 \pm 1{,}8$ | $45{,}3 \pm 30{,}8$ | $102{,}9 \pm 7{,}1$ $\ll$ | 200 $\ll$ |
| | | 0,1 | 25 | $67{,}2 \pm 3{,}3$ | $68{,}0 \pm 4{,}0$ | $66{,}0 \pm 2{,}1$ | $66{,}0 \pm 3{,}6$ | $5{,}2 \pm 0{,}9$ | $10{,}2 \pm 2{,}7$ $<$ | $13{,}0 \pm 1{,}6$ $\ll$ | 25 $\ll$ |
| | | | 50 | $66{,}7 \pm 2{,}8$ | $65{,}2 \pm 2{,}9$ | $64{,}3 \pm 3{,}5$ | $64{,}6 \pm 4{,}2$ | $5{,}2 \pm 1{,}7$ | $13{,}6 \pm 5{,}4$ | $24{,}0 \pm 3{,}5$ $\ll$ | 50 $\ll$ |
| | | | 100 | $67{,}7 \pm 3{,}7$ | $64{,}1 \pm 3{,}2$ | $64{,}3 \pm 2{,}8$ | $63{,}5 \pm 3{,}5$ | $5{,}7 \pm 2{,}3$ | $21{,}5 \pm 13{,}0$ | $48{,}2 \pm 3{,}8$ $\ll$ | 100 $\ll$ |
| | | | 200 | $68{,}7 \pm 4{,}3$ | $62{,}0 \pm 5{,}6$ | $59{,}5 \pm 2{,}2$ | $62{,}0 \pm 2{,}5$ | $6{,}6 \pm 2{,}4$ | $45{,}5 \pm 35{,}8$ | $96{,}7 \pm 6{,}1$ $\ll$ | 200 $\ll$ |

$l$-regla de clasificación; $r$-número de atributos relevantes; $\sigma$-ruido Gaussiano;BD-Búsqueda Dispersa; AG-Algoritmo Genético; FSSEBNA-Feature Subset Selection by Estimation of Bayesian Network Algorithm; base-clasificador sin selección de atributos; $>$ y $\gg$ mejora significativa en un 95 y 99 %; $<$ y $\ll$ empeora significativamente.

### Problemas reales

Los resultados en los datos de la UCI usando el clasificador Naïve Bayes podemos verlo en la Tabla 4.7. El identificador del conjunto de datos está en la primera columna, y los resultados obtenidos con la BD están en la segunda y sexta columnas y las del clasificador base en la quinta y novena columnas. En ambos casos la primera columna muestra los resultados en el acierto y la segunda el tamaño de los conjuntos de atributos.

En promedio, el acierto aplicando la BD mejora o empeora dependiendo del conjunto de datos considerado. De todos los resultados, en 11 problemas las diferencias son estadísticamente significativas, de las que 5 corresponden a mejoras con la BD y 6 al clasificador base. La reducción es significativa en todos los casos.

Aplicando el test de Nemenyi sobre todos los resultados del acierto encontramos que las diferencias son estadísticamente significativas al nivel $\alpha = 0{,}1$. Analizando estos resultados en función del tamaño de los datos encontramos que sólo en el caso de problemas pequeños las diferencias son estadísticamente significativas al nivel $\alpha = 0{,}1$. En promedio, los valores obtenidos en ambos casos son muy próximos, siendo del 76,3 % para el caso en que se usa la BD y 76,5 % con el clasificador base.

Los resultados con el IB1 se muestran en la Tabla 4.8. Como en el caso anterior, dependiendo del conjunto de datos el uso de los atributos seleccionados por la BD mejora o empeora el acierto del clasificador. El test F obtiene que las diferencia son estadísticamente significativas en 4 casos y sólo en uno de ellos la diferencia es a favor de la BD.

Las diferencias en el acierto, según el test de Nemenyi aplicado sobre todos los resultados, son estadísticamente significativas al nivel $\alpha = 0{,}05$. En este caso, los valores promedio obtenidos con la BD son peores, siendo de 75,5 % frente a 78,2 % obtenido con el clasificador base. Sin embargo, analizando los resultados en función del tamaño del problema se tiene que las diferencias no son estadísticamente significativas.

La Tabla 4.9 muestra los resultados obtenidos con el clasificador J48. Dependiendo del conjunto de datos considerados, los resultados con BD mejoran

o empeoran los resultados obtenidos con el clasificador base. De todos los resultados, sólo en 4 de ellos las diferencias son estadísticamente significativas a favor del clasificador base, de los cuales 3 de ellas pertenecen a conjuntos de tamaño mediano.

El test de Nemenyi aplicado a todos los resultados indica que las diferencias en el acierto no son estadísticamente significativas. Estudiando los resultados en función del tamaño encontramos que sólo para los conjuntos medianos hay diferencias estadísticamente significativas al nivel $\alpha = 0{,}05$. En este caso el clasificador base obtiene, en promedio, mejores resultados que la BD, siendo de 72,36 % frente a 71,9 % para el caso en que se considera la BD.

La reducción, usando los tres clasificadores, es estadísticamente significativa al nivel $\alpha = 0{,}05$ considerando el test de Nemenyi. Analizando los resultados en función del tamaño, obtenemos que independientemente del tamaño de los datos considerados la diferencia es significativa con $\alpha = 0{,}05$.

Tabla 4.7: Comparativa entre la metaheurística BD y las estrategias AG y FSSEBNA con el clasificador Naïve-Bayes. Porcentaje de acierto y reducción en conjuntos reales.

| cdd | Naïve Bayes | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | % aciertos | | | | # atributos | | | |
| | BD | AG | FSSEBNA | base | BD | AG | FSSEBNA | base |
| pdi | $77{,}8 \pm 1{,}6$ | $77{,}8 \pm 1{,}6$ | $78{,}0 \pm 1{,}9$ | $75{,}1 \pm 1{,}7$ | $4{,}2 \pm 0{,}6$ | $4{,}3 \pm 0{,}7$ | $4{,}1 \pm 0{,}9$ | 8 $\ll$ |
| gla | $56{,}4 \pm 3{,}6$ | $55{,}4 \pm 4{,}8$ | $53{,}5 \pm 5{,}9$ | $50{,}7 \pm 3{,}8$ | $3{,}0 \pm 1{,}5$ | $3{,}5 \pm 1{,}4$ | $4{,}2 \pm 1{,}7$ | 9 $\ll$ |
| bca | $69{,}4 \pm 1{,}9$ | $69{,}9 \pm 2{,}3$ | $70{,}1 \pm 3{,}4$ | $71{,}8 \pm 2{,}6$ | $2{,}2 \pm 1{,}5$ | $3{,}9 \pm 1{,}4$ | $4{,}7 \pm 1{,}3$ | 9 $\ll$ |
| bwi | $96{,}4 \pm 1{,}1$ | $95{,}9 \pm 1{,}3$ | $96{,}9 \pm 1{,}0$ | $97{,}3 \pm 0{,}5$ $>$ | $5{,}0 \pm 1{,}1$ | $4{,}2 \pm 1{,}3$ | $5{,}7 \pm 1{,}1$ | 9 $\ll$ |
| win | $96{,}2 \pm 2{,}7$ | $95{,}4 \pm 2{,}0$ | $95{,}6 \pm 2{,}9$ | $97{,}0 \pm 2{,}5$ | $5{,}6 \pm 1{,}4$ | $5{,}5 \pm 1{,}0$ | $7{,}7 \pm 1{,}8$ | 13 $\ll$ |
| hst | $82{,}1 \pm 4{,}4$ | $82{,}5 \pm 2{,}2$ | $82{,}9 \pm 3{,}4$ | $84{,}4 \pm 2{,}2$ | $5{,}1 \pm 2{,}1$ | $6{,}8 \pm 1{,}2$ | $8{,}1 \pm 1{,}4$ | 13 $\ll$ |
| hhu | $81{,}1 \pm 2{,}4$ | $81{,}9 \pm 2{,}4$ | $81{,}8 \pm 2{,}7$ | $84{,}7 \pm 1{,}6$ $>$ | $4{,}2 \pm 1{,}0$ | $6{,}0 \pm 1{,}6$ | $7{,}2 \pm 1{,}7$ | 13 $\ll$ |
| hcl | $81{,}6 \pm 3{,}5$ | $80{,}1 \pm 3{,}7$ | $82{,}3 \pm 3{,}2$ | $83{,}4 \pm 3{,}2$ | $5{,}0 \pm 1{,}5$ | $6{,}6 \pm 2{,}0$ $\ll$ | $7{,}3 \pm 2{,}1$ | 13 $\ll$ |
| vow | $67{,}5 \pm 3{,}0$ | $67{,}7 \pm 3{,}9$ | $67{,}6 \pm 3{,}1$ | $61{,}4 \pm 2{,}3$ $<$ | $7{,}9 \pm 1{,}2$ | $8{,}0 \pm 1{,}4$ | $7{,}6 \pm 1{,}3$ | 13 $\ll$ |
| cra | $85{,}0 \pm 1{,}8$ | $85{,}5 \pm 1{,}2$ | $85{,}4 \pm 1{,}4$ | $81{,}1 \pm 1{,}4$ | $3{,}3 \pm 1{,}1$ | $5{,}2 \pm 1{,}4$ | $6{,}7 \pm 1{,}9$ $<$ | 15 $\ll$ |
| ptu | $40{,}9 \pm 3{,}0$ | $43{,}5 \pm 2{,}5$ | $43{,}3 \pm 2{,}8$ | $46{,}9 \pm 2{,}5$ $>$ | $7{,}3 \pm 2{,}1$ | $10{,}1 \pm 1{,}6$ | $10{,}2 \pm 1{,}9$ | 17 $\ll$ |
| vot | $94{,}9 \pm 1{,}7$ | $95{,}4 \pm 0{,}8$ | $94{,}5 \pm 1{,}0$ | $90{,}0 \pm 1{,}2$ $\ll$ | $1{,}7 \pm 0{,}7$ | $2{,}8 \pm 1{,}1$ | $4{,}6 \pm 1{,}0$ $\ll$ | 16 $\ll$ |
| lym | $76{,}9 \pm 3{,}6$ | $77{,}7 \pm 3{,}6$ | $80{,}0 \pm 4{,}0$ | $80{,}8 \pm 3{,}5$ | $4{,}7 \pm 1{,}3$ | $7{,}8 \pm 2{,}7$ $\ll$ | $9{,}2 \pm 2{,}3$ | 18 $\ll$ |
| veh | $56{,}3 \pm 2{,}8$ | $59{,}1 \pm 2{,}8$ | $59{,}8 \pm 2{,}9$ | $58{,}7 \pm 2{,}2$ $>$ | $5{,}0 \pm 1{,}7$ | $9{,}5 \pm 1{,}8$ | $9{,}6 \pm 1{,}7$ | 18 $\ll$ |
| hep | $81{,}4 \pm 2{,}5$ | $84{,}0 \pm 3{,}3$ | $83{,}6 \pm 4{,}3$ | $84{,}6 \pm 4{,}9$ | $2{,}3 \pm 1{,}1$ | $7{,}6 \pm 2{,}1$ | $10{,}1 \pm 2{,}3$ $\ll$ | 19 $\ll$ |
| hco | $82{,}9 \pm 2{,}8$ | $83{,}6 \pm 3{,}1$ | $83{,}6 \pm 2{,}2$ | $80{,}1 \pm 2{,}2$ | $4{,}9 \pm 1{,}4$ | $5{,}6 \pm 1{,}7$ | $7{,}9 \pm 2{,}0$ | 22 $\ll$ |
| aut | $60{,}2 \pm 6{,}2$ | $63{,}7 \pm 5{,}4$ | $63{,}5 \pm 4{,}8$ | $59{,}6 \pm 7{,}0$ | $5{,}6 \pm 1{,}5$ | $7{,}6 \pm 1{,}8$ $\ll$ | $10{,}2 \pm 1{,}9$ $<$ | 25 $\ll$ |
| abp | $96{,}6 \pm 0{,}2$ | $96{,}5 \pm 0{,}4$ | $96{,}5 \pm 0{,}4$ | $94{,}4 \pm 0{,}5$ $<$ | $1{,}3 \pm 0{,}5$ | $7{,}1 \pm 3{,}0$ | $12{,}1 \pm 2{,}3$ $\ll$ | 29 $\ll$ |
| ion | $91{,}7 \pm 2{,}7$ | $92{,}6 \pm 2{,}8$ | $91{,}9 \pm 2{,}5$ | $91{,}1 \pm 2{,}5$ | $7{,}1 \pm 1{,}7$ | $12{,}8 \pm 2{,}5$ $\ll$ | $14{,}6 \pm 3{,}0$ $\ll$ | 34 $\ll$ |
| soy | $86{,}1 \pm 1{,}6$ | $89{,}8 \pm 2{,}0$ | $90{,}6 \pm 1{,}3$ $>$ | $91{,}2 \pm 1{,}2$ $>$ | $12{,}8 \pm 2{,}7$ | $21{,}4 \pm 3{,}1$ | $22{,}6 \pm 2{,}8$ $<$ | 35 $\ll$ |
| kvk | $93{,}4 \pm 1{,}4$ | $93{,}5 \pm 0{,}8$ | $93{,}9 \pm 0{,}7$ | $87{,}5 \pm 0{,}9$ $<$ | $4{,}4 \pm 1{,}2$ | $13{,}6 \pm 2{,}9$ $\ll$ | $14{,}7 \pm 1{,}7$ $\ll$ | 36 $\ll$ |
| ann | $88{,}3 \pm 0{,}6$ | $91{,}6 \pm 1{,}3$ $>$ | $92{,}2 \pm 1{,}7$ $\gg$ | $91{,}5 \pm 0{,}6$ $\gg$ | $2{,}7 \pm 0{,}5$ | $18{,}7 \pm 3{,}6$ $\ll$ | $19{,}9 \pm 2{,}9$ $\ll$ | 38 $\ll$ |
| lca | $47{,}5 \pm 11{,}5$ | $44{,}4 \pm 9{,}5$ | $47{,}5 \pm 8{,}4$ | $52{,}5 \pm 8{,}4$ | $2{,}5 \pm 1{,}3$ | $7{,}9 \pm 4{,}2$ | $25{,}8 \pm 3{,}3$ $\ll$ | 56 $\ll$ |
| dnp | $85{,}8 \pm 3{,}4$ | $84{,}2 \pm 6{,}1$ | $87{,}0 \pm 5{,}1$ | $87{,}4 \pm 5{,}6$ | $5{,}6 \pm 1{,}5$ | $22{,}5 \pm 8{,}2$ $\ll$ | $27{,}9 \pm 4{,}1$ $\ll$ | 57 $\ll$ |
| son | $73{,}6 \pm 3{,}3$ | $71{,}3 \pm 4{,}7$ | $75{,}0 \pm 3{,}5$ | $70{,}6 \pm 5{,}5$ | $3{,}3 \pm 1{,}7$ | $16{,}7 \pm 8{,}8$ | $27{,}3 \pm 3{,}7$ $\ll$ | 60 $\ll$ |
| dna | $95{,}4 \pm 1{,}0$ | $95{,}3 \pm 0{,}5$ | $95{,}6 \pm 0{,}3$ | $95{,}3 \pm 0{,}3$ | $14{,}6 \pm 2{,}7$ | $34{,}0 \pm 4{,}4$ $<$ | $34{,}5 \pm 3{,}5$ $\ll$ | 60 $\ll$ |
| spl | $95{,}0 \pm 0{,}7$ | $95{,}7 \pm 0{,}4$ | $95{,}7 \pm 0{,}6$ | $95{,}3 \pm 0{,}6$ | $13{,}8 \pm 3{,}0$ | $35{,}2 \pm 2{,}1$ $\ll$ | $34{,}0 \pm 2{,}9$ $\ll$ | 61 $\ll$ |
| aud | $67{,}4 \pm 3{,}6$ | $67{,}3 \pm 3{,}0$ | $68{,}5 \pm 1{,}3$ | $63{,}5 \pm 2{,}9$ | $7{,}2 \pm 1{,}8$ | $16{,}6 \pm 6{,}7$ | $29{,}7 \pm 2{,}5$ $\ll$ | 69 $\ll$ |
| rat | $71{,}1 \pm 3{,}1$ | $69{,}3 \pm 5{,}8$ | $55{,}4 \pm 5{,}7$ $\ll$ | $52{,}2 \pm 4{,}5$ $\ll$ | $2{,}9 \pm 1{,}1$ | $4{,}1 \pm 2{,}2$ | $67{,}3 \pm 6{,}2$ $\ll$ | 141 $\ll$ |
| arr | $63{,}5 \pm 2{,}4$ | $64{,}6 \pm 1{,}6$ | $65{,}4 \pm 1{,}9$ | $64{,}2 \pm 1{,}6$ | $6{,}6 \pm 2{,}2$ | $74{,}8 \pm 30{,}6$ $<$ | $138{,}4 \pm 10{,}6$ $\ll$ | 279 $\ll$ |

BD-Búsqueda Dispersa; AG-Algoritmos Genético; FSSEBNA-Feature Subset Selection by Estimation of Bayesian Network Algorithm; $>$ y $\gg$ mejora significativa en un 95 y 99 %; $<$ y $\ll$ empeora significativamente.

Tabla 4.8: Comparativa entre la BD y los algoritmos AG y FSSEBNA y el clasificador base IB1 en conjuntos reales.

| cdd | IB1 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | % aciertos | | | | # atributos | | | |
| | BD | AG | FSSEBNA | base | BD | AG | FSSEBNA | base |
| pdi | $67{,}2 \pm 4{,}4$ | $68{,}8 \pm 2{,}2$ | $68{,}1 \pm 2{,}2$ | $70{,}1 \pm 1{,}9$ | $3{,}7 \pm 1{,}8$ | $4{,}2 \pm 0{,}9$ | $4{,}2 \pm 0{,}8$ | 8 $\ll$ |
| gla | $67{,}2 \pm 5{,}6$ | $67{,}6 \pm 4{,}0$ | $68{,}0 \pm 3{,}0$ | $66{,}2 \pm 3{,}6$ | $5{,}3 \pm 0{,}9$ | $4{,}8 \pm 0{,}6$ | $5{,}3 \pm 1{,}5$ | 9 $\ll$ |
| bca | $66{,}2 \pm 9{,}3$ | $62{,}8 \pm 9{,}6$ | $62{,}3 \pm 8{,}6$ | $66{,}0 \pm 3{,}1$ | $2{,}0 \pm 0{,}9$ | $3{,}1 \pm 1{,}4$ | $3{,}6 \pm 1{,}5$ | 9 $\ll$ |
| bwi | $94{,}9 \pm 1{,}0$ | $94{,}7 \pm 1{,}1$ | $94{,}6 \pm 1{,}0$ | $95{,}6 \pm 0{,}9$ | $5{,}5 \pm 1{,}5$ | $4{,}4 \pm 1{,}3$ | $5{,}0 \pm 1{,}1$ | 9 $<$ |
| win | $93{,}7 \pm 1{,}7$ | $94{,}9 \pm 2{,}6$ | $95{,}4 \pm 2{,}0$ | $94{,}8 \pm 2{,}1$ | $4{,}7 \pm 0{,}9$ | $6{,}2 \pm 1{,}8$ | $7{,}5 \pm 1{,}1$ $\ll$ | 13 $\ll$ |
| hst | $71{,}6 \pm 5{,}7$ | $75{,}9 \pm 3{,}7$ | $75{,}6 \pm 3{,}3$ | $77{,}0 \pm 2{,}3$ $>$ | $5{,}2 \pm 1{,}9$ | $5{,}9 \pm 2{,}0$ | $7{,}0 \pm 1{,}2$ | 13 $\ll$ |
| hhu | $75{,}6 \pm 5{,}7$ | $74{,}4 \pm 5{,}4$ | $74{,}3 \pm 4{,}5$ | $78{,}9 \pm 2{,}6$ | $3{,}2 \pm 1{,}5$ | $5{,}4 \pm 2{,}2$ | $5{,}2 \pm 2{,}4$ | 13 $\ll$ |
| hcl | $71{,}7 \pm 5{,}7$ | $74{,}4 \pm 4{,}6$ | $75{,}4 \pm 4{,}0$ | $78{,}0 \pm 2{,}6$ | $5{,}0 \pm 1{,}9$ | $5{,}4 \pm 1{,}6$ | $6{,}8 \pm 1{,}1$ | 13 $\ll$ |
| vow | $93{,}9 \pm 0{,}9$ | $93{,}2 \pm 1{,}1$ | $94{,}4 \pm 1{,}1$ | $93{,}5 \pm 2{,}1$ | $8{,}8 \pm 1{,}6$ | $8{,}6 \pm 1{,}3$ | $9{,}3 \pm 0{,}8$ | 13 $<$ |
| cra | $82{,}5 \pm 6{,}0$ | $81{,}1 \pm 3{,}3$ | $81{,}2 \pm 2{,}2$ | $81{,}0 \pm 1{,}9$ | $1{,}9 \pm 1{,}3$ | $4{,}4 \pm 1{,}6$ $<$ | $6{,}5 \pm 2{,}1$ | 15 $\ll$ |
| ptu | $26{,}7 \pm 9{,}0$ | $31{,}6 \pm 4{,}7$ | $31{,}6 \pm 4{,}8$ | $31{,}9 \pm 4{,}4$ | $4{,}9 \pm 3{,}1$ | $11{,}0 \pm 1{,}9$ | $10{,}2 \pm 1{,}5$ | 17 $\ll$ |
| vot | $94{,}3 \pm 1{,}8$ | $94{,}3 \pm 1{,}9$ | $94{,}0 \pm 1{,}9$ | $92{,}3 \pm 1{,}0$ $\ll$ | $2{,}9 \pm 2{,}0$ | $4{,}7 \pm 1{,}1$ | $5{,}7 \pm 1{,}6$ | 16 $\ll$ |
| lym | $72{,}6 \pm 7{,}2$ | $75{,}3 \pm 5{,}7$ | $77{,}3 \pm 6{,}6$ | $78{,}6 \pm 4{,}8$ | $4{,}5 \pm 2{,}0$ | $8{,}2 \pm 2{,}0$ | $9{,}7 \pm 2{,}2$ | 18 $\ll$ |
| veh | $67{,}0 \pm 2{,}1$ | $68{,}3 \pm 1{,}9$ | $68{,}8 \pm 1{,}7$ | $68{,}7 \pm 2{,}4$ | $9{,}8 \pm 1{,}9$ | $8{,}4 \pm 2{,}5$ | $9{,}4 \pm 1{,}8$ | 18 $\ll$ |
| hep | $79{,}3 \pm 6{,}1$ | $79{,}1 \pm 4{,}2$ | $77{,}4 \pm 6{,}4$ | $79{,}0 \pm 3{,}6$ | $2{,}2 \pm 0{,}9$ | $6{,}9 \pm 2{,}1$ $\ll$ | $8{,}5 \pm 1{,}4$ $\ll$ | 19 $\ll$ |
| hco | $75{,}7 \pm 6{,}0$ | $81{,}4 \pm 3{,}6$ | $81{,}0 \pm 2{,}7$ | $78{,}4 \pm 2{,}2$ | $2{,}5 \pm 1{,}6$ | $6{,}5 \pm 3{,}6$ | $9{,}3 \pm 2{,}4$ | 22 $\ll$ |
| aut | $72{,}3 \pm 2{,}6$ | $70{,}1 \pm 3{,}7$ | $67{,}8 \pm 4{,}5$ | $67{,}5 \pm 5{,}0$ | $3{,}3 \pm 1{,}3$ | $4{,}7 \pm 2{,}7$ | $12{,}3 \pm 3{,}1$ $<$ | 25 $\ll$ |
| abp | $95{,}2 \pm 0{,}9$ | $96{,}4 \pm 0{,}7$ | $96{,}3 \pm 0{,}4$ $\gg$ | $96{,}1 \pm 0{,}3$ $>$ | $2{,}3 \pm 1{,}3$ | $14{,}0 \pm 3{,}4$ $<$ | $17{,}6 \pm 2{,}3$ $\ll$ | 29 $\ll$ |
| ion | $87{,}7 \pm 3{,}1$ | $87{,}2 \pm 3{,}0$ | $88{,}1 \pm 2{,}0$ | $85{,}2 \pm 1{,}9$ | $4{,}1 \pm 1{,}8$ | $7{,}0 \pm 3{,}3$ | $12{,}1 \pm 2{,}6$ $\ll$ | 34 $\ll$ |
| soy | $84{,}5 \pm 4{,}4$ | $87{,}3 \pm 1{,}2$ | $90{,}3 \pm 2{,}3$ | $88{,}6 \pm 1{,}8$ | $12{,}9 \pm 2{,}9$ | $21{,}7 \pm 3{,}8$ | $22{,}0 \pm 1{,}6$ | 35 $\ll$ |
| kvk | $74{,}6 \pm 8{,}8$ | $94{,}2 \pm 2{,}7$ | $95{,}4 \pm 1{,}4$ | $89{,}2 \pm 0{,}6$ | $5{,}5 \pm 2{,}1$ | $17{,}6 \pm 2{,}7$ $\ll$ | $16{,}6 \pm 2{,}1$ $\ll$ | 36 $\ll$ |
| ann | $93{,}4 \pm 1{,}7$ | $95{,}9 \pm 1{,}4$ | $95{,}0 \pm 2{,}0$ | $94{,}2 \pm 0{,}8$ | $6{,}0 \pm 1{,}8$ | $16{,}1 \pm 2{,}9$ $\ll$ | $20{,}1 \pm 3{,}0$ $\ll$ | 38 $\ll$ |
| lca | $38{,}8 \pm 17{,}4$ | $38{,}8 \pm 7{,}1$ | $44{,}4 \pm 13{,}0$ | $46{,}3 \pm 8{,}9$ | $3{,}0 \pm 1{,}9$ | $13{,}4 \pm 7{,}7$ | $23{,}9 \pm 4{,}7$ $\ll$ | 56 $\ll$ |
| dnp | $77{,}4 \pm 11{,}7$ | $71{,}7 \pm 7{,}8$ | $75{,}8 \pm 4{,}2$ | $79{,}2 \pm 6{,}0$ | $3{,}4 \pm 2{,}0$ | $19{,}5 \pm 9{,}4$ | $29{,}0 \pm 2{,}9$ $\ll$ | 57 $\ll$ |
| son | $69{,}7 \pm 4{,}1$ | $79{,}9 \pm 4{,}0$ $>$ | $81{,}9 \pm 2{,}4$ $>$ | $82{,}0 \pm 2{,}6$ $\gg$ | $3{,}8 \pm 2{,}1$ | $20{,}6 \pm 4{,}9$ $\ll$ | $27{,}9 \pm 3{,}5$ $\ll$ | 60 $\ll$ |
| dna | $82{,}3 \pm 5{,}7$ | $80{,}1 \pm 1{,}1$ | $76{,}0 \pm 1{,}1$ | $73{,}6 \pm 0{,}9$ | $5{,}5 \pm 1{,}5$ | $6{,}6 \pm 1{,}6$ | $27{,}4 \pm 3{,}0$ $\ll$ | 60 $\ll$ |
| spl | $78{,}6 \pm 4{,}3$ | $75{,}7 \pm 1{,}9$ | $76{,}7 \pm 1{,}7$ | $73{,}7 \pm 0{,}8$ | $5{,}9 \pm 1{,}5$ | $25{,}7 \pm 7{,}1$ $\ll$ | $28{,}0 \pm 4{,}4$ $\ll$ | 61 $\ll$ |
| aud | $52{,}6 \pm 13{,}1$ | $70{,}6 \pm 5{,}5$ | $70{,}4 \pm 5{,}7$ $>$ | $70{,}6 \pm 2{,}7$ | $6{,}6 \pm 2{,}9$ | $33{,}4 \pm 7{,}9$ $<$ | $35{,}0 \pm 3{,}0$ $\ll$ | 69 $\ll$ |
| rat | $64{,}7 \pm 4{,}4$ | $65{,}4 \pm 5{,}6$ | $64{,}8 \pm 4{,}1$ | $66{,}4 \pm 4{,}8$ | $2{,}2 \pm 1{,}5$ | $26{,}0 \pm 16{,}9$ | $68{,}7 \pm 6{,}2$ $\ll$ | 141 $\ll$ |
| arr | $43{,}1 \pm 5{,}9$ | $55{,}1 \pm 5{,}3$ | $56{,}0 \pm 3{,}3$ | $51{,}8 \pm 4{,}3$ | $3{,}1 \pm 3{,}0$ | $67{,}7 \pm 30{,}1$ $<$ | $136{,}0 \pm 5{,}2$ $\ll$ | 279 $\ll$ |

BD-Búsqueda Dispersa; AG-Algoritmos Genético; FSSEBNA-Feature Subset Selection by Estimation of Bayesian Network Algorithm; $>$ y $\gg$ mejora significativa en un 95 y 99 %; $<$ y $\ll$ empeora significativamente.

Tabla 4.9: Comparativa entre la metaheurística BD y las estrategias AG y FSSEBNA con el clasificador J48. Porcentaje de acierto y reducción en conjuntos reales.

| cdd | J48 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | % aciertos | | | | # atributos | | | |
| | BD | AG | FSSEBNA | base | BD | AG | FSSEBNA | base |
| pdi | $73{,}7 \pm 1{,}5$ | $73{,}9 \pm 2{,}1$ | $73{,}0 \pm 2{,}0$ | $72{,}5 \pm 1{,}7$ | $2{,}3 \pm 1{,}2$ | $2{,}8 \pm 1{,}4$ | $3{,}4 \pm 1{,}0$ $<$ | 8 $\ll$ |
| gla | $65{,}0 \pm 6{,}4$ | $65{,}7 \pm 3{,}8$ | $62{,}1 \pm 5{,}8$ | $63{,}2 \pm 6{,}1$ | $4{,}5 \pm 1{,}6$ | $4{,}2 \pm 0{,}6$ | $4{,}8 \pm 0{,}6$ | 9 $<$ |
| bca | $70{,}0 \pm 2{,}3$ | $71{,}9 \pm 2{,}7$ | $72{,}0 \pm 2{,}7$ | $69{,}6 \pm 1{,}7$ | $1{,}2 \pm 0{,}6$ | $2{,}3 \pm 0{,}9$ | $4{,}2 \pm 1{,}2$ | 9 $\ll$ |
| bwi | $94{,}3 \pm 1{,}3$ | $93{,}9 \pm 1{,}2$ | $94{,}2 \pm 0{,}9$ | $94{,}4 \pm 1{,}0$ | $2{,}7 \pm 1{,}1$ | $3{,}0 \pm 1{,}2$ | $3{,}6 \pm 1{,}3$ | 9 $\ll$ |
| win | $89{,}4 \pm 3{,}4$ | $90{,}6 \pm 2{,}7$ | $91{,}9 \pm 3{,}8$ | $91{,}9 \pm 2{,}7$ | $2{,}5 \pm 0{,}5$ | $2{,}9 \pm 1{,}0$ | $5{,}9 \pm 2{,}1$ $<$ | 13 $\ll$ |
| hst | $74{,}6 \pm 3{,}1$ | $75{,}5 \pm 2{,}4$ | $77{,}3 \pm 4{,}4$ | $78{,}7 \pm 3{,}2$ | $2{,}4 \pm 1{,}2$ | $4{,}3 \pm 1{,}5$ | $6{,}1 \pm 1{,}5$ $\ll$ | 13 $\ll$ |
| hhu | $79{,}4 \pm 3{,}4$ | $78{,}8 \pm 3{,}4$ | $80{,}5 \pm 2{,}3$ | $78{,}6 \pm 3{,}0$ | $1{,}9 \pm 1{,}3$ | $2{,}6 \pm 1{,}7$ | $6{,}3 \pm 1{,}4$ $\ll$ | 13 $\ll$ |
| hcl | $74{,}5 \pm 3{,}0$ | $76{,}2 \pm 3{,}2$ | $77{,}3 \pm 2{,}2$ | $76{,}9 \pm 3{,}2$ | $2{,}4 \pm 1{,}1$ | $4{,}6 \pm 1{,}4$ | $6{,}2 \pm 1{,}3$ | 13 $\ll$ |
| vow | $69{,}2 \pm 2{,}9$ | $69{,}1 \pm 3{,}1$ | $70{,}9 \pm 2{,}4$ | $69{,}9 \pm 2{,}6$ | $7{,}5 \pm 2{,}0$ | $6{,}8 \pm 1{,}2$ | $7{,}1 \pm 1{,}3$ | 13 $<$ |
| cra | $85{,}3 \pm 1{,}8$ | $85{,}4 \pm 1{,}1$ | $85{,}3 \pm 1{,}6$ | $84{,}3 \pm 1{,}5$ | $1{,}7 \pm 1{,}5$ | $6{,}3 \pm 1{,}8$ $<$ | $7{,}6 \pm 1{,}6$ $<$ | 15 $\ll$ |
| ptu | $35{,}8 \pm 1{,}6$ | $38{,}2 \pm 3{,}4$ | $39{,}2 \pm 3{,}4$ | $38{,}5 \pm 4{,}5$ | $5{,}0 \pm 2{,}4$ | $8{,}5 \pm 2{,}1$ | $9{,}1 \pm 1{,}8$ | 17 $\ll$ |
| vot | $95{,}6 \pm 0{,}7$ | $95{,}5 \pm 0{,}7$ | $95{,}4 \pm 0{,}9$ | $95{,}3 \pm 1{,}1$ | $1{,}0 \pm 0{,}0$ | $1{,}6 \pm 1{,}3$ | $8{,}3 \pm 1{,}3$ $\ll$ | 16 $\ll$ |
| lym | $76{,}5 \pm 2{,}9$ | $75{,}4 \pm 4{,}4$ | $74{,}5 \pm 6{,}5$ | $75{,}9 \pm 3{,}4$ | $2{,}7 \pm 1{,}2$ | $3{,}5 \pm 1{,}8$ | $8{,}8 \pm 2{,}9$ $<$ | 18 $\ll$ |
| veh | $66{,}0 \pm 3{,}8$ | $67{,}8 \pm 2{,}5$ | $68{,}7 \pm 1{,}7$ | $69{,}7 \pm 2{,}2$ | $7{,}7 \pm 3{,}3$ | $10{,}6 \pm 1{,}8$ | $11{,}1 \pm 1{,}7$ | 18 $\ll$ |
| hep | $80{,}1 \pm 4{,}6$ | $78{,}1 \pm 4{,}4$ | $78{,}7 \pm 4{,}2$ | $80{,}1 \pm 2{,}9$ | $1{,}4 \pm 0{,}5$ | $3{,}9 \pm 2{,}8$ | $9{,}3 \pm 2{,}7$ $\ll$ | 19 $\ll$ |
| hco | $82{,}3 \pm 3{,}5$ | $85{,}1 \pm 2{,}4$ | $85{,}2 \pm 2{,}0$ | $85{,}4 \pm 2{,}3$ | $2{,}0 \pm 1{,}4$ | $5{,}0 \pm 2{,}0$ | $10{,}0 \pm 2{,}7$ $<$ | 22 $\ll$ |
| aut | $66{,}9 \pm 7{,}0$ | $64{,}8 \pm 5{,}5$ | $63{,}6 \pm 6{,}5$ | $65{,}6 \pm 6{,}1$ | $3{,}9 \pm 0{,}9$ | $7{,}5 \pm 2{,}5$ | $12{,}0 \pm 2{,}6$ $\ll$ | 25 $\ll$ |
| abp | $96{,}4 \pm 0{,}3$ | $97{,}0 \pm 0{,}3$ | $97{,}0 \pm 0{,}3$ | $97{,}1 \pm 0{,}4$ | $1{,}4 \pm 1{,}0$ | $12{,}6 \pm 2{,}9$ $\ll$ | $14{,}6 \pm 2{,}5$ $\ll$ | 29 $\ll$ |
| ion | $85{,}6 \pm 2{,}7$ | $88{,}3 \pm 2{,}1$ | $87{,}8 \pm 2{,}3$ | $87{,}5 \pm 2{,}4$ | $2{,}4 \pm 0{,}8$ | $9{,}0 \pm 3{,}9$ $<$ | $15{,}9 \pm 3{,}3$ $\ll$ | 34 $\ll$ |
| soy | $81{,}6 \pm 4{,}0$ | $88{,}7 \pm 2{,}8$ $>$ | $88{,}9 \pm 1{,}0$ $\gg$ | $88{,}1 \pm 1{,}5$ $>$ | $11{,}4 \pm 2{,}3$ | $21{,}3 \pm 2{,}7$ $\ll$ | $20{,}8 \pm 2{,}1$ $\ll$ | 35 $\ll$ |
| kvk | $95{,}0 \pm 1{,}9$ | $98{,}2 \pm 0{,}7$ | $98{,}6 \pm 0{,}4$ $>$ | $99{,}1 \pm 0{,}2$ $>$ | $8{,}4 \pm 2{,}8$ | $24{,}7 \pm 3{,}9$ $<$ | $26{,}2 \pm 2{,}3$ $\ll$ | 36 $\ll$ |
| ann | $82{,}7 \pm 1{,}6$ | $89{,}7 \pm 2{,}2$ | $89{,}1 \pm 2{,}9$ | $89{,}5 \pm 1{,}8$ $>$ | $2{,}4 \pm 0{,}5$ | $18{,}2 \pm 4{,}0$ $\ll$ | $19{,}7 \pm 2{,}7$ $\ll$ | 38 $\ll$ |
| lca | $40{,}0 \pm 9{,}9$ | $39{,}4 \pm 5{,}1$ | $42{,}5 \pm 6{,}5$ | $41{,}9 \pm 10{,}2$ | $1{,}5 \pm 0{,}7$ | $6{,}5 \pm 4{,}6$ $<$ | $28{,}5 \pm 4{,}8$ $\ll$ | 56 $\ll$ |
| dnp | $75{,}5 \pm 2{,}0$ | $75{,}1 \pm 3{,}4$ | $73{,}4 \pm 5{,}7$ | $72{,}6 \pm 4{,}9$ | $1{,}1 \pm 0{,}3$ | $3{,}7 \pm 4{,}1$ | $25{,}0 \pm 3{,}9$ $\ll$ | 57 $\ll$ |
| son | $69{,}4 \pm 4{,}8$ | $69{,}1 \pm 5{,}0$ | $68{,}5 \pm 5{,}5$ | $69{,}3 \pm 4{,}6$ | $1{,}6 \pm 0{,}8$ | $19{,}0 \pm 8{,}8$ $<$ | $27{,}2 \pm 4{,}8$ $\ll$ | 60 $\ll$ |
| dna | $92{,}6 \pm 0{,}4$ | $93{,}0 \pm 0{,}7$ | $93{,}0 \pm 0{,}7$ | $92{,}8 \pm 0{,}7$ | $8{,}2 \pm 1{,}5$ | $28{,}9 \pm 4{,}6$ $\ll$ | $32{,}4 \pm 3{,}7$ $\ll$ | 60 $\ll$ |
| spl | $92{,}6 \pm 0{,}5$ | $92{,}5 \pm 0{,}4$ | $92{,}5 \pm 0{,}8$ | $92{,}5 \pm 0{,}5$ | $7{,}7 \pm 1{,}2$ | $33{,}2 \pm 5{,}3$ $\ll$ | $31{,}5 \pm 4{,}4$ $\ll$ | 61 $\ll$ |
| aud | $66{,}9 \pm 3{,}8$ | $71{,}6 \pm 1{,}9$ $\gg$ | $72{,}8 \pm 4{,}0$ $>$ | $74{,}4 \pm 2{,}8$ $\gg$ | $4{,}8 \pm 2{,}1$ | $26{,}0 \pm 5{,}0$ $\ll$ | $35{,}5 \pm 6{,}0$ $\ll$ | 69 $\ll$ |
| rat | $71{,}2 \pm 2{,}8$ | $70{,}7 \pm 4{,}8$ | $70{,}5 \pm 4{,}9$ | $70{,}3 \pm 5{,}7$ | $2{,}3 \pm 1{,}3$ | $36{,}9 \pm 18{,}1$ | $66{,}6 \pm 7{,}8$ $\ll$ | 141 $\ll$ |
| arr | $66{,}9 \pm 3{,}0$ | $64{,}6 \pm 3{,}1$ | $64{,}7 \pm 1{,}8$ | $65{,}1 \pm 2{,}3$ | $5{,}9 \pm 1{,}4$ | $114{,}0 \pm 30{,}9$ $\ll$ | $140{,}7 \pm 6{,}6$ $\ll$ | 279 $\ll$ |

BD-Búsqueda Dispersa; AG-Algoritmos Genético; FSSEBNA-Feature Subset Selection by Estimation of Bayesian Network Algorithm; $>$ y $\gg$ mejora significativa en un 95 y 99 %; $<$ y $\ll$ empeora significativamente.

### 4.4.6 Comparativa del rendimiento de la Búsqueda Dispersa con el AG y el FSSEBNA

A continuación pasamos a comparar el rendimiento de la BD con los algoritmos evolutivos AG y FSSEBNA.

#### Problemas artificiales

Los resultados de las tres estrategias usando el Naïve Bayes pueden verse en la Tabla 4.4. Los resultados muestran que en promedio, la BD es la estrategia que obtiene el mejor resultado, aumentando la diferencia en problemas de mayor tamaño. Además, es la estrategia que encuentra el subconjunto de atributos de menor tamaño en todos los casos excepto en uno. La BD presenta mayores diferencias con el FSSEBNA que con el AG, siendo las diferencias estadísticamente significativas en un mayor número de casos, tanto en el acierto como en el número de atributos encontrados. En el acierto, la BD mejora significativamente en 2 casos al AG y en 13 al FSSEBNA mientras que en la reducción, mejora al AG en 3 casos y al FSSEBNA en todos los casos menos en 5.

El test de Friedman aplicado a todos los resultados nos indica que existen diferencias en el rendimiento de las estrategias estudiadas, y aplicando el test de Nemenyi obtenemos que las diferencias en el acierto con el AG son estadísticamente significativas al nivel $\alpha = 0{,}1$ y en la reducción a $\alpha = 0{,}05$. En ambos casos los valores promedios obtenidos con la BD mejoran los del AG alcanzando un acierto de 84,6 % con la BD frente al 81,9 % del AG. El tamaño promedio del subconjunto encontrado es de 8,9 atributos con la BD y de 18,8 del AG. Las diferencias con el FSSEBNA son estadísticamente significativas al nivel de $\alpha = 0{,}05$ tanto en el acierto como en la reducción. En promedio el acierto con el FSSEBNA es del 78,4 % y los subconjuntos encontrados tienen 44,5 atributos.

Analizando los resultados en función de las reglas de clasificación encontramos que en el caso lineal las diferencias son estadísticamente significativas en el acierto con el FSSEBNA a $\alpha = 0{,}05$. En el caso no lineal estas diferencias son significativas con el FSSEBNA y el AG a $\alpha = 0{,}05$. La reducción

proporcionada por la BD es mayor que la de los otros métodos, siendo esta diferencia significativa a $\alpha = 0{,}05$. Llevando a cabo este análisis en función del número de atributos relevantes obtenemos que con 5 y 10 atributos las diferencias entre la BD y el FSSEBNA en el acierto son estadísticamente significativas a $\alpha = 0{,}05$. La reducción de la BD es mayor en todas las pruebas realizadas. Sin embargo, en los problemas con 5 atributos relevantes las diferencia con el FSSEBNA es significativa a $\alpha = 0{,}05$ mientras que con el AG lo es al nivel de $\alpha = 0{,}1$. En el caso de 10 atributos relevantes la diferencia es significativa con el AG y FSSEBNA con $\alpha = 0{,}05$.

Los resultados con el IB1 están en la Tabla 4.5. Comparando los resultados, observamos que en el acierto hay dos diferencias estadísticamente significativas con el AG a favor de éste y con el FSSEBNA hay 4 de las que 2 son a favor del FSSEBNA y otras 2 a favor de la BD. En lo relativo a la reducción, se encontraron 9 diferencias estadísticamente significativas con el AG y con el FSSEBNA en todos los resultados, excepto en 3, la diferencia era significativa. En todos estos casos la BD es la estrategia que reduce en mayor medida.

El test de Friedman nos indica que las diferencias en los resultados obtenidos con el IB1 se debe a diferencias en el rendimiento de cada estrategia. Analizando los resultados, se observa que las diferencias en el acierto con el AG son significativas al nivel $\alpha = 0{,}05$ y con el FSSEBNA a $\alpha = 0{,}1$. En la reducción las diferencias con el AG y FSSEBNA son significativas a $\alpha = 0{,}05$.

El análisis en función de la regla de clasificación indica que en el acierto, las diferencias son estadísticamente significativas con el AG al nivel $\alpha = 0{,}05$. En ambos casos la eficacia promedio con el AG es superior a la alcanzada con la BD. En los conjuntos lineales, el AG obtuvo un promedio de 70,3 % frente a 64,4 % de la BD y en el no lineal de 70,3 % frente a 65,0 %. Este mismo análisis en función del número de atributos relevantes nos indica que las diferencias con el FSSEBNA son significativas al nivel $\alpha = 0{,}1$ en el caso en que hay 5 atributos relevantes, alcanzado un acierto, en promedio, de 73,5 % la BD y de 67,45 % el FSSEBNA. Con 10 atributos relevantes la diferencias con el AG y FSSEBNA son significativas a $\alpha = 0{,}05$. En este caso la BD es la que obtiene peores resultados.

En todos los casos anteriormente estudiados la BD es la estrategia que reduce en mayor medida y la diferencia es estadísticamente significativa al nivel $\alpha = 0{,}05$.

La Tabla 4.6 muestra los resultados obtenidos con el clasificador J48. La eficacia alcanzada con la BD supera, en la mayoría de los casos, la alcanzada por las estrategias AG y FSSEBNA. Además, en los problemas con mayor número de atributos es la estrategia que degrada en menor medida el clasificador, seleccionando siempre el subconjunto de atributos de menor tamaño. Las diferencias, en el acierto, con el AG son estadísticamente significativas en un caso y en 3 con el FSSEBNA. La reducción, como en los casos anteriores, es mayor para la estrategia BD. En 12 problemas las diferencias con el AG son estadísticamente significativas mientras que con el FSSEBNA las diferencias son significativas en casi todos los problemas estudiados.

El test de Friedman nos indica que existen diferencias en el rendimiento de estas estrategias con el J48. Las diferencias en el acierto y la reducción obtenidas son estadísticamente significativas al nivel $\alpha = 0{,}05$ con el AG y FSSEBNA. El análisis del acierto en función de la regla de clasificación indica que tanto en los problemas lineales como en los no lineales las diferencias con el AG y FSSEBNA son significativas a $\alpha = 0{,}05$. En ambos casos la BD es la estrategia con la que el J48 obtiene mejor promedio. Análogamente, haciendo el estudio en función del número de atributos relevantes encontramos que las diferencias de la BD con el AG y el FSSEBNA son significativas al nivel $\alpha = 0{,}05$ y la BD es la estrategia que obtiene mejores promedios. En ambos análisis la reducción obtenida con la BD es mayor significativamente al nivel $\alpha = 0{,}05$.

### Problemas reales

En la Tabla 4.7 se muestran los resultados obtenidos en los conjuntos reales usando el clasificador Naïve Bayes. En la comparativa puede verse que dependiendo del problema considerado variará la estrategia que obtiene mejores resultados. El AG presenta diferencias en el acierto estadísticamente

significativas en el *ann*, a favor de éste. Las diferencias con el FSSEBNA son significativas en 3 conjuntos. Las 2 diferencias en los conjuntos de tamaño mediano (*soy* y *ann*) son favorables al FSSEBNA mientras que el resultado del conjunto de tamaño grande (*rat*) es favorable a la BD. Analizando las diferencias encontradas en los resultados en función del tamaño del subconjunto de atributos encontrado se puede observar que las diferencias estadísticamente significativas aumentan a medida que aumenta el tamaño del problema. La estrategia FSSEBNA es la que obtiene en promedio el mayor acierto aunque las diferencias con las estrategias AG y BD son pequeñas. La estrategia que más reduce es la BD, seguida del AG y FSSEBNA.

El test de Friedman aplicado a todos los resultados indica que existen diferencias en el rendimiento de las estrategias en estudio. En el acierto estas diferencias son estadísticamente significativas al nivel $\alpha = 0{,}1$ y en la reducción a $\alpha = 0{,}05$. Haciendo el estudio en función del tamaño no encontramos diferencias en el rendimiento de las estrategias presentadas. En lo que a la reducción se refiere, sí existen diferencias entre el AG y la BD al nivel $\alpha = 0{,}1$ en conjuntos grandes y con el FSSEBNA al nivel $\alpha = 0{,}05$ en todos los casos.

La Tabla 4.8 contiene los resultados obtenidos con el clasificador IB1. Como puede verse el acierto obtenido con la BD es inferior en diversos casos, aunque estas mejoras son estadísticamente significativas en el *son* con el AG y con los conjuntos *abp*, *son* y *aud* con el FSSEBNA. Como en el caso anterior, a medida que aumenta el tamaño del problema, el número de diferencias significativas en la reducción aumenta, sobre todo para el FSSEBNA que es la estrategia que selecciona subconjuntos de mayor tamaño.

El test de Friedman aplicado sobre todos los resultados obtenidos encuentra diferencias en el rendimiento entre la BD y el AG y FSSEBNA tanto en el acierto como en la reducción. Estas diferencias se dan al nivel $\alpha = 0{,}05$. Un análisis más detallado en función del tamaño del conjunto de datos nos indica que en el acierto, las diferencias con el AG son estadísticamente significativas en conjuntos medianos al nivel $\alpha = 0{,}1$. En la reducción, las diferencias con el AG se dan en problemas medianos al nivel $\alpha = 0{,}05$ y con el FSSEBNA a $\alpha = 0{,}05$ en todos los casos. Con este clasificador la BD es la que obtiene, en promedio, peor porcentaje de acierto.

En la Tabla 4.9 se muestran los resultados obtenidos con el clasificador J48. Como en los casos anteriores, no existe ninguna estrategia que mejore la eficacia en todos los problemas. En los problemas de tamaño mediano hay, en los resultados del AG, una diferencia estadísticamente significativa (*soy*) y en 2 del FSSEBNA dos (*soy* y *kvk*). En los conjuntos grandes, sólo en el problema *aud* hay diferencia estadísticamente significativa entre la BD y las estrategias AG y FSSEBNA. En todos estos conjuntos en que las diferencias son estadísticamente significativas la BD obtiene una menor precisión. Como en los casos anteriores, a medida que aumenta el número de atributos de un conjunto de datos, mayor es la diferencia en la reducción entre BD y las otras dos estrategias. Esto se refleja en el aumento de las diferencias estadísticamente significativas a medida que los conjuntos de datos aumentan el número de atributos. Como en los casos anteriores, el FSSEBNA es el que encuentra subconjuntos de mayor tamaño seguido del AG y BD.

El test de Friedman aplicado sobre todos los resultados en el acierto no encuentra que haya diferencias en el rendimiento de las distintas estrategias. Este análisis en la reducción si encuentra diferencias con el AG y FSSEBNA al nivel $\alpha = 0{,}05$. Aplicando el test de Nemenyi en función del tamaño de los datos encontramos que en el acierto sólo en los conjuntos de tamaño mediano hay diferencias estadísticamente significativas con el AG y FSSEBNA al nivel $\alpha = 0{,}1$. La BD es la que obtiene en este caso el peor promedio obteniendo 84,4 % frente a 87,4 % del AG y 87,2 % del FSSEBNA. Comparando el tamaño de los conjuntos de atributos encontrados que en los conjuntos pequeños sólo hay diferencias significativas con el FSSEBNA a $\alpha = 0{,}1$ y en los problemas medianos y grandes, las diferencias son con el FSSEBNA al nivel $\alpha = 0{,}1$ y con el AG a $\alpha = 0{,}1$. En todos estos casos la BD es la que reduce en mayor medida.

### 4.4.7 Tiempos de ejecución

A continuación pasamos a analizar los tiempos de ejecución empleados por cada una de las estrategias presentadas. Este análisis es llevado a cabo en problemas artificiales y reales. Los resultados correspondientes a los conjuntos

artificiales están en las Tablas 4.10, 4.11 y 4.12 y los correspondientes a los reales en las Tablas 4.13, 4.14 y 4.15.

### Problemas artificiales

Los tiempos de ejecución empleados por cada estrategia en los conjuntos artificiales son mostrados en las Tablas 4.10, 4.11 y 4.12 empleando los clasificadores Naïve Bayes, IB1 y J48 respectivamente. Como puede verse, la BD es la estrategia que emplea, en casi todos los problemas, el menor tiempo en realizar la búsqueda. Las diferencias de la BD con el AG y FSSEBNA son estadísticamente significativa en la mayoría de los resultados, y en promedio aumenta a medida que incrementamos el número de atributos presente en los conjuntos de datos.

El test de Friedman indica que las diferencias presentes en estos resultados se deben al diferente rendimiento de las estrategias. Las diferencias entre la BD y el AG y el FSSEBNA son estadísticamente significativas al nivel $\alpha = 0{,}05$. En todos los estudios realizados (la regla de clasificación y el número de atributos relevantes), se obtuvo que las diferencias entre la BD y las estrategias AG y FSSEBNA eran estadísticamente significativas al nivel $\alpha = 0{,}05$. En promedio, la estrategia FSSEBNA es la que tarda más tiempo seguida del AG y la BD.

Tabla 4.10: Comparativa entre la metaheurística BD y las estrategias AG y FSSEBNA con el clasificador Naïve Bayes. Tiempo de ejecución en conjuntos artificiales.

| conjuntos de datos | | | | tiempos de ejecución | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Naïve Bayes | | | | | |
| l | r | $\sigma$ | d | BD | AG | | FSSEBNA | | |
| 1 | 5 | 0 | 25 | 178,7 ± 52,5 | 636,2 ± 15,4 | $\ll$ | 1281,9 ± 20,5 | $\ll$ |
| | | | 50 | 175,1 ± 41,9 | 1309,4 ± 24,8 | $\ll$ | 2435,9 ± 26,8 | $\ll$ |
| | | | 100 | 200,6 ± 41,7 | 2703,0 ± 71,3 | $\ll$ | 4778,6 ± 31,4 | $\ll$ |
| | | | 200 | 244,0 ± 68,9 | 5508,4 ± 174,6 | $\ll$ | 9583,4 ± 75,2 | $\ll$ |
| | | 0,1 | 25 | 155,7 ± 50,8 | 623,2 ± 8,8 | $\ll$ | 1268,7 ± 28,8 | $\ll$ |
| | | | 50 | 188,4 ± 50,0 | 1306,8 ± 26,5 | $\ll$ | 2465,9 ± 33,3 | $\ll$ |
| | | | 100 | 215,0 ± 42,5 | 2824,4 ± 190,8 | $\ll$ | 4839,7 ± 40,3 | $\ll$ |
| | | | 200 | 227,8 ± 57,7 | 5567,4 ± 204,7 | $\ll$ | 9634,0 ± 158,7 | $\ll$ |
| | 10 | 0 | 25 | 184,6 ± 66,6 | 633,4 ± 17,9 | $\ll$ | 1303,7 ± 23,7 | $\ll$ |
| | | | 50 | 197,9 ± 46,8 | 1311,3 ± 33,1 | $\ll$ | 2481,9 ± 19,0 | $\ll$ |
| | | | 100 | 272,5 ± 59,4 | 2683,8 ± 77,9 | $\ll$ | 4822,4 ± 46,5 | $\ll$ |
| | | | 200 | 297,1 ± 75,0 | 5488,9 ± 120,9 | $\ll$ | 9676,5 ± 91,1 | $\ll$ |
| | | 0,1 | 25 | 170,1 ± 45,3 | 638,8 ± 18,2 | $\ll$ | 1327,2 ± 41,5 | $\ll$ |
| | | | 50 | 218,6 ± 22,2 | 1310,0 ± 31,3 | $\ll$ | 2495,1 ± 23,8 | $\ll$ |
| | | | 100 | 252,9 ± 39,0 | 3136,5 ± 69,4 | $\ll$ | 4865,3 ± 40,5 | $\ll$ |
| | | | 200 | 299,9 ± 84,8 | 5507,0 ± 100,0 | $\ll$ | 9636,8 ± 56,9 | $\ll$ |
| 2 | 5 | 0 | 25 | 156,1 ± 24,5 | 630,6 ± 14,7 | $\ll$ | 1268,1 ± 16,5 | $\ll$ |
| | | | 50 | 161,7 ± 34,5 | 1286,9 ± 16,2 | $\ll$ | 2439,2 ± 30,6 | $\ll$ |
| | | | 100 | 186,7 ± 35,5 | 2671,6 ± 40,6 | $\ll$ | 4775,2 ± 57,2 | $\ll$ |
| | | | 200 | 227,1 ± 84,1 | 5510,4 ± 130,1 | $\ll$ | 9593,2 ± 60,9 | $\ll$ |
| | | 0,1 | 25 | 140,6 ± 44,0 | 629,7 ± 14,2 | $\ll$ | 1262,9 ± 15,7 | $\ll$ |
| | | | 50 | 151,0 ± 38,2 | 1324,0 ± 37,5 | $\ll$ | 2439,1 ± 21,1 | $\ll$ |
| | | | 100 | 195,7 ± 54,2 | 2709,7 ± 70,7 | $\ll$ | 7161,8 ± 2552,5 | $\ll$ |
| | | | 200 | 241,8 ± 79,0 | 5505,8 ± 97,7 | $\ll$ | 9577,0 ± 62,3 | $\ll$ |
| | 10 | 0 | 25 | 167,1 ± 60,9 | 630,9 ± 17,0 | $\ll$ | 1311,0 ± 20,1 | $\ll$ |
| | | | 50 | 201,9 ± 52,7 | 1282,4 ± 21,9 | $\ll$ | 2478,7 ± 34,9 | $\ll$ |
| | | | 100 | 232,5 ± 42,2 | 2657,1 ± 39,1 | $\ll$ | 4819,9 ± 46,4 | $\ll$ |
| | | | 200 | 330,8 ± 116,7 | 5448,9 ± 117,9 | $\ll$ | 9621,1 ± 62,2 | $\ll$ |
| | | 0,1 | 25 | 176,7 ± 52,8 | 634,1 ± 14,2 | $\ll$ | 1308,7 ± 26,3 | $\ll$ |
| | | | 50 | 188,6 ± 43,1 | 1290,3 ± 33,8 | $\ll$ | 2469,6 ± 19,6 | $\ll$ |
| | | | 100 | 252,5 ± 33,1 | 2694,0 ± 96,0 | $\ll$ | 4829,9 ± 49,8 | $\ll$ |
| | | | 200 | 361,8 ± 84,5 | 5456,3 ± 140,0 | $\ll$ | 9588,7 ± 92,3 | $\ll$ |

*l*-regla de clasificación; *r*-número de atributos relevantes; $\sigma$-ruido Gaussiano;BD-Búsqueda Dispersa; AG-Algoritmo Genético; FSSEBNA-Feature Subset Selection by Estimation of Bayesian Network Algorithm; $>$ y $\gg$ mejora significativa en un 95 y 99 %; $<$ y $\ll$ empeora significativamente.

Tabla 4.11: Comparativa entre la metaheurística BD y las estrategias AG y FSSEBNA con el clasificador IB1. Tiempo de ejecución en conjuntos artificiales.

| conjuntos de datos | | | | tiempos de ejecución IB1 | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| l | r | $\sigma$ | d | BD | AG | | FSSEBNA | | |
| 1 | 5 | 0 | 25 | 62,6 ± 37,2 | 139,5 ± 17,1 | < | 231,8 ± 12,8 | ≪ | |
| | | | 50 | 51,0 ± 49,4 | 275,2 ± 34,6 | ≪ | 465,8 ± 58,4 | ≪ | |
| | | | 100 | 43,7 ± 37,0 | 553,5 ± 79,0 | ≪ | 866,9 ± 52,8 | ≪ | |
| | | | 200 | 43,9 ± 20,9 | 1060,0 ± 117,1 | ≪ | 1723,2 ± 18,6 | ≪ | |
| | | 0,1 | 25 | 69,7 ± 41,2 | 140,3 ± 16,2 | | 240,1 ± 21,4 | ≪ | |
| | | | 50 | 47,2 ± 20,7 | 276,2 ± 34,2 | ≪ | 449,0 ± 31,7 | ≪ | |
| | | | 100 | 42,9 ± 30,4 | 498,6 ± 18,5 | ≪ | 910,6 ± 83,2 | ≪ | |
| | | | 200 | 50,0 ± 13,5 | 1009,0 ± 36,4 | ≪ | 1835,0 ± 151,4 | ≪ | |
| | 10 | 0 | 25 | 38,1 ± 41,2 | 141,7 ± 17,8 | | 253,2 ± 24,7 | ≪ | |
| | | | 50 | 29,6 ± 35,0 | 263,7 ± 30,5 | ≪ | 469,9 ± 43,4 | ≪ | |
| | | | 100 | 15,5 ± 6,5 | 587,4 ± 66,4 | ≪ | 905,3 ± 77,4 | ≪ | |
| | | | 200 | 24,6 ± 11,4 | 1107,1 ± 136,2 | ≪ | 1844,0 ± 153,6 | ≪ | |
| | | 0,1 | 25 | 35,6 ± 23,5 | 136,7 ± 12,7 | ≪ | 252,9 ± 22,7 | ≪ | |
| | | | 50 | 15,4 ± 12,3 | 273,3 ± 38,0 | ≪ | 462,1 ± 37,2 | ≪ | |
| | | | 100 | 23,1 ± 22,4 | 551,6 ± 75,8 | ≪ | 887,1 ± 65,8 | ≪ | |
| | | | 200 | 19,2 ± 7,1 | 1108,4 ± 143,3 | ≪ | 1838,5 ± 157,2 | ≪ | |
| 2 | 5 | 0 | 25 | 69,2 ± 33,6 | 141,5 ± 17,3 | | 227,6 ± 3,4 | ≪ | |
| | | | 50 | 46,5 ± 20,1 | 250,0 ± 13,6 | ≪ | 462,8 ± 37,6 | ≪ | |
| | | | 100 | 61,9 ± 26,8 | 521,4 ± 55,4 | ≪ | 904,5 ± 73,6 | ≪ | |
| | | | 200 | 69,5 ± 27,1 | 1069,4 ± 106,0 | ≪ | 1866,6 ± 151,4 | ≪ | |
| | | 0,1 | 25 | 47,1 ± 27,8 | 145,4 ± 21,3 | ≪ | 234,6 ± 20,1 | ≪ | |
| | | | 50 | 43,8 ± 34,0 | 277,3 ± 38,6 | ≪ | 456,0 ± 38,8 | ≪ | |
| | | | 100 | 47,0 ± 27,8 | 494,0 ± 21,6 | ≪ | 874,2 ± 48,0 | ≪ | |
| | | | 200 | 55,5 ± 24,5 | 1056,7 ± 124,0 | ≪ | 1743,5 ± 97,2 | ≪ | |
| | 10 | 0 | 25 | 26,6 ± 26,6 | 144,0 ± 18,7 | ≪ | 236,5 ± 14,9 | ≪ | |
| | | | 50 | 27,2 ± 42,1 | 265,0 ± 30,2 | ≪ | 442,6 ± 27,3 | ≪ | |
| | | | 100 | 16,6 ± 14,4 | 519,0 ± 55,6 | ≪ | 897,5 ± 76,3 | ≪ | |
| | | | 200 | 18,6 ± 7,7 | 1020,7 ± 40,2 | ≪ | 1808,8 ± 140,3 | ≪ | |
| | | 0,1 | 25 | 9,9 ± 5,1 | 135,0 ± 12,5 | ≪ | 236,2 ± 14,9 | ≪ | |
| | | | 50 | 16,8 ± 15,8 | 252,4 ± 21,6 | ≪ | 459,0 ± 40,5 | ≪ | |
| | | | 100 | 17,1 ± 10,2 | 555,6 ± 72,4 | ≪ | 917,6 ± 117,4 | ≪ | |
| | | | 200 | 22,3 ± 7,5 | 1137,2 ± 139,2 | ≪ | 1744,7 ± 96,1 | ≪ | |

$l$-regla de clasificación; $r$-número de atributos relevantes; $\sigma$-ruido Gaussiano;BD-Búsqueda Dispersa; AG-Algoritmo Genético; FSSEBNA-Feature Subset Selection by Estimation of Bayesian Network Algorithm; $>$ y $\gg$ mejora significativa en un 95 y 99 %; $<$ y $\ll$ empeora significativamente.

Tabla 4.12: Comparativa entre la metaheurística BD y las estrategias AG y FSSEBNA con el clasificador J48. Tiempo de ejecución en conjuntos artificiales.

| conjuntos de datos | | | | tiempos de ejecución J48 | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| l | r | $\sigma$ | d | BD | AG | | FSSEBNA | | |
| 1 | 5 | 0 | 25 | 54,2 ± 17,6 | 67,8 ± 7,9 | | 104,5 ± 4,1 | $\ll$ | |
| | | | 50 | 53,5 ± 13,4 | 131,7 ± 10,7 | $\ll$ | 203,7 ± 6,8 | $\ll$ | |
| | | | 100 | 50,0 ± 7,9 | 272,3 ± 11,6 | $\ll$ | 397,9 ± 15,3 | $\ll$ | |
| | | | 200 | 66,9 ± 26,0 | 560,6 ± 23,8 | $\ll$ | 831,7 ± 32,7 | $\ll$ | |
| | | 0,1 | 25 | 43,5 ± 18,2 | 67,3 ± 7,2 | | 105,8 ± 3,7 | $<$ | |
| | | | 50 | 55,3 ± 25,2 | 132,5 ± 8,7 | $<$ | 204,2 ± 6,5 | $\ll$ | |
| | | | 100 | 52,8 ± 17,9 | 273,9 ± 13,4 | $\ll$ | 398,8 ± 16,3 | $\ll$ | |
| | | | 200 | 69,7 ± 23,4 | 566,0 ± 20,9 | $\ll$ | 832,5 ± 23,5 | $\ll$ | |
| | 10 | 0 | 25 | 89,3 ± 32,7 | 74,6 ± 7,6 | | 116,7 ± 3,5 | | |
| | | | 50 | 94,9 ± 29,4 | 143,9 ± 11,5 | | 225,4 ± 9,9 | $\ll$ | |
| | | | 100 | 92,3 ± 34,4 | 297,4 ± 7,4 | $\ll$ | 441,2 ± 14,3 | $\ll$ | |
| | | | 200 | 90,8 ± 38,2 | 599,1 ± 15,4 | $\ll$ | 908,0 ± 37,0 | $\ll$ | |
| | | 0,1 | 25 | 95,9 ± 37,1 | 74,1 ± 10,2 | | 114,4 ± 5,6 | | |
| | | | 50 | 77,0 ± 41,2 | 144,8 ± 11,3 | | 224,4 ± 9,2 | $\ll$ | |
| | | | 100 | 68,5 ± 31,7 | 293,8 ± 13,9 | $\ll$ | 436,0 ± 16,5 | $\ll$ | |
| | | | 200 | 67,7 ± 42,5 | 602,3 ± 21,3 | $\ll$ | 905,7 ± 44,4 | $\ll$ | |
| 2 | 5 | 0 | 25 | 53,3 ± 26,8 | 65,8 ± 8,9 | | 99,1 ± 4,4 | | |
| | | | 50 | 57,9 ± 29,3 | 128,6 ± 11,2 | $<$ | 193,5 ± 7,4 | $\ll$ | |
| | | | 100 | 60,2 ± 33,1 | 268,2 ± 12,3 | $\ll$ | 379,5 ± 8,2 | $\ll$ | |
| | | | 200 | 68,2 ± 56,2 | 542,7 ± 20,7 | $\ll$ | 777,2 ± 28,9 | $\ll$ | |
| | | 0,1 | 25 | 50,6 ± 28,1 | 66,2 ± 8,0 | | 98,8 ± 3,8 | $<$ | |
| | | | 50 | 35,7 ± 21,1 | 127,1 ± 10,0 | $\ll$ | 190,6 ± 7,1 | $\ll$ | |
| | | | 100 | 67,7 ± 48,4 | 267,0 ± 13,7 | $\ll$ | 375,2 ± 13,4 | $\ll$ | |
| | | | 200 | 62,9 ± 20,4 | 547,2 ± 16,8 | $\ll$ | 785,0 ± 22,7 | $\ll$ | |
| | 10 | 0 | 25 | 70,3 ± 48,4 | 74,0 ± 9,5 | | 116,7 ± 5,5 | | |
| | | | 50 | 68,0 ± 41,0 | 139,3 ± 10,1 | | 228,5 ± 11,5 | $\ll$ | |
| | | | 100 | 73,3 ± 42,8 | 290,6 ± 15,4 | $\ll$ | 448,6 ± 22,6 | $\ll$ | |
| | | | 200 | 89,8 ± 52,4 | 599,2 ± 17,0 | $\ll$ | 936,5 ± 45,9 | $\ll$ | |
| | | 0,1 | 25 | 87,0 ± 49,4 | 73,3 ± 9,8 | | 115,7 ± 3,7 | | |
| | | | 50 | 67,1 ± 50,9 | 141,6 ± 10,6 | $<$ | 224,2 ± 8,3 | $\ll$ | |
| | | | 100 | 70,9 ± 32,1 | 290,0 ± 10,2 | $\ll$ | 444,2 ± 25,9 | $\ll$ | |
| | | | 200 | 85,1 ± 47,8 | 593,4 ± 24,8 | $\ll$ | 930,5 ± 54,5 | $\ll$ | |

$l$-regla de clasificación; $r$-número de atributos relevantes; $\sigma$-ruido Gaussiano;BD-Búsqueda Dispersa; AG-Algoritmo Genético; FSSEBNA-Feature Subset Selection by Estimation of Bayesian Network Algorithm; $>$ y $\gg$ mejora significativa en un 95 y 99 %; $<$ y $\ll$ empeora significativamente.

### Problemas reales

Los resultados en los conjuntos de datos reales están en las Tablas 4.13, 4.14 y 4.15. Como en el caso anterior, los resultados mostrados fueron obtenidos con los clasificadores Naïve Bayes, IB1 y J48 respectivamente. Los resultados en los problemas que tardan poco tiempo no está claro cuál es la estrategia que tarda menos tiempo pues la desviación típica, en estos casos es elevada y por tanto las diferencias no son estadísticamente significativas en la mayoría de los casos. Sin embargo, en los problemas que requieren un mayor tiempo de ejecución, la BD es la estrategia que emplea menos tiempo. Esto se refleja en los conjuntos de mayor tamaño, tanto en número de atributos como en número de ejemplos. Casi todas las diferencias que son estadísticamente significativas son favorables a la BD y en algunos casos el tiempo empleado llega a ser un orden de magnitud inferior.

El test de Friedman aplicado sobre todos los resultados presenta diferencias en el rendimiento de las estrategias presentadas con el FSSEBNA al nivel $\alpha = 0{,}05$ en los clasificadores IB1 y J48. En promedio la BD es la que emplea menos tiempo en realizar la búsqueda.

Analizando los resultados en función del tamaño de los conjuntos de datos, encontramos que con el Naïve Bayes en los conjuntos de datos pequeños y medianos, el test de Friedman indica que hay diferencias en el rendimiento entre las estrategias. Las diferencias entre la BD y AG son estadísticamente significativas al nivel $\alpha = 0{,}05$.

Con el clasificador IB1 la proporción de diferencias estadísticamente significativas aumenta a medida que aumenta el tamaño del problema. Para los conjuntos de datos grandes y pequeños el test de Friedman no desvela que el rendimiento sea diferente, mientras que para el caso de los conjuntos de datos mediano sí. Las diferencias con el AG son significativas con $\alpha = 0{,}1$ y con el FSSEBNA con $\alpha = 0{,}05$.

Las conclusiones para el clasificador J48 son similares, teniendo una interpretación análoga. Por último destacar que el test de Friedman indica diferencias significativas con el FSSEBNA al nivel $\alpha = 0{,}05$ en los conjuntos medianos y grandes.

Tabla 4.13: Comparativa entre la metaheurística $VNS_m$, SFFS, AG y FSSEBNA con el clasificador Naïve Bayes. Tiempo de ejecución en conjuntos reales.

| cdd | tiempos de ejecución | | |
|---|---|---|---|
| | Naïve Bayes | | |
| | BD | AG | FSSEBNA |
| pdi | 68,8 ± 44,1 | 26,8 ± 0,5 | 26,8 ± 0,5 |
| gla | 39,6 ± 23,8 | 12,6 ± 1,0 | 11,9 ± 0,4 |
| bca | 14,6 ± 15,0 | 1,9 ± 0,4 | 1,6 ± 0,0 |
| bwi | 111,7 ± 36,2 | 16,4 ± 0,4 > | 15,7 ± 0,3 > |
| win | 110,8 ± 36,7 | 46,6 ± 3,2 | 68,1 ± 2,0 |
| hst | 70,8 ± 44,1 | 42,0 ± 2,8 | 59,6 ± 2,1 |
| hhu | 54,0 ± 38,7 | 35,0 ± 2,7 | 46,4 ± 3,2 |
| hcl | 69,6 ± 47,3 | 45,3 ± 3,7 | 61,1 ± 3,0 |
| vow | 270,7 ± 48,9 | 1179,3 ± 25,4 ≪ | 1670,4 ± 59,8 ≪ |
| cra | 43,3 ± 15,8 | 125,9 ± 5,3 ≪ | 202,2 ± 8,2 ≪ |
| ptu | 92,1 ± 68,3 | 21,1 ± 8,8 | 23,3 ± 0,8 |
| vot | 23,1 ± 8,0 | 9,0 ± 2,0 | 12,1 ± 0,3 |
| lym | 62,0 ± 49,7 | 12,2 ± 3,8 | 16,1 ± 1,5 |
| veh | 57,8 ± 21,9 | 730,1 ± 26,1 ≪ | 1484,5 ± 49,9 ≪ |
| hep | 10,5 ± 7,7 | 17,8 ± 3,7 | 30,3 ± 2,1 < |
| hco | 99,3 ± 49,5 | 45,6 ± 3,2 | 78,1 ± 3,2 |
| aut | 82,4 ± 34,3 | 84,0 ± 8,4 | 155,2 ± 8,5 < |
| abp | 30,5 ± 7,5 | 762,1 ± 23,6 ≪ | 1392,8 ± 64,1 ≪ |
| ion | 122,4 ± 55,3 | 296,6 ± 9,1 ≪ | 551,5 ± 18,0 ≪ |
| soy | 33,8 ± 14,9 | 58,8 ± 18,3 | 75,0 ± 1,8 < |
| kvk | 36,0 ± 15,4 | 104,0 ± 5,8 < | 160,6 ± 2,1 ≪ |
| ann | 17,7 ± 8,5 | 125,7 ± 8,4 ≪ | 224,1 ± 6,5 ≪ |
| lca | 11,5 ± 9,3 | 6,8 ± 3,7 | 6,7 ± 0,1 |
| dnp | 63,0 ± 29,3 | 12,8 ± 8,6 | 9,7 ± 0,2 > |
| son | 26,7 ± 16,4 | 303,9 ± 8,3 ≪ | 577,2 ± 6,2 ≪ |
| dna | 78,2 ± 21,4 | 143,0 ± 12,7 | 221,3 ± 4,2 ≪ |
| spl | 82,0 ± 25,0 | 210,1 ± 128,9 | 226,6 ± 3,4 ≪ |
| aud | 157,1 ± 53,1 | 46,9 ± 22,9 | 49,7 ± 1,9 |
| rat | 28,3 ± 22,7 | 334,3 ± 17,6 ≪ | 634,5 ± 25,5 ≪ |
| arr | 159,4 ± 77,8 | 3848,9 ± 230,8 ≪ | 6540,6 ± 54,4 ≪ |

$l$-regla de clasificación; $r$-número de atributos relevantes; $\sigma$-ruido Gaussiano;BD-Búsqueda Dispersa; AG-Algoritmo Genético; FSSEBNA-Feature Subset Selection by Estimation of Bayesian Network Algorithm; > y ≫ mejora significativa en un 95 y 99 %; < y ≪ empeora significativamente.

Tabla 4.14: Comparativa entre la metaheurística $VNS_m$, SFFS, AG y FSSEBNA con el clasificador IB1. Tiempo de ejecución en conjuntos reales.

| cdd | tiempos de ejecución | | |
|---|---|---|---|
| | IB1 | | |
| | BD | AG | FSSEBNA |
| pdi | 31,9 ± 21,2 | 15,3 ± 2,5 | 14,5 ± 1,2 |
| gla | 142,0 ± 33,8 | 4,4 ± 1,0 ≫ | 3,5 ± 0,3 ≫ |
| bca | 7,8 ± 4,6 | 5,3 ± 1,2 | 4,2 ± 0,1 |
| bwi | 123,2 ± 54,3 | 26,3 ± 4,0 | 23,4 ± 1,9 |
| win | 116,9 ± 46,8 | 13,7 ± 3,7 > | 14,2 ± 0,6 > |
| hst | 50,4 ± 26,1 | 22,7 ± 4,7 | 28,6 ± 1,2 |
| hhu | 56,3 ± 41,5 | 24,5 ± 5,6 | 27,3 ± 0,9 |
| hcl | 36,7 ± 24,7 | 26,4 ± 5,6 | 31,7 ± 1,7 |
| vow | 239,4 ± 49,5 | 245,1 ± 24,8 | 297,6 ± 14,7 |
| cra | 67,3 ± 39,7 | 145,8 ± 17,1 < | 218,1 ± 11,0 ≪ |
| ptu | 57,1 ± 56,2 | 42,7 ± 10,7 | 63,7 ± 1,1 |
| vot | 38,0 ± 35,9 | 53,1 ± 8,8 | 78,9 ± 1,8 |
| lym | 50,6 ± 56,0 | 14,1 ± 4,9 | 17,5 ± 0,4 |
| veh | 128,9 ± 62,4 | 291,4 ± 30,4 < | 499,9 ± 32,8 ≪ |
| hep | 7,4 ± 5,0 | 14,6 ± 4,6 | 19,7 ± 0,4 < |
| hco | 31,7 ± 32,6 | 61,0 ± 9,2 | 97,8 ± 1,3 < |
| aut | 27,8 ± 14,0 | 28,9 ± 6,1 | 43,6 ± 1,9 |
| abp | 109,9 ± 39,3 | 6267,3 ± 179,3 ≪ | 12074,1 ± 122,9 ≪ |
| ion | 61,1 ± 35,9 | 91,6 ± 9,9 | 154,5 ± 12,2 ≪ |
| soy | 106,3 ± 78,0 | 267,5 ± 33,1 ≪ | 442,2 ± 4,5 ≪ |
| kvk | 286,7 ± 104,1 | 5679,8 ± 161,8 ≪ | 9644,6 ± 218,5 ≪ |
| ann | 152,6 ± 52,4 | 341,6 ± 33,6 ≪ | 573,1 ± 11,3 ≪ |
| lca | 18,0 ± 14,4 | 9,0 ± 6,1 | 7,0 ± 0,1 |
| dnp | 30,1 ± 27,4 | 18,9 ± 6,7 | 23,0 ± 0,2 |
| son | 27,1 ± 34,6 | 63,0 ± 9,8 | 104,7 ± 8,8 |
| dna | 305,1 ± 119,4 | 8504,0 ± 114,4 ≪ | 15425,3 ± 127,6 ≪ |
| spl | 286,1 ± 60,7 | 9083,4 ± 217,2 ≪ | 15667,5 ± 97,7 ≪ |
| aud | 115,0 ± 73,9 | 71,7 ± 17,6 | 95,0 ± 0,7 |
| rat | 13,7 ± 10,6 | 122,8 ± 20,1 ≪ | 214,8 ± 11,0 ≪ |
| arr | 43,4 ± 25,7 | 1167,0 ± 157,6 ≪ | 1999,2 ± 99,2 ≪ |

$l$-regla de clasificación; $r$-número de atributos relevantes; $\sigma$-ruido Gaussiano;BD-Búsqueda Dispersa; AG-Algoritmo Genético; FSSEBNA-Feature Subset Selection by Estimation of Bayesian Network Algorithm; > y ≫ mejora significativa en un 95 y 99 %; < y ≪ empeora significativamente.

Tabla 4.15: Comparativa entre la metaheurística $VNS_m$, SFFS, AG y FSSEBNA con el clasificador J48. Tiempo de ejecución en conjuntos reales.

| cdd | tiempos de ejecución | | |
|---|---|---|---|
| | J48 | | |
| | BD | AG | FSSEBNA |
| pdi | 15,6 ± 11,7 | 4,2 ± 0,3 | 3,9 ± 0,2 |
| gla | 45,8 ± 13,2 | 5,7 ± 1,2 > | 4,6 ± 0,2 > |
| bca | 3,1 ± 1,4 | 2,8 ± 0,6 | 2,3 ± 0,1 |
| bwi | 26,9 ± 17,3 | 5,8 ± 0,6 | 5,0 ± 0,2 |
| win | 29,9 ± 13,2 | 11,0 ± 2,7 | 11,9 ± 0,7 |
| hst | 12,6 ± 10,3 | 13,9 ± 2,9 | 17,1 ± 1,0 |
| hhu | 6,8 ± 3,8 | 11,6 ± 2,5 | 12,7 ± 1,1 |
| hcl | 7,2 ± 2,6 | 14,5 ± 3,6 | 16,2 ± 1,0 ≪ |
| vow | 96,3 ± 35,0 | 189,9 ± 15,8 < | 230,8 ± 16,6 < |
| cra | 15,5 ± 12,4 | 36,3 ± 6,4 < | 50,2 ± 3,0 < |
| ptu | 30,1 ± 23,3 | 37,2 ± 7,7 | 57,9 ± 1,5 |
| vot | 19,5 ± 8,6 | 13,4 ± 3,0 | 17,9 ± 1,0 |
| lym | 9,4 ± 4,6 | 10,3 ± 3,8 | 12,2 ± 0,8 |
| veh | 104,5 ± 54,2 | 120,0 ± 11,3 | 199,1 ± 5,7 |
| hep | 5,4 ± 1,9 | 10,4 ± 3,6 | 13,6 ± 0,7 < |
| hco | 10,8 ± 4,4 | 22,8 ± 3,9 < | 32,1 ± 1,7 ≪ |
| aut | 29,9 ± 11,0 | 29,5 ± 8,6 | 40,5 ± 2,1 |
| abp | 9,4 ± 3,0 | 280,4 ± 12,1 ≪ | 462,3 ± 22,1 ≪ |
| ion | 27,5 ± 18,7 | 53,4 ± 5,4 < | 88,5 ± 5,1 ≪ |
| soy | 41,5 ± 15,0 | 120,6 ± 16,3 < | 177,6 ± 4,1 ≪ |
| kvk | 60,2 ± 32,7 | 296,2 ± 12,0 ≪ | 413,4 ± 8,7 ≪ |
| ann | 12,7 ± 4,5 | 88,8 ± 9,6 ≪ | 152,9 ± 5,2 ≪ |
| lca | 5,9 ± 3,2 | 9,6 ± 5,3 | 8,4 ± 0,1 |
| dnp | 5,2 ± 1,8 | 12,0 ± 4,8 | 13,5 ± 0,5 ≪ |
| son | 5,0 ± 2,7 | 50,6 ± 6,4 ≪ | 80,8 ± 2,8 ≪ |
| dna | 207,2 ± 34,8 | 292,8 ± 25,3 < | 420,7 ± 3,2 ≪ |
| spl | 134,5 ± 42,0 | 515,9 ± 108,2 < | 693,5 ± 43,0 ≪ |
| aud | 72,5 ± 21,8 | 52,1 ± 12,8 > | 68,0 ± 1,1 |
| rat | 10,7 ± 4,6 | 91,0 ± 9,3 ≪ | 167,7 ± 6,2 ≪ |
| arr | 105,7 ± 63,7 | 1063,2 ± 40,1 ≪ | 1842,6 ± 82,1 ≪ |

$l$-regla de clasificación; $r$-número de atributos relevantes; $\sigma$-ruido Gaussiano;BD-Búsqueda Dispersa; AG-Algoritmo Genético; FSSEBNA-Feature Subset Selection by Estimation of Bayesian Network Algorithm; > y ≫ mejora significativa en un 95 y 99 %; < y ≪ empeora significativamente.

## 4.5 Conclusiones

En este trabajo se propone una implementación de la metaheurística Búsqueda Dispersa para resolver el problema de selección de atributos. Diseñamos dos métodos de combinación de soluciones: la Combinación Voraz y la Combinación Voraz Reducida. Con el propósito de hacer uso de ambas combinaciones de forma simultánea e incrementar así la capacidad de exploración de la estrategia, desarrollamos una paralelización de la Búsqueda Dispersa. Dicha paralelización consiste en ejecutar cada método de combinación de soluciones en un procesador distinto. De los resultados computacionales podemos extraer las siguientes conclusiones:

- Del estudio de los valores de los parámetros de la Búsqueda Dispersa concluimos que si bien se llevó a cabo sobre un reducido número de problemas, no se encontraron diferencias estadísticamente significativas en el comportamiento. Sólo en promedio, la BD que consideraba un mayor número de soluciones en el conjunto de referencia fue la que reducía en mayor medida. Por lo tanto habría que ampliar estas pruebas para ajustar mejor los valores de la Búsqueda Dispersa. Si bien sería conveniente extender estas pruebas a otros clasificadores, en los conjuntos de datos probados la BD mostró que es una estrategia robusta frente a distintos valores en los parámetros de entrada.

- Comparando los métodos de combinación Voraz y Voraz Reducida, concluimos que ambos muestran un comportamiento similar en los problemas estudiados, tanto en el acierto como en el tamaño de los subconjuntos de atributos encontrados.

- De la comparativa entre las versiones secuenciales y paralela de la BD concluimos que la versión paralela obtiene un acierto similar a las versiones secuenciales pues las diferencias no son estadísticamente significativas. Sin embargo, en promedio, es la estrategia que reduce en mayor medida. Esto se refleja en los conjuntos de datos de mayor tamaño donde las diferencias son estadísticamente significativas. Sería

conveniente aumentar la experiencia computacional considerando conjuntos de datos de mayor tamaño así como problemas artificiales.

- De los resultados obtenidos con los clasificadores Naïve Bayes, IB1 y J48 con y sin selección de atributos considerando la estrategia BD concluimos que en los problemas artificiales considerados, la selección de atributos mejora los resultados obtenidos pues consigue eliminar atributos que dificultan la tarea de clasificar. En los problemas reales es difícil saber cuando la estrategia BD mejora o empeora los resultados del clasificador original. El IB1 es el clasificador con el que la BD rinde peor en promedio, si bien hay pocas diferencias estadísticamente significativas.

- De la comparativa entre las estrategias BD, AG y FSSEBNA concluimos que la BD es una estrategia competitiva, tal y como puede observarse en los resultados presentados. Es una estrategia que tiene gran capacidad de reducción, sobre todo en problemas en los que hay un número pequeño de atributos relevantes. Tiene un peor comportamiento con el clasificador IB1 de modo que sólo sería conveniente aplicarlo con este clasificador si el número de atributos relevantes es muy pequeño. En los problemas artificiales estudiados la BD es superior a las otras estrategias presentadas en los clasificadores Naïve Bayes y J48. Con el clasificador IB1 los resultados de la BD empeoran para el caso en que hay más atributos relevantes, siendo competitivo en el caso en que hay un número pequeño de atributos relevantes.

  En los problemas reales demuestra que es competitivo aunque ninguna de las estrategia presentadas mostraba ser superior a ninguna otra sino que hay que estudiar, para cada problema real, el rendimiento de cada estrategia. El desconocimiento de las características de los problemas reales impide poder establecer una regla que nos indique la estrategia a utilizar en función de las propiedades del problema.

- Del análisis de los tiempo empleados por las estrategias presentadas concluimos que la BD es una estrategia rápida pues es la que emplea,

de las estrategias estudiadas, menos tiempo en problemas en los que hay un número elevado de atributos y de ejemplos. En los problemas de menor tamaño (en número de atributos, de ejemplos, o ambos), las diferencias encontradas no son significativas, con lo que ninguna estrategia es superior a las otras en este aspecto.

# Capítulo 5

# Búsqueda por Entorno Variable

## 5.1 Introducción

La Búsqueda por Entorno Variable o VNS (*Variable Neighborhood Search*) [67, 68, 69] es una metaheurística reciente que resuelve problemas de optimización combinatoria y global basándose en un principio simple: cambiar de forma sistemática de estructura de entorno durante la búsqueda.

Esta metaheurística se basa en tres hechos:

- *Un mínimo local con una estructura de entorno no lo es necesariamente con otra.* Por ello, cambiar de estructura de entorno aumenta la posibilidad de encontrar óptimos locales de alta calidad.

- *Un mínimo global es mínimo local con todas las posibles estructuras de entorno.* Esto asegura que el cambio de estructura de entorno no impedirá al método encontrar, en su caso, el óptimo global.

- *Para muchos problemas, los mínimos locales con la misma o distinta estructura de entorno están cerca.* Este último hecho, comprobado empíricamente, indica que los óptimos locales proporcionan información acerca del óptimo global, y que es una buena estrategia buscar en los alrededores de los óptimos locales previamente encontrados.

Dado un espacio de soluciones $D$, una estructura de entornos es un aplicación $\mathcal{N} : D \rightarrow 2^D$ que asocia a cada solución $S \in D$ un conjunto de soluciones $\mathcal{N}(S) \subset D$, llamadas vecinas de $S$.

Sea $\mathcal{N}_k,\ k = 1, \ldots, k_{max}$ un conjunto finito de estructuras de entorno en el espacio de soluciones $D$ y $\mathcal{N}_k(S)$ el conjunto de soluciones pertenecientes a la $k$-ésima vecindad de la solución $S$. La Figura 5.1 muestra el pseudocódigo de una Búsqueda por Entorno Variable Básica o BVNS (*Basic Variable Neighborhood Search*).

---

*Inicializar*.

Seleccionar el conjunto de estructuras de entorno $\mathcal{N}_k$, para $k = 1, \ldots, k_{max}$.

Encontrar una solución inicial $S$.

Elegir una condición de parada.

*Iteraciones*.

Repetir la siguiente secuencia hasta que se cumpla la condición de parada:

(1) $k \leftarrow 1$.

(2) Repetir los siguientes pasos hasta que $k = k_{max}$:

(a) *Agitar*.
Generar al azar una solución $S'$ en el $k$-ésimo entorno de $S$ ($S' \in \mathcal{N}_k(S)$).

(b) *Búsqueda local*.
Aplicar alguna búsqueda local con $S'$ como solución inicial; denotar por $S''$ al óptimo local obtenido.

(c) *Movimiento o no*.
Si este óptimo local es mejor que la solución actual, entonces hacer $S \leftarrow S''$, y continuar la búsqueda con $\mathcal{N}_1$ ($k \leftarrow 1$); en otro caso, hacer $k \leftarrow k + 1$.

---

Figura 5.1: Búsqueda por Entorno Variable Básica.

Una Búsqueda por Entorno Variable combina cambios determinísticos y aleatorios de la estructura de entorno. Se introduce aleatoriedad en la agita-

ción para evitar que la búsqueda sea cíclica y repita las regiones exploradas. La condición de parada depende del problema considerado, aunque lo habitual es considerar un tiempo máximo de CPU o un número máximo de iteraciones.

Una estrategia ampliamente usada para obtener un conjunto de estructuras de entorno es considerar la existencia de una métrica $dis$ en el espacio de soluciones $D$, de modo que, a mayor índice $k$, mayor será el valor de la distancia. Así, puede definirse la $k$-ésima vecindad de una solución $S$, $N_k(S)$, como:

$$\mathcal{N}_k(S) = \{S' \in D : dis(S, S') \leq dis_k\}$$

De esta forma, se obtiene una serie de entornos anidados a los que puede llegarse partiendo de la primera vecindad de la solución según la siguiente expresión

$$\mathcal{N}_k(S) = \mathcal{N}_1(\mathcal{N}_{k-1}(S)).$$

Es decir, para movernos a la $k$-ésima vecindad debemos desplazarnos $k$ veces considerando la primera vecindad.

## 5.2 Aplicación al problema de selección de atributos

En esta sección describiremos las características de los métodos de la metaheurística VNS adaptados al problema de la selección de atributos.

1. *Estructuras de entorno.* La $k$-ésima vecindad de una solución $S$, $\mathcal{N}_k(S)$, está formada por todas las soluciones que pueden obtenerse desde $S$ al intercambiar $k$ atributos de la solución por $k$ atributos que no pertenecen a la solución.

   Sea $dis(S, S')$ la distancia entre las soluciones $S = \{X_1, ..., X_r\}$ y $S' = \{X'_1, ..., X'_r\}$ definida como sigue.

   $$dis(S, S') = r - |\{X_1, ..., X_r\} \bigcap \{X'_1, ..., X'_r\}|.$$

Entonces la $k$-ésima vecindad de la solución $S$, $\mathcal{N}_k(S)$, se expresa como sigue

$$\mathcal{N}_k(S) = \{S' : dis(S, S') \leq k\}$$

2. *Solución inicial.* Para obtener una buena solución de comienzo que hiciera más eficiente la búsqueda posterior, se emplearon dos métodos secuenciales simples de búsqueda local aplicados habitualmente al resolver el problema de selección de atributos: Búsqueda Secuencial hacia Adelante (*Sequential Forward Search* o SFS) y Eliminación Secuencial hacia Atrás (*Sequential Backward Elimination* o SBE). Las Figuras 5.2 y 5.3 muestran, respectivamente, los pseudocódigos de estas búsquedas. Esta búsqueda que alterna el uso de las estrategias SFS y SBE aparece en la literatura bajo el nombre de *Squential Forward Floating Selection* (SFFS) [116].

   Hemos aplicado secuencialmente ambas búsquedas para obtener la solución inicial. Los pasos del proceso seguidos son los siguientes:

   - Sea $S = \emptyset$.
   - Aplicar $SFS$ a la solución $S$. Sea $S'$ la solución obtenida.
   - Aplicar $SBE$ a la solución $S'$. Sea $S''$ la solución obtenida.
   - Si $S''$ es mejor que $S'$, hacer $S = S''$ y repetir desde el paso 2. En caso contrario, parar con $S'$ como solución

3. *Método de agitación.* Este método genera aleatoriamente una solución $S'$ de la $k$-ésima vecindad de $S$ ($S' \in \mathcal{N}_k(S)$). Sean $X_{i(1)}, ..., X_{i(k)}$ $k$ atributos presentes en la solución $S$ seleccionados aleatoriamente e $Y_{i(1)}, ..., Y_{i(k)}$ $k$ atributos no pertenecientes a la solución $S$ elegidos también de forma aleatoria. La solución $S'$ se obtiene del modo siguiente

$$S' = S \setminus \{X_{i(1)}, ..., X_{i(k)}\} \bigcup \{Y_{i(1)}, ..., Y_{i(k)}\}.$$

4. *Búsqueda local.* Consiste en aplicar secuencialmente a la solución $S'$ los métodos *Búsqueda Secuencial hacia Adelante* y *Eliminación Secuencial hacia Atrás*.

**Procedimiento** *Búsqueda Secuencial hacia Adelante*
{
**repetir**
  **para** $X_j \notin S$
    $f_j \leftarrow f(S \cup \{X_j\})$
  *sea* $j^* \leftarrow arg\,\text{máx}\{f_j\}$
  $S^* = S \cup \{X_{j^*}\}$
  $f^* \leftarrow f(S^*)$
  **si**($f^* > f$)
    $f \leftarrow f^*$
    $S \leftarrow S^*$
**hasta** ($f^* \leq f$)
}

Figura 5.2: Pseudocódigo de la Búsqueda Secuencial hacia Adelante.

5. *Criterio de parada.* En la *Búsqueda por Entorno Variable* consideramos que en una iteración se ejecutan los métodos de agitación y búsqueda local.

   La condición de parada se alcanza si el número de variables a intercambiar, $k$, es mayor que el mínimo entre el número de atributos pertenecientes y no pertenecientes a la solución.

## 5.3 Hibridación con la Búsqueda Tabú

La Búsqueda Tabú [57, 61, 59, 62] es un procedimiento heurístico de búsqueda por entornos que se caracteriza por considerar un historial selectivo, $\mathcal{H}$, de soluciones encontradas durante la búsqueda. El uso que se le da al historial $\mathcal{H}$ es variado (evita que la búsqueda se cicle, sirve para escapar de óptimos locales no globales y puede usarse para explorar regiones poco exploradas del espacio de soluciones).

Formalmente, en una Búsqueda Tabú se reemplaza la vecindad, $\mathcal{N}(S)$, de

**Procedimiento** *Eliminación Secuencial hacia Atrás*
{
**repetir**
  **para** $X_j \in S$
    $f_j \leftarrow f(S \setminus \{X_j\})$
  *sea* $j^* \leftarrow arg\,\text{máx}\{f_j\}$
  $S^* = S \setminus \{X_{j^*}\}$
  $f^* \leftarrow f(S^*)$
  **si**($f^* > f$)
    $f \leftarrow f^*$
    $S \leftarrow S^*$
**hasta** ($f^* \leq f$)
}

Figura 5.3: Eliminación Secuencial hacia Atrás.

la solución actual por otra modificada, que denotaremos $\mathcal{N}(\mathcal{H}, s)$. El historial $\mathcal{H}$ determina qué soluciones pueden ser alcanzadas y cuáles son prohibidas (tabú) al aplicar los movimientos disponibles.

Según la estrategia usada al gestionar el historial $\mathcal{H}$, se habla de memoria a corto, medio y largo plazo. En las memorias a corto plazo, $\mathcal{N}(\mathcal{H}, s)$, suele ser un subconjunto de $\mathcal{N}(s)$, y la lista tabú sirve para identificar los elementos de $\mathcal{N}(s)$ excluidos de $\mathcal{N}(\mathcal{H}, s)$. En las memorias a medio y largo plazo, $N(\mathcal{H}, s)$ puede contener soluciones no pertenecientes a la vecindad $\mathcal{N}(s)$. En este caso, habitualmente $\mathcal{N}(\mathcal{H}, s)$ se forma añadiendo a $\mathcal{N}(s)$ soluciones de alta calidad encontradas durante la búsqueda.

En la hibridación propuesta hacemos uso de memoria a corto plazo en el método de agitación. Así, los atributos $\{X_{i(1)}, ..., X_{i(k)}\} \bigcup \{Y_{i(1)}, \ldots, Y_{i(k)}\}$, implicados en la agitación, son añadidos a la lista tabú, y no podrán usarse en la agitación de la iteración siguiente. Denotaremos por $\text{VNS}_m$ a la variante de VNS que hace uso de memoria.

## 5.4 Resultados computacionales

### 5.4.1 Motivación y objetivos

La motivación principal de este apartado es analizar y estudiar el comportamiento de las estrategias VNS propuestas en este trabajo. Para llevar a cabo este análisis, se consideraron las medidas del porcentaje de aciertos, tamaño de los subconjuntos encontrados y tiempo de ejecución empleado.

La experiencia computacional se desarrolló en dos etapas. En la primera, se analizó el diseño de las estrategias VNS propuestas. Para ello, se compararon los resultados obtenidos por VNS y por el clasificador base sin selección de atributos, se comparó VNS con la búsqueda SFFS, ya que ésta se emplea en diferentes fases del VNS y, por último, se analizó la ventaja de la hibridación, comparando VNS con VNS$_m$. En la segunda etapa, se comparó VNS$_m$ con otras estrategias de selección de atributos presentes en la literatura: un Algoritmo Genético [64] y el FSSEBNA [76, 77]. Como medida para la comparación, se usaron el porcentaje de aciertos y el tamaño del subconjunto de atributos presentes en la mejor solución encontrada por cada procedimiento. Finalmente, se analizaron los tiempos de ejecución de todas las estrategias de selección de atributos presentadas.

Por tanto, los objetivos perseguidos son

- Examinar la aportación de las estrategias VNS comparando su rendimiento con el de los clasificadores bases con y sin selección de atributos.
- Analizar la aportación del VNS sobre el SFFS.
- Estudiar las diferencias de rendimiento de VNS y VNS$_m$.
- Comparar los resultados de la estrategia VNS$_m$ con los alcanzados con el AG y FSSEBNA.
- Comparar los tiempos de ejecución de la estrategia VNS$_m$ con los de SFFS, AG y FSSEBNA.

### 5.4.2 Descripción de los experimentos

Para los experimentos computacionales se usaron los clasificadores Naïve Bayes, IB1 y J48 de la librería Weka [146]. Se consideró evaluación validación cruzada 5x2cv [28]. Se aplicó el test estadístico F [4] para determinar si las diferencias obtenidas sobre cada conjunto de datos eran o no significativas. Siguiendo las recomendaciones de Demšar [26] se aplicó el test de Friedman [45, 46] para contrastar las diferencias de rendimiento de cada estrategia. En aquellos casos en los que las diferencias son estadísticamente significativas se realizaron comparaciones dos a dos empleando el test de Nemenyi [112] mediante el procedimiento de Hochberg [70].

Dado que el test F tiende a cometer errores tipo I altos, pues se aplica sobre los resultados obtenidos con un único conjunto de datos (y los distintos resultados no son independientes), se consideran intervalos de confianza conservadores al 95 % y al 99 %. Por otro lado, tal y como señala Demšar, el test de Friedman comete pocos errores de tipo I, de modo que los intervalos de confianza fueron del 90 % y 95 %.

Las pruebas se llevaron a cabo sobre problemas generados artificialmente (ver descripción en B.1) y en problemas reales obtenidos del repositorio de la UCI [113] (ver descripción en B.2).

Para la comparación del rendimiento de la estrategia VNS con el AG y FSSEBNA se impuso, como condición de parada en las estrategias AG y FSSEBNA, que ambas estrategias examinaran un total de 4000 soluciones. Para el FSSEBNA se fijó una población inicial de 1000 individuos. Siguiendo las recomendaciones encontradas en [76, 77], el número de individuos seleccionados por iteración fueron la mitad de la población inicial (500) y el tamaño de la nueva población se fijó en 999 individuos. En la selección de los individuos se consideró elitismo y se consideraron 3 generaciones. En realidad esta estrategia examina, con los parámetros considerados, un total de 3996 soluciones. Igualmente para el AG se consideró una población inicial de 1000 individuos y un total de 3 generaciones. Siguiendo las recomendaciones de Yang y Honavar [149] la probabilidad de cruce se fijó en 0,6 y la de mutación en 0,001.

### 5.4.3 Comparativa del rendimiento de los clasificadores con y sin estrategia VNS de selección de atributos

#### Problemas artificiales

En la Tabla 5.1 se muestra el porcentaje de aciertos y el número de atributos obtenidos por VNS y el algoritmo base sin selección de atributos aplicando el clasificador Naïve Bayes. Los resultados obtenidos aplicando VNS mejoran los obtenidos con el clasificador base sin selección de atributos. Estas diferencias son estadísticamente significativas en diversos problemas, aunque se concentran en aquellos con 5 atributos relevantes. En todos los casos, la reducción del conjunto de atributos es estadísticamente significativa.

Se aplicó el test Nemenyi con el procedimiento de Hochberg sobre todos los resultados obtenidos, sugiriéndonos que las diferencias son estadísticamente significativas al nivel $\alpha = 0{,}05$. Estudiando los problemas en mayor detalle, en función de la regla de clasificación y del número de atributos relevantes, se obtuvieron conclusiones similares: las diferencias son estadísticamente significativas con $\alpha = 0{,}05$. Estas conclusiones también se obtuvieron al analizar la reducción de la dimensionalidad de los problemas.

En la Tabla 5.2 están los resultados obtenidos con el clasificador IB1. Al igual que antes, el porcentaje de aciertos del clasificador mejora con el subconjunto de atributos seleccionado por el VNS. Sólo en un caso es inferior, no siendo esta diferencia estadísticamente significativa. Las diferencias significativas a favor de VNS se concentran en los problemas con 5 atributos relevantes, aunque también las hay en problemas con 10 atributos relevantes. En todos los casos, la reducción del conjunto de atributos es estadísticamente significativa.

El test de Nemenyi indica que existen diferencias estadísticamente significativas entre los resultados obtenidos por el clasificador con y sin selección de atributos al nivel $\alpha = 0{,}05$ en los distintos estudios realizados (considerando todos los resultados, diferenciando según la regla de clasificación y el número de atributos relevantes).

Los resultados obtenidos al usar el clasificador J48 se muestran en la Tabla 5.3. En este caso, los porcentajes de aciertos obtenidos con VNS mejoran los del clasificador base en todos los problemas menos en tres de ellos. Sin embargo, sólo en uno de los casos la diferencia es estadísticamente significativa. Con respecto al tamaño del subconjunto de atributos, como sucedió en los casos anteriores, todas las diferencias son estadísticamente significativas.

Aplicando el test de Nemenyi sobre todos los resultados, tanto en el porcentaje de aciertos como en el tamaño de los conjuntos de atributos encontrados, existen diferencias estadísticamente significativas en el rendimiento del clasificador con y sin selección de atributos al nivel $\alpha = 0{,}05$. Se obtienen los mismos resultados si el análisis se realiza diferenciando por regla de clasificación y número de atributos relevantes.

Tabla 5.1: Comparativa entre las metaheurísticas VNS y VNS$_m$, la búsqueda local SFFS y el clasificador base con Naïve Bayes. Porcentaje de acierto y reducción en conjuntos artificiales.

| conjuntos de datos | | | | Naïve Bayes | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | % aciertos | | | | # atributos | | | |
| l | r | $\sigma$ | d | VNS | VNS$_m$ | SFFS | base | VNS | VNS$_m$ | SFFS | base |
| 1 | 5 | 0 | 25 | 89,8 ± 3,3 | 90,0 ± 2,5 | 88,4 ± 6,2 | 83,9 ± 2,2 | 8,2 ± 2,3 | 7,9 ± 2,7 | 6,3 ± 1,3 | 25 ≪ |
| | | | 50 | 88,5 ± 2,9 | 88,8 ± 2,5 ≫ | 87,9 ± 5,9 | 77,6 ± 2,5 ≪ | 9,1 ± 3,0 | 8,6 ± 2,3 ≫ | 6,7 ± 1,2 | 50 ≪ |
| | | | 100 | 88,3 ± 3,6 | 88,5 ± 3,5 ≫ | 88,4 ± 3,7 | 72,5 ± 1,6 ≪ | 8,9 ± 1,7 | 8,5 ± 1,4 ≫ | 7,6 ± 1,6 | 100 ≪ |
| | | | 200 | 88,0 ± 3,2 | 88,3 ± 2,9 | 86,9 ± 6,3 | 68,1 ± 3,1 ≪ | 10,3 ± 4,3 | 9,5 ± 2,6 | 9,0 ± 1,7 | 200 ≪ |
| | | 0,1 | 25 | 90,6 ± 2,6 | 90,8 ± 1,9 | 88,8 ± 6,1 | 83,2 ± 2,2 | 6,8 ± 1,5 | 6,8 ± 1,5 | 6,0 ± 1,6 | 25 ≪ |
| | | | 50 | 88,0 ± 5,5 | 88,6 ± 5,3 | 88,9 ± 5,4 | 77,4 ± 2,5 ≪ | 7,7 ± 2,8 | 7,2 ± 2,1 | 6,8 ± 1,6 | 50 ≪ |
| | | | 100 | 87,6 ± 5,0 | 88,2 ± 3,3 | 88,5 ± 5,0 | 72,9 ± 2,7 ≪ | 9,7 ± 3,0 | 10,8 ± 2,7 ≪ | 7,8 ± 1,9 | 100 ≪ |
| | | | 200 | 88,3 ± 3,2 | 88,0 ± 4,5 | 86,0 ± 6,1 | 68,7 ± 3,3 ≪ | 10,4 ± 3,3 | 10,1 ± 5,2 | 8,1 ± 1,8 | 200 ≪ |
| | 10 | 0 | 25 | 82,8 ± 3,0 | 83,6 ± 3,1 | 76,2 ± 7,0 | 81,1 ± 3,2 | 13,5 ± 2,4 | 13,2 ± 2,3 | 7,2 ± 2,9 | 25 ≪ |
| | | | 50 | 80,5 ± 3,4 | 79,9 ± 2,8 | 71,6 ± 7,9 | 77,2 ± 2,2 | 15,5 ± 3,3 | 15,1 ± 4,0 | 7,2 ± 3,2 | 50 ≪ |
| | | | 100 | 77,2 ± 4,0 | 78,0 ± 3,3 | 71,2 ± 7,6 | 69,4 ± 3,0 | 19,2 ± 4,1 | 16,7 ± 5,9 | 7,9 ± 2,9 > | 100 ≪ |
| | | | 200 | 73,3 ± 5,4 | 73,8 ± 6,8 | 66,2 ± 8,5 | 64,9 ± 3,0 | 20,5 ± 10,1 | 19,1 ± 6,6 | 6,9 ± 3,2 | 200 ≪ |
| | | 0,1 | 25 | 83,2 ± 2,9 | 82,0 ± 2,5 | 74,2 ± 7,0 | 81,1 ± 3,8 | 12,9 ± 1,3 | 12,2 ± 2,3 | 6,8 ± 2,0 ≫ | 25 ≪ |
| | | | 50 | 81,1 ± 3,0 | 81,1 ± 3,2 | 68,7 ± 6,5 < | 77,3 ± 2,0 | 15,9 ± 3,4 | 15,4 ± 4,0 | 6,4 ± 2,2 ≫ | 50 ≪ |
| | | | 100 | 78,5 ± 5,5 | 77,0 ± 5,1 | 75,6 ± 6,8 | 70,0 ± 3,8 | 17,3 ± 3,6 | 17,9 ± 4,6 | 7,7 ± 1,9 ≫ | 100 ≪ |
| | | | 200 | 74,3 ± 2,6 | 73,2 ± 4,6 | 66,4 ± 5,0 | 64,8 ± 2,1 ≪ | 24,7 ± 7,2 | 20,1 ± 8,8 | 6,7 ± 2,8 | 200 ≪ |
| 2 | 5 | 0 | 25 | 89,3 ± 1,7 | 89,8 ± 1,8 | 88,7 ± 4,4 | 84,3 ± 2,2 < | 8,4 ± 3,0 | 8,0 ± 2,7 | 6,4 ± 1,0 | 25 ≪ |
| | | | 50 | 88,2 ± 2,8 | 88,8 ± 2,7 | 89,5 ± 2,0 | 79,8 ± 1,4 < | 9,3 ± 1,9 | 7,6 ± 2,0 | 7,0 ± 1,3 | 50 ≪ |
| | | | 100 | 88,5 ± 1,9 | 88,6 ± 2,5 | 86,7 ± 4,9 | 73,3 ± 2,9 ≪ | 10,2 ± 3,6 | 9,1 ± 4,1 | 6,4 ± 1,8 | 100 ≪ |
| | | | 200 | 84,6 ± 5,2 | 85,8 ± 3,8 | 83,9 ± 5,1 | 69,7 ± 1,7 < | 12,4 ± 3,4 | 11,9 ± 4,0 | 7,2 ± 3,4 | 200 ≪ |
| | | 0,1 | 25 | 89,5 ± 2,5 | 90,1 ± 2,3 | 89,0 ± 5,0 | 84,2 ± 2,8 < | 7,9 ± 2,6 | 7,3 ± 1,8 | 6,1 ± 0,9 | 25 ≪ |
| | | | 50 | 89,1 ± 2,2 | 88,4 ± 2,7 | 87,5 ± 4,4 | 79,2 ± 1,6 ≪ | 8,9 ± 3,3 | 8,7 ± 2,3 | 7,2 ± 1,2 | 50 ≪ |
| | | | 100 | 86,8 ± 4,6 | 88,6 ± 3,9 | 88,1 ± 3,8 | 73,2 ± 2,2 < | 10,8 ± 4,1 | 7,9 ± 1,4 | 6,9 ± 1,6 | 100 ≪ |
| | | | 200 | 86,1 ± 4,0 | 85,1 ± 5,5 | 84,1 ± 5,0 | 69,6 ± 1,6 ≪ | 10,1 ± 2,8 | 11,6 ± 6,5 | 6,6 ± 2,0 | 200 ≪ |
| | 10 | 0 | 25 | 83,1 ± 3,5 | 82,2 ± 2,9 | 76,7 ± 4,6 | 80,3 ± 2,9 | 12,0 ± 1,8 | 13,3 ± 2,8 | 7,6 ± 2,3 | 25 ≪ |
| | | | 50 | 81,8 ± 3,5 | 80,6 ± 2,7 | 73,6 ± 6,3 | 74,8 ± 1,8 | 14,4 ± 2,6 | 14,3 ± 2,8 | 7,2 ± 3,2 > | 50 ≪ |
| | | | 100 | 75,1 ± 4,3 | 75,2 ± 3,7 | 68,7 ± 3,6 | 69,0 ± 2,7 | 19,6 ± 4,1 | 17,9 ± 6,8 | 7,3 ± 2,8 > | 100 ≪ |
| | | | 200 | 72,8 ± 4,4 | 72,4 ± 3,9 | 67,7 ± 4,0 < | 64,9 ± 3,3 < | 18,6 ± 8,5 | 17,7 ± 8,4 | 7,0 ± 2,7 > | 200 ≪ |
| | | 0,1 | 25 | 81,5 ± 4,7 | 81,2 ± 4,6 < | 71,4 ± 5,5 | 80,7 ± 2,2 | 11,6 ± 1,8 | 12,0 ± 2,1 | 6,1 ± 2,8 | 25 ≪ |
| | | | 50 | 77,1 ± 5,4 | 74,5 ± 4,5 | 70,9 ± 4,6 | 74,4 ± 2,1 | 13,9 ± 5,6 | 12,4 ± 5,1 | 6,2 ± 2,3 | 50 ≪ |
| | | | 100 | 78,0 ± 4,7 | 77,2 ± 5,1 | 68,6 ± 6,7 < | 69,0 ± 2,6 | 18,4 ± 5,1 | 17,1 ± 5,5 | 5,7 ± 2,9 > | 100 ≪ |
| | | | 200 | 72,7 ± 2,5 | 71,7 ± 3,7 | 66,2 ± 5,4 | 65,5 ± 3,0 ≪ | 22,5 ± 8,7 | 19,6 ± 9,3 | 6,3 ± 2,9 > | 200 ≪ |

$l$-regla de clasificación; $r$-número de atributos relevantes; $\sigma$-ruido Gaussiano;VNS-Variable Neighbourhood Search; VNS$_m$-VNS híbrida con memoria; SFFS-Sequential Forward Floating Selection; > y ≫ mejora significativa en un 95 y 99 %; < y ≪ empeora significativamente.

Tabla 5.2: Comparativa entre las metaheurísticas VNS y VNS$_m$, la búsqueda local SFFS y el clasificador base con IB1. Porcentaje de acierto y reducción en conjuntos artificiales.

| conjuntos de datos | | | | IB1 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | % aciertos | | | | # atributos | | | |
| l | r | $\sigma$ | d | VNS | VNS$_m$ | SFFS | base | VNS | VNS$_m$ | SFFS | base |
| 1 | 5 | 0 | 25 | $84{,}2 \pm 3{,}5$ | $84{,}6 \pm 3{,}5$ | $82{,}0 \pm 7{,}7$ | $69{,}7 \pm 2{,}2$ $\ll$ | $5{,}8 \pm 1{,}1$ | $5{,}7 \pm 1{,}1$ | $5{,}1 \pm 1{,}3$ | 25 $\ll$ |
| | | | 50 | $80{,}3 \pm 3{,}8$ | $80{,}0 \pm 4{,}0$ | $73{,}8 \pm 12{,}8$ | $61{,}9 \pm 3{,}3$ $\ll$ | $7{,}1 \pm 1{,}9$ | $7{,}1 \pm 1{,}9$ | $4{,}9 \pm 2{,}7$ | 50 $\ll$ |
| | | | 100 | $77{,}7 \pm 7{,}3$ | $77{,}7 \pm 7{,}2$ | $74{,}8 \pm 11{,}1$ | $55{,}4 \pm 1{,}6$ $\ll$ | $9{,}1 \pm 4{,}2$ | $9{,}6 \pm 5{,}6$ | $5{,}7 \pm 2{,}8$ | 100 $\ll$ |
| | | | 200 | $75{,}8 \pm 5{,}5$ | $74{,}5 \pm 7{,}7$ | $67{,}3 \pm 12{,}0$ | $54{,}6 \pm 2{,}2$ $<$ | $11{,}8 \pm 5{,}1$ | $11{,}3 \pm 7{,}6$ | $4{,}2 \pm 2{,}7$ | 200 $\ll$ |
| | | 0,1 | 25 | $84{,}2 \pm 3{,}5$ | $84{,}4 \pm 3{,}3$ | $82{,}0 \pm 7{,}8$ | $69{,}9 \pm 2{,}4$ $\ll$ | $5{,}9 \pm 0{,}9$ | $5{,}8 \pm 0{,}6$ | $5{,}1 \pm 1{,}3$ | 25 $\ll$ |
| | | | 50 | $82{,}5 \pm 5{,}3$ | $81{,}7 \pm 4{,}5$ | $78{,}0 \pm 9{,}0$ | $63{,}6 \pm 3{,}5$ $<$ | $6{,}1 \pm 1{,}3$ | $7{,}0 \pm 1{,}7$ | $5{,}5 \pm 1{,}2$ | 50 $\ll$ |
| | | | 100 | $80{,}8 \pm 4{,}4$ | $79{,}6 \pm 3{,}2$ | $72{,}1 \pm 12{,}6$ | $55{,}3 \pm 1{,}5$ $\ll$ | $7{,}7 \pm 2{,}4$ | $8{,}2 \pm 2{,}2$ | $5{,}6 \pm 2{,}6$ | 100 $\ll$ |
| | | | 200 | $74{,}9 \pm 9{,}6$ | $75{,}6 \pm 9{,}1$ | $73{,}7 \pm 9{,}5$ | $54{,}8 \pm 2{,}7$ $<$ | $10{,}4 \pm 4{,}1$ | $8{,}8 \pm 2{,}3$ | $5{,}7 \pm 2{,}5$ | 200 $\ll$ |
| | 10 | 0 | 25 | $71{,}3 \pm 5{,}7$ | $70{,}4 \pm 5{,}4$ | $66{,}5 \pm 7{,}1$ | $68{,}0 \pm 2{,}5$ | $9{,}8 \pm 3{,}0$ | $9{,}9 \pm 2{,}8$ | $6{,}2 \pm 2{,}1$ | 25 $\ll$ |
| | | | 50 | $65{,}8 \pm 5{,}2$ | $66{,}2 \pm 3{,}8$ | $62{,}2 \pm 5{,}5$ | $62{,}2 \pm 2{,}6$ | $12{,}2 \pm 6{,}4$ | $11{,}4 \pm 5{,}5$ | $5{,}7 \pm 2{,}6$ | 50 $\ll$ |
| | | | 100 | $63{,}4 \pm 5{,}9$ | $61{,}8 \pm 5{,}2$ | $55{,}2 \pm 6{,}2$ | $59{,}8 \pm 2{,}8$ | $18{,}2 \pm 8{,}9$ | $19{,}7 \pm 11{,}1$ | $3{,}6 \pm 3{,}7$ | 100 $\ll$ |
| | | | 200 | $59{,}4 \pm 4{,}8$ | $57{,}3 \pm 5{,}6$ | $53{,}2 \pm 6{,}0$ | $53{,}6 \pm 2{,}3$ | $21{,}2 \pm 9{,}1$ | $16{,}3 \pm 10{,}7$ | $2{,}5 \pm 2{,}7$ | 200 $\ll$ |
| | | 0,1 | 25 | $68{,}1 \pm 5{,}8$ | $69{,}2 \pm 6{,}6$ | $61{,}5 \pm 7{,}7$ | $68{,}9 \pm 3{,}4$ | $8{,}4 \pm 4{,}3$ | $10{,}2 \pm 4{,}9$ | $4{,}2 \pm 2{,}3$ $>$ | 25 $\ll$ |
| | | | 50 | $68{,}2 \pm 3{,}4$ | $66{,}1 \pm 7{,}5$ | $60{,}6 \pm 8{,}0$ | $61{,}0 \pm 1{,}6$ | $12{,}8 \pm 5{,}2$ | $11{,}9 \pm 2{,}8$ | $4{,}4 \pm 2{,}3$ $>$ | 50 $\ll$ |
| | | | 100 | $63{,}0 \pm 3{,}9$ | $63{,}2 \pm 5{,}5$ | $55{,}6 \pm 6{,}4$ | $58{,}5 \pm 2{,}2$ | $20{,}5 \pm 7{,}1$ | $19{,}8 \pm 6{,}1$ | $4{,}0 \pm 3{,}4$ $\gg$ | 100 $\ll$ |
| | | | 200 | $60{,}2 \pm 3{,}1$ | $57{,}8 \pm 3{,}5$ | $51{,}3 \pm 4{,}5$ | $53{,}5 \pm 2{,}7$ | $23{,}1 \pm 10{,}6$ | $19{,}7 \pm 9{,}5$ | $2{,}6 \pm 1{,}8$ | 200 $\ll$ |
| 2 | 5 | 0 | 25 | $82{,}5 \pm 3{,}4$ | $83{,}0 \pm 3{,}5$ | $82{,}9 \pm 3{,}6$ | $70{,}2 \pm 2{,}3$ $\ll$ | $6{,}8 \pm 1{,}6$ | $6{,}1 \pm 1{,}2$ | $5{,}7 \pm 1{,}2$ | 25 $\ll$ |
| | | | 50 | $80{,}6 \pm 6{,}0$ | $80{,}2 \pm 6{,}3$ | $81{,}1 \pm 4{,}4$ | $63{,}0 \pm 1{,}8$ $<$ | $7{,}1 \pm 2{,}8$ | $7{,}0 \pm 2{,}6$ | $5{,}3 \pm 1{,}1$ | 50 $\ll$ |
| | | | 100 | $75{,}8 \pm 4{,}5$ | $76{,}6 \pm 3{,}9$ | $76{,}6 \pm 9{,}0$ | $60{,}5 \pm 2{,}7$ $<$ | $9{,}0 \pm 1{,}8$ | $8{,}1 \pm 3{,}0$ | $5{,}4 \pm 2{,}3$ | 100 $\ll$ |
| | | | 200 | $76{,}2 \pm 4{,}4$ | $76{,}1 \pm 4{,}9$ | $74{,}4 \pm 7{,}3$ | $56{,}1 \pm 2{,}1$ $\ll$ | $7{,}2 \pm 2{,}2$ | $7{,}8 \pm 3{,}1$ | $5{,}6 \pm 2{,}0$ | 200 $\ll$ |
| | | 0,1 | 25 | $85{,}3 \pm 3{,}8$ | $83{,}2 \pm 4{,}5$ | $83{,}4 \pm 4{,}5$ | $70{,}4 \pm 1{,}8$ $\ll$ | $5{,}8 \pm 1{,}0$ | $5{,}9 \pm 1{,}6$ | $4{,}9 \pm 1{,}1$ | 25 $\ll$ |
| | | | 50 | $81{,}0 \pm 4{,}7$ | $81{,}6 \pm 4{,}1$ | $80{,}0 \pm 6{,}3$ | $62{,}4 \pm 2{,}6$ $<$ | $6{,}6 \pm 2{,}4$ | $6{,}1 \pm 1{,}5$ | $5{,}3 \pm 1{,}9$ | 50 $\ll$ |
| | | | 100 | $77{,}8 \pm 7{,}4$ | $79{,}1 \pm 3{,}7$ | $80{,}0 \pm 3{,}9$ | $59{,}2 \pm 3{,}0$ $\ll$ | $7{,}7 \pm 3{,}0$ | $21{,}1 \pm 10{,}4$ | $4{,}3 \pm 3{,}4$ | 100 $\ll$ |
| | | | 200 | $75{,}5 \pm 3{,}8$ | $75{,}4 \pm 3{,}6$ | $75{,}3 \pm 3{,}6$ | $55{,}8 \pm 2{,}0$ $\ll$ | $6{,}9 \pm 1{,}9$ | $16{,}8 \pm 8{,}7$ | $3{,}5 \pm 2{,}8$ | 200 $<$ |
| | 10 | 0 | 25 | $66{,}9 \pm 8{,}6$ | $67{,}0 \pm 7{,}6$ | $58{,}0 \pm 7{,}3$ | $65{,}1 \pm 2{,}7$ | $9{,}5 \pm 4{,}4$ | $12{,}8 \pm 5{,}1$ | $2{,}5 \pm 2{,}0$ | 25 $<$ |
| | | | 50 | $64{,}2 \pm 5{,}1$ | $65{,}6 \pm 4{,}9$ | $56{,}1 \pm 5{,}1$ | $58{,}8 \pm 2{,}8$ | $10{,}1 \pm 4{,}4$ | $10{,}7 \pm 5{,}2$ | $2{,}8 \pm 2{,}9$ | 50 $\ll$ |
| | | | 100 | $61{,}5 \pm 6{,}1$ | $62{,}9 \pm 6{,}1$ | $57{,}2 \pm 5{,}8$ | $59{,}0 \pm 2{,}7$ $\ll$ | $14{,}3 \pm 5{,}9$ | $14{,}0 \pm 5{,}7$ | $3{,}5 \pm 3{,}5$ | 100 $\ll$ |
| | | | 200 | $58{,}5 \pm 4{,}5$ | $60{,}2 \pm 3{,}7$ | $55{,}8 \pm 6{,}3$ | $55{,}1 \pm 2{,}7$ $<$ | $18{,}4 \pm 6{,}4$ | $16{,}3 \pm 8{,}1$ | $3{,}6 \pm 2{,}8$ | 200 $\ll$ |
| | | 0,1 | 25 | $70{,}0 \pm 3{,}8$ | $70{,}3 \pm 4{,}7$ | $61{,}8 \pm 8{,}0$ $<$ | $65{,}8 \pm 2{,}2$ $<$ | $9{,}9 \pm 2{,}8$ | $10{,}1 \pm 1{,}6$ | $4{,}6 \pm 3{,}0$ | 25 $\ll$ |
| | | | 50 | $64{,}8 \pm 6{,}4$ | $64{,}3 \pm 4{,}9$ | $55{,}6 \pm 5{,}2$ | $59{,}4 \pm 3{,}7$ | $10{,}1 \pm 4{,}7$ | $7{,}8 \pm 3{,}7$ | $2{,}9 \pm 2{,}6$ | 50 $\ll$ |
| | | | 100 | $61{,}1 \pm 3{,}8$ | $60{,}7 \pm 3{,}9$ | $54{,}6 \pm 7{,}8$ | $58{,}9 \pm 2{,}8$ | $19{,}9 \pm 9{,}2$ | $6{,}1 \pm 1{,}4$ | $5{,}1 \pm 1{,}0$ | 100 $\ll$ |
| | | | 200 | $59{,}9 \pm 4{,}1$ | $59{,}2 \pm 5{,}8$ | $52{,}8 \pm 7{,}1$ | $54{,}8 \pm 2{,}0$ | $17{,}0 \pm 9{,}1$ | $6{,}7 \pm 2{,}1$ | $5{,}5 \pm 1{,}7$ | 200 $\ll$ |

$l$-regla de clasificación; $r$-número de atributos relevantes; $\sigma$-ruido Gaussiano;VNS-Variable Neighbourhood Search; VNS$_m$-VNS híbrida con memoria; SFFS-Sequential Forward Floating Selection; $>$ y $\gg$ mejora significativa en un 95 y 99 %; $<$ y $\ll$ empeora significativamente.

### Problemas reales

En la Tabla 5.4 se hallan los resultados obtenidos con los problemas del repositorio de la UCI considerando el clasificador Naïve Bayes. Los porcentajes de aciertos aplicando VNS son mejores o peores dependiendo del problema considerado. Las diferencias son estadísticamente significativas en 7 conjuntos, siendo en 5 de ellos favorables a VNS. En todos los casos las diferencias en el tamaño del subconjunto de atributos son estadísticamente significativas.

El test de Nemenyi aplicado sobre todos los resultados muestra que el rendimiento del clasificador con y sin selección de atributos es similar (no hay diferencias estadísticamente significativas). Sin embargo, sí existen diferencias en lo concerniente a la reducción de atributos al nivel $\alpha = 0{,}05$.

Con el clasificador IB1 (Tabla 5.5) los resultados, al igual que antes, son favorables o no a VNS dependiendo del conjunto de datos considerado. Hay 4 casos en los que las diferencias en el porcentaje de aciertos son estadísticamente significativas y favorables a VNS. La reducción de la dimensionalidad sí es estadísticamente significativa.

El análisis conjunto sobre todas las bases de datos indica que la diferencia en el número de atributos es significativa al nivel $\alpha = 0{,}05$.

Las conclusiones con el clasificador J48 expuestos en la Tabla 5.6, son similares a los casos anteriores. Sólo existe una diferencia significativa en contra de VNS cuando se considera el porcentaje de aciertos. La reducción de la dimensionalidad también es estadísticamente significativa.

El análisis conjunto sobre todas las bases de datos aplicando el test de Nemenyi indica que la diferencia en el número de atributos es significativa al nivel $\alpha = 0{,}05$.

## 5.4.4 Comparativa del rendimiento del VNS y el SFFS

Dado que en diferentes fases de la estrategia VNS propuesta se emplea la búsqueda SFFS, una pregunta que surge de manera natural es si nuestro diseño de VNS mejora el comportamiento de SFFS. En la presente sección analizamos y comparamos el comportamiento de VNS y SFFS sobre problemas artificiales y reales.

Tabla 5.3: Comparativa entre las metaheurísticas VNS y VNS$_m$, la búsqueda local SFFS y el clasificador base con J48. Porcentaje de acierto y reducción en conjuntos artificiales.

| conjuntos de datos | | | | J48 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | % aciertos | | | | # atributos | | | |
| l | r | $\sigma$ | d | VNS | VNS$_t$ | SFFS | base | VNS | VNS$_t$ | SFFS | base |
| 1 | 5 | 0 | 25 | 77,1 ± 3,9 | 77,5 ± 3,9 | 74,7 ± 4,4 | 79,1 ± 2,8 | 6,6 ± 2,2 | 6,3 ± 1,3 | 5,0 ± 1,9 | 25 ≪ |
| | | | 50 | 76,8 ± 3,8 | 77,3 ± 4,7 | 74,2 ± 4,1 | 75,8 ± 2,1 | 8,0 ± 1,9 | 8,0 ± 1,9 | 5,7 ± 1,6 | 50 ≪ |
| | | | 100 | 75,5 ± 4,6 | 75,5 ± 4,0 | 73,9 ± 4,3 | 72,9 ± 2,9 | 7,6 ± 1,3 | 7,6 ± 2,3 | 6,8 ± 1,8 | 100 ≪ |
| | | | 200 | 74,7 ± 4,5 | 74,3 ± 3,4 | 74,7 ± 3,4 | 70,6 ± 2,4 | 9,7 ± 1,9 | 8,9 ± 1,9 | 7,6 ± 1,9 | 200 ≪ |
| | | 0,1 | 25 | 76,6 ± 4,0 | 76,0 ± 4,3 | 76,2 ± 4,5 | 76,4 ± 3,1 | 5,2 ± 1,3 | 4,7 ± 1,1 | 4,5 ± 1,2 | 25 ≪ |
| | | | 50 | 75,2 ± 4,0 | 76,1 ± 4,6 | 77,0 ± 3,3 | 75,4 ± 2,3 | 7,5 ± 1,6 | 7,2 ± 2,1 | 5,7 ± 0,9 | 50 ≪ |
| | | | 100 | 73,7 ± 2,2 | 73,9 ± 2,9 | 73,4 ± 3,0 | 72,3 ± 2,2 | 8,2 ± 2,2 | 8,7 ± 2,4 | 6,0 ± 1,1 | 100 ≪ |
| | | | 200 | 74,1 ± 4,6 | 74,9 ± 4,4 | 73,8 ± 5,1 | 70,2 ± 3,3 | 10,0 ± 2,4 | 9,2 ± 1,7 | 7,0 ± 1,6 ≫ | 200 ≪ |
| | 10 | 0 | 25 | 67,7 ± 3,6 | 67,4 ± 4,7 | 65,7 ± 3,8 | 67,4 ± 2,2 | 6,9 ± 2,8 | 7,5 ± 3,1 | 5,0 ± 2,2 | 25 ≪ |
| | | | 50 | 65,8 ± 3,1 | 66,4 ± 3,1 | 64,2 ± 3,7 | 65,2 ± 2,7 | 8,3 ± 2,8 | 8,0 ± 3,1 | 5,2 ± 2,3 | 50 ≪ |
| | | | 100 | 66,7 ± 2,7 | 65,0 ± 3,6 | 63,2 ± 3,9 | 63,2 ± 2,3 | 10,8 ± 2,6 | 10,7 ± 3,9 | 5,9 ± 2,6 | 100 ≪ |
| | | | 200 | 62,5 ± 3,7 | 62,4 ± 5,1 | 61,6 ± 4,8 | 61,9 ± 2,8 | 12,5 ± 2,6 | 12,6 ± 2,4 | 8,8 ± 2,3 | 200 ≪ |
| | | 0,1 | 25 | 66,6 ± 2,2 | 65,5 ± 3,8 | 65,5 ± 2,4 | 66,5 ± 2,2 | 8,8 ± 2,9 | 7,9 ± 2,4 | 5,6 ± 1,7 | 25 ≪ |
| | | | 50 | 64,0 ± 4,4 | 63,5 ± 4,3 | 62,2 ± 4,3 | 63,0 ± 4,0 | 9,8 ± 3,3 | 9,4 ± 3,6 | 6,0 ± 2,9 > | 50 ≪ |
| | | | 100 | 65,8 ± 4,4 | 64,8 ± 4,0 | 62,7 ± 6,0 | 63,2 ± 2,8 | 11,3 ± 3,5 | 11,6 ± 4,7 | 7,0 ± 2,4 | 100 ≪ |
| | | | 200 | 63,2 ± 3,6 | 63,1 ± 4,3 | 62,2 ± 4,1 | 60,9 ± 3,4 | 13,4 ± 3,6 | 12,9 ± 3,6 | 9,2 ± 2,8 > | 200 ≪ |
| 2 | 5 | 0 | 25 | 82,4 ± 2,7 | 82,4 ± 2,7 | 81,2 ± 4,2 | 78,5 ± 2,9 | 6,5 ± 1,4 | 6,5 ± 1,4 | 5,6 ± 0,8 | 25 ≪ |
| | | | 50 | 81,3 ± 3,5 | 80,7 ± 3,6 | 78,9 ± 3,7 | 77,3 ± 3,6 < | 6,5 ± 1,0 | 7,3 ± 1,8 | 6,2 ± 1,0 | 50 ≪ |
| | | | 100 | 80,0 ± 2,7 | 80,5 ± 3,4 | 78,1 ± 4,4 ≪ | 75,4 ± 3,0 | 8,8 ± 2,5 | 8,4 ± 1,9 | 7,0 ± 1,2 | 100 ≪ |
| | | | 200 | 77,8 ± 4,5 | 79,8 ± 3,1 | 77,4 ± 4,8 | 73,2 ± 3,8 | 9,8 ± 2,0 | 8,5 ± 2,0 | 7,9 ± 2,1 > | 200 ≪ |
| | | 0,1 | 25 | 79,4 ± 3,4 | 79,0 ± 3,0 | 77,0 ± 4,7 | 76,8 ± 4,5 | 5,9 ± 2,1 | 5,2 ± 2,2 | 4,7 ± 2,4 | 25 ≪ |
| | | | 50 | 78,7 ± 4,1 | 79,9 ± 3,9 | 79,7 ± 3,6 | 78,9 ± 3,3 | 6,4 ± 2,0 | 6,0 ± 1,8 | 5,3 ± 1,6 | 50 ≪ |
| | | | 100 | 78,9 ± 1,5 | 78,4 ± 1,5 | 78,7 ± 3,0 | 75,5 ± 3,5 | 7,5 ± 1,7 | 7,9 ± 1,7 | 6,4 ± 1,3 | 100 ≪ |
| | | | 200 | 78,2 ± 5,7 | 80,2 ± 3,8 | 77,8 ± 5,3 | 72,8 ± 3,8 | 8,7 ± 2,2 | 8,7 ± 1,9 | 7,6 ± 1,8 | 200 ≪ |
| | 10 | 0 | 25 | 68,8 ± 1,8 | 67,6 ± 3,6 | 67,5 ± 2,1 | 67,4 ± 3,8 | 7,0 ± 1,4 | 7,5 ± 2,8 | 6,4 ± 1,4 | 25 ≪ |
| | | | 50 | 67,8 ± 4,2 | 67,6 ± 2,4 | 67,1 ± 3,2 | 66,3 ± 3,5 | 9,1 ± 2,0 | 9,0 ± 2,2 | 6,7 ± 0,9 | 50 ≪ |
| | | | 100 | 65,0 ± 2,2 | 66,1 ± 3,6 | 65,5 ± 3,0 | 64,4 ± 3,2 | 11,2 ± 1,9 | 12,2 ± 3,7 | 8,1 ± 1,1 | 100 ≪ |
| | | | 200 | 63,7 ± 4,1 | 63,7 ± 4,9 | 63,8 ± 3,7 | 60,8 ± 2,1 | 13,2 ± 3,4 | 12,7 ± 3,6 | 8,9 ± 2,0 | 200 ≪ |
| | | 0,1 | 25 | 67,0 ± 1,9 | 66,8 ± 2,5 | 64,1 ± 3,1 | 66,0 ± 3,6 | 7,2 ± 1,6 | 8,1 ± 2,6 | 4,7 ± 2,2 | 25 ≪ |
| | | | 50 | 65,4 ± 5,2 | 64,8 ± 5,9 | 64,5 ± 3,8 | 64,6 ± 4,2 | 9,4 ± 2,8 | 9,0 ± 2,4 | 6,2 ± 2,7 | 50 ≪ |
| | | | 100 | 63,6 ± 2,8 | 64,9 ± 5,0 | 64,8 ± 4,6 | 63,5 ± 3,5 | 11,8 ± 3,5 | 9,7 ± 3,1 | 7,3 ± 2,3 | 100 ≪ |
| | | | 200 | 62,8 ± 4,0 | 63,4 ± 4,6 | 63,1 ± 5,7 | 62,0 ± 2,5 | 15,6 ± 3,4 | 12,6 ± 3,4 | 8,8 ± 3,8 | 200 ≪ |

$l$-regla de clasificación; $r$-número de atributos relevantes; $\sigma$-ruido Gaussiano;VNS-Variable Neighbourhood Search; VNS$_m$-VNS híbrida con memoria; SFFS-Sequential Forward Floating Selection; > y ≫ mejora significativa en un 95 y 99 %; < y ≪ empeora significativamente.

Tabla 5.4: Comparativa entre las metaheurísticas VNS y VNS$_m$, la búsqueda local SFFS y el clasificador base con Naïve Bayes. Porcentaje de acierto y reducción en conjuntos reales.

| cdd | Naïve Bayes | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | % aciertos | | | | # atributos | | | |
| | VNS | VNS$_m$ | SFFS | base | VNS | VNS$_m$ | SFFS | base |
| pdi | 77,8 ± 1,8 | 77,5 ± 2,0 | 77,1 ± 2,1 | 75,1 ± 1,7 | 4,0 ± 0,9 | 3,8 ± 0,9 | 3,4 ± 1,0 | 9 ≪ |
| gla | 55,1 ± 5,0 | 57,1 ± 3,0 ≫ | 55,7 ± 4,7 | 50,7 ± 3,8 | 3,6 ± 1,7 | 3,5 ± 1,5 | 3,0 ± 0,9 | 10 ≪ |
| bca | 69,0 ± 3,6 | 68,5 ± 3,1 | 68,1 ± 3,3 | 71,8 ± 2,6 | 3,2 ± 1,2 | 3,4 ± 1,5 | 2,3 ± 1,4 | 10 ≪ |
| bwi | 95,9 ± 0,9 | 96,1 ± 1,2 | 95,8 ± 1,2 | 97,3 ± 0,5 | 4,2 ± 0,8 | 4,2 ± 0,6 | 3,9 ± 0,9 | 10 ≪ |
| win | 96,1 ± 2,8 | 96,1 ± 2,7 | 94,4 ± 2,4 | 97,0 ± 2,5 | 5,4 ± 1,3 | 5,3 ± 1,3 | 4,0 ± 0,9 | 14 ≪ |
| hst | 83,0 ± 1,8 | 83,5 ± 1,7 | 78,1 ± 5,7 | 84,4 ± 2,2 | 6,9 ± 2,4 | 6,5 ± 1,8 | 3,5 ± 1,6 > | 14 ≪ |
| hhu | 81,0 ± 2,2 | 81,6 ± 2,9 | 79,4 ± 3,2 | 84,7 ± 1,6 > | 4,4 ± 1,6 | 4,8 ± 1,8 | 3,3 ± 1,6 | 14 ≪ |
| hcl | 80,5 ± 4,5 | 80,5 ± 4,3 | 79,9 ± 5,7 | 83,4 ± 3,2 | 6,0 ± 2,2 | 5,6 ± 1,9 | 4,2 ± 1,8 | 14 < |
| vow | 66,8 ± 2,6 | 66,7 ± 3,1 | 64,8 ± 3,3 | 61,4 ± 2,3 ≪ | 7,1 ± 1,4 | 6,9 ± 1,4 | 6,2 ± 1,1 | 14 ≪ |
| cra | 85,6 ± 1,4 | 85,5 ± 1,3 | 85,3 ± 1,3 | 81,1 ± 1,4 | 5,2 ± 1,4 | 5,2 ± 1,4 | 3,9 ± 2,4 | 16 ≪ |
| ptu | 42,2 ± 3,2 | 42,4 ± 3,6 | 40,9 ± 3,7 | 46,9 ± 2,5 | 9,5 ± 2,3 | 9,9 ± 2,6 | 7,9 ± 1,7 | 18 ≪ |
| vot | 95,1 ± 0,9 | 95,1 ± 0,9 | 95,1 ± 0,9 | 90,0 ± 1,2 < | 2,2 ± 1,1 | 2,2 ± 1,1 | 2,0 ± 1,2 | 17 ≪ |
| lym | 76,6 ± 2,6 | 77,6 ± 3,3 | 75,0 ± 3,3 | 80,8 ± 3,5 > | 4,4 ± 2,1 | 5,5 ± 3,5 | 3,0 ± 1,2 | 19 ≪ |
| veh | 59,4 ± 3,5 | 59,3 ± 3,3 | 57,1 ± 3,9 < | 58,7 ± 2,2 | 7,8 ± 2,6 | 7,8 ± 2,6 | 4,9 ± 2,1 | 19 ≪ |
| hep | 81,8 ± 4,4 | 81,8 ± 4,8 | 80,5 ± 4,1 | 84,6 ± 4,9 | 4,2 ± 3,1 | 3,5 ± 2,0 | 2,2 ± 0,8 | 20 ≪ |
| hco | 82,8 ± 4,0 | 82,8 ± 4,0 | 83,0 ± 3,1 | 80,1 ± 2,2 | 5,5 ± 1,6 | 5,5 ± 1,7 | 3,6 ± 1,3 | 23 ≪ |
| aut | 61,9 ± 5,0 | 61,5 ± 6,1 | 59,6 ± 6,1 | 59,6 ± 7,0 | 6,3 ± 2,1 | 6,0 ± 1,6 | 4,8 ± 1,5 | 26 ≪ |
| abp | 96,5 ± 0,3 | 96,5 ± 0,3 | 96,5 ± 0,3 | 94,4 ± 0,5 ≪ | 2,3 ± 0,8 | 2,3 ± 0,8 | 2,3 ± 0,8 | 30 ≪ |
| ion | 91,1 ± 3,5 | 91,1 ± 2,9 | 90,8 ± 2,6 | 91,1 ± 2,5 | 7,8 ± 2,5 | 7,2 ± 1,5 | 5,7 ± 0,8 | 35 ≪ |
| soy | 89,3 ± 1,8 | 89,9 ± 2,4 | 89,2 ± 1,9 | 91,2 ± 1,2 | 15,4 ± 2,2 | 16,5 ± 2,4 < | 11,8 ± 1,0 | 36 ≪ |
| kvk | 94,3 ± 0,4 | 94,3 ± 0,4 | 94,3 ± 0,4 | 87,5 ± 0,9 ≪ | 5,6 ± 1,3 | 5 | 5 | 37 ≪ |
| ann | 90,7 ± 2,3 | 90,1 ± 1,8 | 89,0 ± 0,6 | 91,5 ± 0,6 | 6,9 ± 3,1 | 7,0 ± 3,1 | 4,5 ± 1,4 | 39 ≪ |
| lca | 46,3 ± 11,1 | 46,3 ± 11,1 | 47,5 ± 11,1 | 52,5 ± 8,4 | 3,1 ± 1,0 | 3,1 ± 1,0 | 3,0 ± 1,1 | 57 ≪ |
| dnp | 83,0 ± 3,1 | 82,8 ± 5,0 | 78,1 ± 5,0 | 87,4 ± 5,6 | 5,9 ± 2,2 | 6,0 ± 3,6 | 3,0 ± 1,2 | 58 ≪ |
| son | 72,2 ± 5,0 | 74,5 ± 4,8 | 71,8 ± 3,2 | 70,6 ± 5,5 | 6,6 ± 4,4 | 10,1 ± 6,2 | 3,5 ± 1,7 | 61 ≪ |
| dna | 95,3 ± 0,4 | 94,8 ± 0,6 | 94,4 ± 0,5 | 95,3 ± 0,3 | 15,2 ± 2,4 | 11,7 ± 3,1 | 9,4 ± 1,6 | 61 ≪ |
| spl | 95,0 ± 0,5 | 94,9 ± 0,6 | 94,1 ± 0,6 | 95,3 ± 0,6 | 15,8 ± 3,1 | 15,3 ± 3,4 | 9,6 ± 2,1 | 62 ≪ |
| aud | 71,2 ± 2,1 | 69,2 ± 3,9 | 69,7 ± 4,3 | 63,5 ± 2,9 | 9,1 ± 1,7 | 8,7 ± 2,0 | 6,2 ± 1,5 | 70 ≪ |
| rat | 69,2 ± 4,1 | 69,2 ± 6,2 | 70,9 ± 4,3 | 52,2 ± 4,5 < | 9,2 ± 6,7 | 9,4 ± 5,6 | 4,2 ± 1,4 | 142 ≪ |
| arr | 65,3 ± 2,0 | 66,2 ± 2,0 | 63,5 ± 1,4 | 64,2 ± 1,6 | 13,9 ± 4,6 | 12,6 ± 4,4 | 7,5 ± 1,7 | 280 ≪ |

VNS-Variable Neighbourhood Search; VNS$_m$-VNS híbrida con memoria; SFFS-Sequential Forward Floating Selection; > y ≫ mejora significativa en un 95 y 99 %; < y ≪ empeora significativamente.

Tabla 5.5: Comparativa entre las metaheurísticas VNS y VNS$_m$, la búsqueda local SFFS y el clasificador base con IB1. Porcentaje de acierto y reducción en conjuntos reales.

| cdd | IB1 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | % aciertos | | | | # atributos | | | |
| | VNS | VNS$_m$ | SFFS | base | VNS | VNS$_m$ | SFFS | base |
| pdi | 69,0 ± 2,3 | 68,5 ± 3,1 | 65,4 ± 8,1 | 70,1 ± 1,9 | 3,9 ± 1,0 | 3,7 ± 0,9 | 2,5 ± 1,1 | 9 ≪ |
| gla | 69,5 ± 3,6 | 68,2 ± 2,9 | 68,5 ± 3,6 | 66,2 ± 3,6 | 5,3 ± 1,3 | 5,5 ± 1,1 | 4,6 ± 1,3 | 10 ≪ |
| bca | 62,7 ± 7,9 | 62,1 ± 8,1 | 62,0 ± 7,9 | 66,0 ± 3,1 | 2,8 ± 1,4 | 2,7 ± 1,2 | 2,5 ± 0,8 | 10 ≪ |
| bwi | 94,5 ± 1,2 | 94,7 ± 1,0 | 94,3 ± 1,6 | 95,6 ± 0,9 | 3,8 ± 1,0 | 4,3 ± 1,3 | 3,4 ± 1,0 | 10 ≪ |
| win | 93,4 ± 3,1 | 93,8 ± 2,9 | 91,8 ± 3,3 ≪ | 94,8 ± 2,1 | 4,8 ± 1,5 | 5,0 ± 1,6 | 3,9 ± 1,4 | 14 ≪ |
| hst | 74,0 ± 2,8 | 72,6 ± 5,0 | 68,8 ± 6,9 | 77,0 ± 2,3 | 5,5 ± 2,5 | 5,9 ± 2,6 | 2,5 ± 1,3 > | 14 ≪ |
| hhu | 71,2 ± 12,1 | 72,2 ± 12,5 | 66,8 ± 17,2 | 78,9 ± 2,6 | 4,0 ± 2,2 | 3,9 ± 2,0 | 3,1 ± 1,6 | 14 ≪ |
| hcl | 73,0 ± 7,8 | 71,7 ± 8,6 | 65,5 ± 10,0 | 78,0 ± 2,6 | 6,1 ± 1,7 | 6,2 ± 2,1 | 3,1 ± 2,0 | 14 ≪ |
| vow | 93,9 ± 1,5 | 94,2 ± 1,4 | 93,3 ± 2,5 | 93,5 ± 2,1 | 8,9 ± 0,7 | 8,8 ± 0,8 | 8,0 ± 1,2 > | 14 ≪ |
| cra | 73,9 ± 13,2 | 73,5 ± 13,0 | 75,6 ± 12,1 | 81,0 ± 1,9 | 3,8 ± 2,6 | 3,5 ± 2,0 | 2,7 ± 1,9 > | 16 ≪ |
| ptu | 34,1 ± 5,1 | 32,2 ± 5,5 | 25,2 ± 5,8 | 31,9 ± 4,4 | 9,4 ± 2,1 | 10,0 ± 1,3 | 4,2 ± 2,7 | 18 ≪ |
| vot | 94,5 ± 2,0 | 94,5 ± 2,0 | 95,3 ± 0,9 | 92,3 ± 1,0 | 3,4 ± 1,9 | 3,4 ± 1,9 | 2,5 ± 1,6 | 17 ≪ |
| lym | 80,4 ± 3,5 | 79,2 ± 3,6 | 72,2 ± 12,7 | 78,6 ± 4,8 | 7,8 ± 2,3 | 7,4 ± 2,3 | 4,5 ± 2,8 | 19 ≪ |
| veh | 68,6 ± 2,9 | 68,6 ± 2,9 | 66,9 ± 2,8 | 68,7 ± 2,4 | 7,7 ± 2,0 | 7,5 ± 1,7 | 6,1 ± 1,2 | 19 ≪ |
| hep | 79,1 ± 6,0 | 80,6 ± 5,6 | 80,8 ± 3,5 | 79,0 ± 3,6 | 4,8 ± 1,9 | 4,5 ± 2,4 | 2,3 ± 1,3 > | 20 ≪ |
| hco | 79,4 ± 5,2 | 79,9 ± 5,7 | 80,7 ± 5,6 | 78,4 ± 2,2 | 6,9 ± 2,7 | 6,5 ± 3,1 | 4,6 ± 2,0 | 23 ≪ |
| aut | 70,3 ± 6,6 | 69,4 ± 5,7 | 70,5 ± 6,4 | 67,5 ± 5,0 | 4,0 ± 0,9 | 4,6 ± 1,8 | 3,5 ± 1,0 | 26 ≪ |
| abp | 95,5 ± 0,8 | 95,8 ± 0,8 | 95,5 ± 0,8 | 96,1 ± 0,3 | 2,1 ± 1,7 | 3,3 ± 4,1 | 2,1 ± 1,7 | 30 ≪ |
| ion | 87,5 ± 3,1 | 87,1 ± 1,9 | 85,4 ± 4,0 | 85,2 ± 1,9 | 5,3 ± 2,1 | 5,4 ± 2,2 | 3,2 ± 1,2 | 35 ≪ |
| soy | 88,8 ± 3,1 | 88,6 ± 3,2 | 86,9 ± 6,5 | 88,6 ± 1,8 | 17,8 ± 3,5 | 18,2 ± 3,4 | 14,7 ± 2,2 | 36 ≪ |
| kvk | 93,9 ± 3,2 | 94,7 ± 3,3 | 94,7 ± 1,7 | 89,2 ± 0,6 ≪ | 8,6 ± 2,8 | 9,3 ± 2,5 | 6,6 ± 2,6 > | 37 ≪ |
| ann | 96,0 ± 1,6 | 96,5 ± 1,8 | 96,3 ± 2,4 | 94,2 ± 0,8 | 9,2 ± 1,1 | 8,9 ± 0,9 | 8,6 ± 1,0 | 39 ≪ |
| lca | 50,6 ± 12,7 | 47,5 ± 14,5 | 40,0 ± 17,7 | 46,3 ± 8,9 | 5,2 ± 1,9 | 4,9 ± 1,7 | 3,1 ± 1,1 | 57 ≪ |
| dnp | 70,4 ± 10,3 | 74,0 ± 9,2 | 71,5 ± 11,1 | 79,2 ± 6,0 | 9,1 ± 4,1 | 9,5 ± 4,7 | 2,5 ± 1,3 > | 58 ≪ |
| son | 81,2 ± 4,5 | 80,5 ± 2,8 | 74,7 ± 6,9 | 82,0 ± 2,6 | 14,7 ± 3,1 | 12,9 ± 4,6 | 5,6 ± 2,5 ≫ | 61 ≪ |
| dna | 85,4 ± 1,8 | 85,4 ± 1,8 | 85,0 ± 2,0 | 73,6 ± 0,9 ≪ | 6,3 ± 1,3 | 6,3 ± 1,3 | 6,1 ± 1,3 | 61 ≪ |
| spl | 84,7 ± 3,9 | 84,0 ± 4,0 | 82,0 ± 4,9 | 73,7 ± 0,8 < | 6,1 ± 1,1 | 5,5 ± 1,2 | 5,1 ± 1,1 | 62 ≪ |
| aud | 69,6 ± 4,7 | 68,5 ± 5,6 | 64,6 ± 8,5 | 70,6 ± 2,7 | 12,7 ± 3,1 | 11,8 ± 2,4 | 8,7 ± 2,5 | 70 ≪ |
| rat | 68,4 ± 6,2 | 68,4 ± 6,4 | 67,9 ± 6,2 | 66,4 ± 4,8 | 5,6 ± 3,6 | 4,4 ± 2,1 | 3,8 ± 2,1 | 142 ≪ |
| arr | 61,1 ± 1,9 | 60,2 ± 2,3 | 47,3 ± 15,6 | 51,8 ± 4,3 < | 19,9 ± 4,8 | 22,2 ± 3,9 | 10,6 ± 3,6 | 280 ≪ |

VNS-Variable Neighbourhood Search; VNS$_m$-VNS híbrida con memoria; SFFS-Sequential Forward Floating Selection; > y ≫ mejora significativa en un 95 y 99 %; < y ≪ empeora significativamente.

Tabla 5.6: Comparativa entre las metaheurísticas VNS y VNS$_m$, la búsqueda local SFFS y el clasificador base con J48. Porcentaje de acierto y reducción en conjuntos reales.

| cdd | J48 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | % aciertos | | | | # atributos | | | |
| | VNS | VNS$_m$ | SFFS | base | VNS | VNS$_m$ | SFFS | base |
| pdi | 73,2 ± 2,1 | 73,2 ± 2,1 | 73,3 ± 2,1 | 72,5 ± 1,7 | 1,8 ± 1,0 | 1,8 ± 1,0 | 1,5 ± 0,7 | 9 ≪ |
| gla | 62,9 ± 5,3 | 63,9 ± 5,8 | 64,5 ± 7,8 | 63,2 ± 6,1 | 4,1 ± 1,1 | 4,2 ± 0,9 | 3,3 ± 0,7 | 10 ≪ |
| bca | 71,3 ± 2,7 | 71,3 ± 2,7 | 70,6 ± 2,0 | 69,6 ± 1,7 | 2,2 ± 1,2 | 2,2 ± 1,2 | 1,9 ± 1,2 > | 10 ≪ |
| bwi | 94,2 ± 1,0 | 94,2 ± 1,1 | 94,1 ± 1,3 | 94,4 ± 1,0 | 2,9 ± 1,1 | 3,1 ± 1,4 | 2,4 ± 0,5 | 10 ≪ |
| win | 89,6 ± 3,9 | 89,6 ± 3,9 | 89,6 ± 3,9 | 91,9 ± 2,7 | 2,4 ± 0,5 | 2,4 ± 0,5 | 2,4 ± 0,5 | 14 ≪ |
| hst | 75,6 ± 4,4 | 73,8 ± 6,2 | 71,3 ± 4,4 | 78,7 ± 3,2 | 2,3 ± 1,5 | 2,4 ± 1,6 | 1,4 ± 0,5 | 14 ≪ |
| hhu | 78,8 ± 2,9 | 78,8 ± 2,9 | 78,8 ± 2,9 | 78,6 ± 3,0 | 1,3 ± 0,5 | 1,3 ± 0,5 | 1,3 ± 0,5 | 14 ≪ |
| hcl | 74,1 ± 3,0 | 74,6 ± 4,1 | 73,9 ± 2,0 | 76,9 ± 3,2 > | 2,6 ± 1,3 | 2,7 ± 1,6 | 1,9 ± 1,1 | 14 ≪ |
| vow | 69,3 ± 3,0 | 69,1 ± 2,9 | 68,2 ± 3,0 | 69,9 ± 2,6 | 6,9 ± 0,9 | 6,9 ± 0,9 | 6,0 ± 1,5 | 14 ≪ |
| cra | 85,5 ± 1,5 | 85,5 ± 1,5 | 85,2 ± 1,0 | 84,3 ± 1,5 | 1,8 ± 1,6 | 1,8 ± 1,6 | 1,3 ± 0,7 | 16 ≪ |
| ptu | 37,4 ± 3,2 | 36,9 ± 3,2 | 34,4 ± 2,9 | 38,5 ± 4,5 | 7,3 ± 1,4 | 7,3 ± 1,6 | 4,2 ± 1,6 | 18 ≪ |
| vot | 95,6 ± 0,7 | 95,6 ± 0,7 | 95,6 ± 0,7 | 95,3 ± 1,1 | 1 | 1 | 1 | 17 ≪ |
| lym | 74,9 ± 4,5 | 75,8 ± 3,3 | 75,8 ± 4,3 | 75,9 ± 3,4 | 2,2 ± 0,8 | 2,3 ± 0,8 | 2,1 ± 0,9 | 19 ≪ |
| veh | 68,0 ± 3,2 | 67,4 ± 2,6 | 66,1 ± 2,3 | 69,7 ± 2,2 | 8,1 ± 2,5 | 6,9 ± 2,8 | 4,8 ± 1,0 | 19 ≪ |
| hep | 79,7 ± 3,1 | 79,7 ± 3,1 | 78,8 ± 6,0 | 80,1 ± 2,9 | 1,8 ± 0,8 | 1,8 ± 0,8 | 1,4 ± 0,5 | 20 ≪ |
| hco | 84,9 ± 2,3 | 84,5 ± 2,8 | 84,5 ± 2,8 | 85,4 ± 2,3 | 2,8 ± 0,6 | 2,8 ± 0,6 | 2,8 ± 0,6 | 23 ≪ |
| aut | 62,8 ± 5,2 | 62,0 ± 4,7 | 62,8 ± 6,3 | 65,6 ± 6,1 | 3,3 ± 1,9 | 3,5 ± 1,8 | 2,5 ± 1,1 | 26 ≪ |
| abp | 97,1 ± 0,3 | 97,1 ± 0,3 | 97,1 ± 0,3 | 97,1 ± 0,4 | 2,5 ± 1,0 | 2,5 ± 1,0 | 2,5 ± 1,0 | 30 ≪ |
| ion | 88,1 ± 2,2 | 88,2 ± 1,6 | 87,6 ± 2,0 | 87,5 ± 2,4 | 3,8 ± 1,3 | 3,8 ± 1,3 | 3,0 ± 0,8 | 35 ≪ |
| soy | 86,6 ± 2,3 | 87,3 ± 1,9 | 85,9 ± 2,9 | 88,1 ± 1,5 | 16,3 ± 2,4 | 15,5 ± 3,0 | 12,5 ± 1,1 ≫ | 36 ≪ |
| kvk | 95,0 ± 1,7 | 97,1 ± 1,3 | 94,1 ± 0,5 | 99,1 ± 0,2 | 5,5 ± 2,4 | 8,5 ± 1,7 | 4,3 ± 0,5 | 37 ≪ |
| ann | 85,1 ± 3,4 | 86,3 ± 4,0 ≫ | 85,4 ± 3,3 | 89,5 ± 1,8 | 4,4 ± 1,7 | 4,3 ± 1,6 > | 4,0 ± 1,3 ≫ | 39 ≪ |
| lca | 37,5 ± 8,8 | 37,5 ± 8,8 | 37,5 ± 8,8 | 41,9 ± 10,2 | 1,3 ± 0,7 | 1,3 ± 0,7 | 1,3 ± 0,7 | 57 ≪ |
| dnp | 74,9 ± 2,7 | 74,9 ± 2,7 | 74,9 ± 2,7 | 72,6 ± 4,9 | 1,3 ± 0,7 | 1,3 ± 0,7 | 1,3 ± 0,7 | 58 ≪ |
| son | 67,6 ± 2,4 | 68,0 ± 2,8 | 67,9 ± 2,9 | 69,3 ± 4,6 | 5,5 ± 1,8 | 5,5 ± 1,8 | 4,7 ± 1,9 | 61 ≪ |
| dna | 92,8 ± 1,1 | 92,9 ± 0,8 | 92,8 ± 0,9 | 92,8 ± 0,7 | 8,6 ± 1,0 | 8,6 ± 1,2 | 8,1 ± 1,2 | 61 ≪ |
| spl | 92,4 ± 0,5 | 92,3 ± 0,6 | 92,4 ± 0,6 | 92,5 ± 0,5 | 8,0 ± 0,8 | 7,9 ± 0,9 | 7,8 ± 0,8 | 62 ≪ |
| aud | 71,0 ± 3,6 | 70,3 ± 3,2 | 69,8 ± 3,4 | 74,4 ± 2,8 | 6,4 ± 1,4 | 6,0 ± 1,4 | 5,3 ± 1,3 | 70 ≪ |
| rat | 70,4 ± 4,5 | 71,3 ± 5,1 | 71,6 ± 5,1 | 70,3 ± 5,7 | 4,6 ± 1,6 | 5,0 ± 1,8 | 3,5 ± 1,4 | 142 ≪ |
| arr | 66,1 ± 3,8 | 66,5 ± 3,3 | 64,5 ± 4,0 | 65,1 ± 2,3 | 11,3 ± 2,9 | 12,8 ± 2,9 | 6,2 ± 1,4 | 280 ≪ |

VNS-Variable Neighbourhood Search; VNS$_m$-VNS híbrida con memoria; SFFS-Sequential Forward Floating Selection; > y ≫ mejora significativa en un 95 y 99 %; < y ≪ empeora significativamente.

### Problemas artificiales

Los porcentajes de aciertos y tamaños de los subconjuntos de atributos encontrados por VNS y SFFS usando Naïve Bayes sobre problemas artificiales pueden consultarse en la Tabla 5.1 (las columnas 5 y 9 muestran los resultados obtenidos con VNS, y las columnas 7 y 11 los obtenidos con SFFS). VNS mejora a SFFS en porcentaje de aciertos en todos los problemas excepto en 5 de ellos. Aplicando el test F se obtiene que sólo existen 3 diferencias significativas a favor de VNS. Las diferencias a favor de SFFS no son significativas. El análisis del tamaño de los subconjuntos de atributos encontrados por VNS y SFFS revela que todas las diferencias son favorables a SFFS, siendo 9 de ellas estadísticamente significativas.

El test de Nemenyi indica que en el acierto existen diferencias significativas favorable a VNS al nivel $\alpha = 0{,}05$. Analizando los problemas en función de la regla de clasificación, se obtiene que, en ambos casos, las diferencias son significativas al nivel $\alpha = 0{,}05$. El estudio en función del número de atributos relevantes revela que las únicas diferencias significativas al nivel $\alpha = 0{,}05$ se presentan en problemas con 10 atributos relevantes. En todos estos casos las diferencias son favorables a VNS.

La Tabla 5.2 muestra los resultados obtenidos con IB1. VNS obtiene mejores porcentajes de aciertos que SFFS en todos los problemas excepto en 4 de ellos. Sólo en un problema esta diferencia es estadísticamente significativa. Sin embargo, SFFS encuentra subconjuntos de atributos de menor tamaño que el VNS, siendo significativas estas diferencias.

Aplicando el test de Nemenyi se obtienen resultados iguales a los obtenidos con Naïve Bayes.

En la Tabla 5.3 se detallan los porcentajes de aciertos y tamaños de los subconjuntos de atributos obtenidos con J48. VNS obtiene, en la mayoría de los casos, mejores porcentajes de aciertos que SFFS, aunque con subconjuntos con mayor número de atributos. De las diferencias obtenidas sólo hay una estadísticamente significativa a favor de VNS cuando se analizan los porcentajes de aciertos, y 4 a favor de SFFS cuando se considera el número de atributos.

De la aplicación del test de Nemenyi considerando todos los problemas se concluye que existe diferencia estadísticamente significativa al nivel $\alpha = 0{,}05$ a favor de VNS cuando se toma como medida de comportamiento el porcentaje de aciertos. Un estudio detallado, analizando los resultados en función del tipo de regla empleada para generar los problemas, revela que las diferencias entre VNS y SFFS son significativas al nivel $\alpha = 0{,}05$. Si el análisis se realiza en función del número de atributos relevantes se concluye que, para problemas con 10 atributos relevantes, las diferencias son estadísticamente significativas a favor de VNS al nivel $\alpha = 0{,}05$.

En todos los casos, SFFS selecciona menos atributos que VNS. Esta diferencia es estadísticamente significativa al nivel $\alpha = 0{,}05$ cuando se consideran todos los problemas y cuando se analizan en función de la regla de clasificación y del número de atributos relevantes.

### Problemas reales

A continuación pasamos a analizar el rendimiento de ambas estrategias en los problemas reales. La Tabla 5.4 muestra los porcentajes de aciertos y números de atributos obtenidos al usar Naïve Bayes (columnas 2 y 6 para VNS y columnas 4 y 8 para SFFS). En general, VNS mejora a SFFS en porcentajes de aciertos y siempre SFFS mejora a VNS en número de atributos. Sin embargo, las diferencias obtenidas sólo son estadísticamente significativas a favor de cada estrategia en un problema.

El test de Nemenyi aplicado sobre todos los problemas indica que sí existen diferencias estadísticamente significativas en el porcentaje de aciertos al nivel $\alpha = 0{,}05$. Por otra parte, la diferencia entre el número de atributos obtenidos por VNS y SFFS son estadísticamente significativas al nivel $\alpha = 0{,}05$ a favor de SFFS. Realizando un análisis detallado por tamaño de problema se sigue que, para problemas pequeños, las diferencias son estadísticamente significativas al nivel $\alpha = 0{,}05$ a favor de VNS. En los conjuntos pequeños y grandes, el tamaño de los subconjuntos seleccionados por SFFS es significativamente menor que el de los seleccionados por VNS al nivel $\alpha = 0{,}05$ a favor de SFFS.

El análisis de los resultados obtenidos con el clasificador IB1 aparecen en la Tabla 5.5, siendo sus conclusiones similares a las obtenidas con Naïve Bayes. Sin embargo, ahora existen más diferencias estadísticamente significativas a favor de SFFS en lo relativo a la reducción de la dimensionalidad.

El análisis sobre todos los problemas indica que ambas estrategias presentan comportamientos estadísticamente significativos al nivel $\alpha = 0{,}05$ tanto en porcentajes de aciertos como en tamaño de los subconjuntos de atributos. La primera es favorable a VNS y la segunda a SFFS. El análisis en función de los tamaños de los problemas indica que las diferencias en porcentajes de aciertos son estadísticamente significativas al nivel $\alpha = 0{,}1$ en problemas pequeños y al nivel $\alpha = 0{,}05$ en problemas grandes.

En lo relativo al tamaño de los subconjuntos encontrados, se tiene que las diferencias estadísticamente significativas se presentan en los problemas pequeños y grandes. En ambos casos, SFFS encuentra subconjuntos de menor tamaño.

Por último, la Tabla 5.6 muestra los resultados alcanzados con J48. Con este clasificador las diferencias son menores, pues sólo en el análisis del tamaño de los subconjuntos de atributos encontramos 3 diferencias que son estadísticamente significativas a favor del SFFS. El porcentaje de acierto obtenido en varios problemas son similares.

El test de Nemenyi aplicado sobre todos los resultados muestra que SFFS selecciona subconjuntos de menor tamaño siendo esta diferencia estadísticamente significativa al nivel $\alpha = 0{,}05$. Además, un análisis detallado prueba que esta diferencia es también estadísticamente significativa en problemas pequeños al nivel $\alpha = 0{,}05$.

### 5.4.5 Comparativa del rendimiento entre VNS y VNS$_m$

En este apartado analizaremos la aportación que hace al VNS su hibridación con la Búsqueda Tabú.

### Problemas artificiales

Los porcentajes de aciertos y tamaño del subconjunto de atributos encontrado con Naïve Bayes pueden consultarse en la Tabla 5.1 (las columnas 5 y 9 muestran los resultados obtenidos con VNS, y las columnas 6 y 10 los obtenidos con $\text{VNS}_m$). Hay tres diferencias estadísticamente significativas: dos a favor de $\text{VNS}_m$ y una a favor de VNS, tanto en porcentaje de aciertos como en número de atributos.

Aplicando el test de Nemenyi sobre los valores promedios de todos los problemas, se sigue que la diferencia en los porcentajes de aciertos no es estadísticamente significativa. Sin embargo, analizando estos resultados en función del número de atributos relevantes encontramos que en ambos casos las diferencias son estadísticamente significativas al nivel $\alpha = 0{,}05$. La diferencia es favorable a $\text{VNS}_m$ para los problemas con 5 atributos relevantes y favorable a VNS en los problemas con 10 atributos relevantes. Al analizar los tamaños de los subconjuntos de atributos sobre todos los problemas, se sigue que la diferencia es estadísticamente significativa al nivel $\alpha = 0{,}05$. Analizando los resultados en función de la regla de clasificación y del número de atributos relevantes encontramos que, en todos los casos, existen diferencias significativas al nivel $\alpha = 0{,}05$ a favor de $\text{VNS}_m$. Es decir, la hibridación encuentra subconjuntos de menor tamaño.

Los resultados con el IB1 se hallan en la Tabla 5.2 (las columnas 5 y 9 muestran los resultados obtenidos con VNS, y las columnas 6 y 10 los obtenidos con $\text{VNS}_m$). Ninguna de las diferencias presentes en los conjuntos de datos son estadísticamente significativas.

El test de Nemenyi no detecta diferencias significativas en el rendimiento de las dos estrategias con el clasificador IB1.

Por último, en la Tabla 5.3 encontramos los resultados correspondientes al clasificador J48 (las columnas 5 y 9 muestran los resultados obtenidos con VNS, y las columnas 6 y 10 los obtenidos con $\text{VNS}_m$). En este caso, no se encuentran diferencias significativas en el porcentaje de aciertos.

La única diferencia se encuentra al aplicar el test de Nemenyi sobre el número de atributos sobre todos los problemas al nivel $\alpha = 0{,}1$. El análisis

detallado en función de las reglas de clasificación y número de atributos relevantes no detecta diferencias significativas.

Finalmente, analizamos la convergencia de cada una de las dos estrategias. En la Tabla 5.7 se muestra el número de iteraciones desarrollado por cada estrategia con cada clasificador. $\text{VNS}_m$ converge antes que VNS en todos los problemas, excepto en tres cuando se usa el clasificador J48. El test F muestra que existen, respectivamente, 5, 1 y 3 diferencias significativas a favor de $\text{VNS}_m$ cuando se usan Naïve Bayes, IB1 y J48.

El test de Nemenyi aplicado a los resultados obtenidos en todos los problemas, detecta que las diferencias son estadísticamente significativas a favor de $\text{VNS}_m$ al nivel $\alpha = 0{,}05$ en los tres clasificadores.

### Problemas reales

El porcentaje de acierto y la reducción obtenida con VNS y $\text{VNS}_m$ están en la Tabla 5.4 (las columnas 2 y 6 muestran los resultados obtenidos con VNS, y las columnas 3 y 7 los obtenidos con $\text{VNS}_m$). Como puede observarse, los resultados alcanzados son bastante similares. Así, en algunos problemas los resultados favorecen a VNS y en otros a $\text{VNS}_m$. Sólo existen dos diferencias estadísticamente significativas: una favorable a $\text{VNS}_m$ en el porcentaje de aciertos, y otra favorable a VNS en el número de atributos.

El test de Nemenyi aplicado sobre los resultados de todos los problemas no detecta diferencias estadísticamente significativas.

Las conclusiones del análisis se extienden a los resultados obtenidos con el clasificador IB1 (Tabla 5.5) y a los obtenidos con J48 (Tabla 5.6). Sólo cabe destacar que, cuando se aplica IB1, no hay diferencias significativas, mientras que, al aplicar J48, se detectan dos diferencias significativas a favor de $\text{VNS}_m$ (una en el porcentaje de aciertos y otra en el número de atributos).

Analizando el número de iteraciones alcanzadas por cada estrategia (Tabla 5.8), observamos que $\text{VNS}_m$ converge en promedio antes que VNS en la mayoría de los casos, independientemente del clasificador usado. Sin embargo, las diferencias estadísticamente significativas son pocas: 3 con Naïve Bayes, una con IB1 y una con J48.

Tabla 5.7: Comparativa entre las metaheurísticas VNS y VNS$_m$. Número de iteraciones alcanzadas en conjuntos artificiales.

| conjuntos de datos | | | | # iteraciones | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Naïve Bayes | | IB1 | | J48 | |
| l | r | $\sigma$ | d | VNS | VNS$_m$ | VNS | VNS$_m$ | VNS | VNS$_m$ |
| 1 | 5 | 0 | 25 | $6{,}6 \pm 4{,}3$ | $4{,}0 \pm 2{,}1$ | $3{,}6 \pm 1{,}4$ | $3{,}0 \pm 1{,}1$ | $4{,}9 \pm 2{,}0$ | $3{,}2 \pm 1{,}1$ |
| | | | 50 | $6{,}7 \pm 5{,}0$ | $4{,}6 \pm 2{,}7$ $\gg$ | $4{,}2 \pm 1{,}0$ | $3{,}9 \pm 1{,}9$ | $6{,}7 \pm 2{,}5$ | $5{,}4 \pm 2{,}5$ |
| | | | 100 | $5{,}6 \pm 2{,}2$ | $3{,}8 \pm 0{,}8$ $>$ | $5{,}8 \pm 3{,}6$ | $4{,}9 \pm 2{,}9$ | $5{,}2 \pm 2{,}0$ | $3{,}9 \pm 1{,}7$ |
| | | | 200 | $6{,}2 \pm 4{,}0$ | $4{,}6 \pm 2{,}8$ | $19{,}0 \pm 9{,}8$ | $9{,}6 \pm 7{,}0$ | $8{,}7 \pm 6{,}2$ | $4{,}4 \pm 1{,}8$ |
| | | 0,1 | 25 | $4{,}6 \pm 2{,}2$ | $3{,}3 \pm 1{,}2$ | $3{,}9 \pm 2{,}5$ | $2{,}9 \pm 0{,}7$ | $4{,}1 \pm 2{,}7$ | $3{,}2 \pm 2{,}9$ |
| | | | 50 | $4{,}8 \pm 2{,}5$ | $3{,}2 \pm 0{,}8$ | $5{,}2 \pm 3{,}5$ | $3{,}7 \pm 1{,}3$ | $5{,}3 \pm 1{,}9$ | $4{,}9 \pm 3{,}7$ $>$ |
| | | | 100 | $5{,}7 \pm 2{,}1$ | $5{,}2 \pm 1{,}8$ $>$ | $5{,}1 \pm 1{,}7$ | $4{,}4 \pm 1{,}9$ | $5{,}8 \pm 3{,}3$ | $5{,}8 \pm 2{,}1$ |
| | | | 200 | $7{,}2 \pm 4{,}0$ | $4{,}6 \pm 3{,}0$ | $7{,}5 \pm 4{,}7$ | $4{,}6 \pm 1{,}3$ | $9{,}9 \pm 4{,}7$ | $6{,}8 \pm 4{,}7$ |
| | 10 | 0 | 25 | $10{,}2 \pm 4{,}2$ | $7{,}8 \pm 2{,}5$ | $7{,}2 \pm 3{,}5$ | $5{,}9 \pm 3{,}2$ | $5{,}1 \pm 3{,}0$ | $6{,}4 \pm 4{,}3$ |
| | | | 50 | $13{,}4 \pm 6{,}7$ | $7{,}9 \pm 3{,}6$ $>$ | $10{,}4 \pm 6{,}9$ | $7{,}5 \pm 5{,}0$ | $7{,}3 \pm 4{,}6$ | $5{,}1 \pm 3{,}4$ |
| | | | 100 | $19{,}5 \pm 8{,}5$ | $13{,}3 \pm 11{,}8$ | $14{,}7 \pm 8{,}0$ | $11{,}6 \pm 6{,}8$ | $11{,}9 \pm 7{,}9$ | $9{,}2 \pm 7{,}6$ |
| | | | 200 | $16{,}5 \pm 10{,}5$ | $12{,}4 \pm 7{,}6$ | $8{,}8 \pm 5{,}1$ | $6{,}4 \pm 5{,}7$ | $12{,}7 \pm 10{,}2$ | $9{,}2 \pm 4{,}5$ $>$ |
| | | 0,1 | 25 | $11{,}6 \pm 3{,}9$ | $7{,}7 \pm 2{,}7$ | $6{,}7 \pm 4{,}5$ | $5{,}6 \pm 3{,}0$ | $6{,}8 \pm 2{,}9$ | $5{,}4 \pm 3{,}5$ |
| | | | 50 | $15{,}9 \pm 7{,}2$ | $10{,}8 \pm 3{,}7$ | $9{,}7 \pm 5{,}9$ | $6{,}5 \pm 1{,}8$ | $8{,}5 \pm 3{,}6$ | $6{,}1 \pm 3{,}2$ |
| | | | 100 | $15{,}6 \pm 5{,}8$ | $11{,}4 \pm 4{,}8$ | $17{,}6 \pm 8{,}0$ | $11{,}5 \pm 4{,}0$ | $11{,}6 \pm 5{,}5$ | $9{,}0 \pm 5{,}2$ |
| | | | 200 | $21{,}7 \pm 10{,}8$ | $11{,}8 \pm 6{,}8$ | $17{,}7 \pm 9{,}8$ | $11{,}4 \pm 6{,}3$ | $10{,}4 \pm 5{,}3$ | $8{,}9 \pm 5{,}2$ |
| 2 | 5 | 0 | 25 | $6{,}3 \pm 3{,}7$ | $4{,}7 \pm 2{,}7$ | $4{,}9 \pm 2{,}6$ | $2{,}7 \pm 0{,}7$ | $4{,}4 \pm 2{,}8$ | $3{,}8 \pm 3{,}0$ |
| | | | 50 | $6{,}7 \pm 2{,}5$ | $3{,}5 \pm 1{,}2$ | $4{,}5 \pm 2{,}1$ | $3{,}5 \pm 2{,}0$ | $4{,}4 \pm 1{,}5$ | $4{,}3 \pm 2{,}4$ |
| | | | 100 | $6{,}8 \pm 3{,}2$ | $5{,}0 \pm 3{,}4$ | $6{,}5 \pm 2{,}1$ | $4{,}3 \pm 2{,}1$ | $7{,}0 \pm 5{,}3$ | $5{,}6 \pm 5{,}6$ |
| | | | 200 | $7{,}9 \pm 2{,}8$ | $7{,}5 \pm 6{,}2$ | $4{,}7 \pm 2{,}6$ | $4{,}0 \pm 1{,}8$ | $7{,}9 \pm 3{,}4$ | $6{,}6 \pm 6{,}0$ |
| | | 0,1 | 25 | $7{,}5 \pm 5{,}3$ | $3{,}8 \pm 1{,}9$ $>$ | $4{,}2 \pm 1{,}7$ | $3{,}1 \pm 1{,}4$ | $5{,}2 \pm 3{,}0$ | $3{,}1 \pm 1{,}9$ |
| | | | 50 | $6{,}3 \pm 3{,}1$ | $4{,}1 \pm 1{,}7$ | $5{,}0 \pm 2{,}7$ | $3{,}1 \pm 1{,}7$ $>$ | $4{,}9 \pm 3{,}0$ | $3{,}1 \pm 1{,}3$ |
| | | | 100 | $8{,}2 \pm 4{,}3$ | $3{,}8 \pm 1{,}0$ | $5{,}6 \pm 2{,}3$ | $3{,}0 \pm 1{,}1$ | $4{,}8 \pm 2{,}3$ | $5{,}4 \pm 3{,}2$ |
| | | | 200 | $7{,}1 \pm 3{,}3$ | $6{,}3 \pm 3{,}8$ | $4{,}4 \pm 2{,}0$ | $3{,}1 \pm 1{,}2$ | $6{,}0 \pm 3{,}4$ | $6{,}0 \pm 5{,}0$ |
| | 10 | 0 | 25 | $9{,}3 \pm 3{,}0$ | $7{,}1 \pm 2{,}4$ | $9{,}9 \pm 6{,}2$ | $7{,}8 \pm 4{,}0$ | $5{,}8 \pm 3{,}4$ | $5{,}5 \pm 3{,}9$ |
| | | | 50 | $14{,}1 \pm 8{,}0$ | $8{,}5 \pm 3{,}4$ | $8{,}4 \pm 5{,}5$ | $7{,}2 \pm 4{,}7$ | $7{,}2 \pm 3{,}6$ | $6{,}2 \pm 3{,}4$ |
| | | | 100 | $15{,}7 \pm 5{,}4$ | $9{,}3 \pm 5{,}0$ | $10{,}8 \pm 5{,}4$ | $7{,}4 \pm 3{,}6$ | $8{,}3 \pm 2{,}8$ | $8{,}9 \pm 6{,}3$ |
| | | | 200 | $17{,}2 \pm 13{,}2$ | $9{,}8 \pm 6{,}0$ | $13{,}7 \pm 6{,}4$ | $9{,}3 \pm 5{,}2$ | $11{,}1 \pm 6{,}3$ | $7{,}9 \pm 3{,}2$ |
| | | 0,1 | 25 | $11{,}0 \pm 5{,}0$ | $6{,}9 \pm 2{,}3$ | $8{,}1 \pm 4{,}8$ | $6{,}9 \pm 3{,}8$ | $6{,}9 \pm 3{,}7$ | $5{,}7 \pm 3{,}1$ |
| | | | 50 | $13{,}0 \pm 6{,}7$ | $7{,}4 \pm 4{,}4$ | $8{,}6 \pm 4{,}8$ | $4{,}7 \pm 2{,}3$ | $7{,}9 \pm 3{,}3$ | $5{,}9 \pm 4{,}4$ |
| | | | 100 | $16{,}0 \pm 6{,}3$ | $11{,}5 \pm 6{,}5$ | $17{,}4 \pm 10{,}6$ | $12{,}8 \pm 8{,}7$ | $10{,}9 \pm 4{,}5$ | $6{,}7 \pm 6{,}0$ |
| | | | 200 | $17{,}5 \pm 9{,}6$ | $11{,}1 \pm 6{,}7$ | $12{,}7 \pm 7{,}6$ | $8{,}6 \pm 4{,}4$ | $14{,}5 \pm 5{,}8$ | $6{,}9 \pm 3{,}3$ $\gg$ |

$l$-regla de clasificación; $r$-número de atributos relevantes; $\sigma$-ruido Gaussiano;VNS-Variable Neighbourhood Search; VNS$_m$-VNS híbrida con memoria; $>$ y $\gg$ mejora significativa en un 95 y 99 %; $<$ y $\ll$ empeora significativamente.

Tabla 5.8: Comparativa entre las metaheurísticas VNS y VNS$_m$. Número de iteraciones alcanzadas en conjuntos reales.

| | # iteraciones | | | | | |
|---|---|---|---|---|---|---|
| | Naïve Bayes | | IB1 | | J48 | |
| cdd | VNS | VNS$_m$ | VNS | VNS$_m$ | VNS | VNS$_m$ |
| pdi | 2,4 ± 0,8 | 1,9 ± 0,3 | 2,6 ± 1,1 | 2,3 ± 0,7 | 1,3 ± 0,7 | 1,2 ± 0,4 |
| gla | 2,4 ± 1,3 | 2,1 ± 0,7 > | 2,2 ± 0,8 | 2,4 ± 0,8 | 3,4 ± 1,3 | 2,5 ± 0,5 |
| bca | 2,3 ± 0,9 | 2,1 ± 0,7 | 1,7 ± 0,9 | 1,6 ± 0,7 | 1,7 ± 0,9 | 1,5 ± 0,7 ≫ |
| bwi | 2,6 ± 0,8 | 2,4 ± 0,7 | 2,9 ± 1,4 | 2,6 ± 1,1 | 2,3 ± 1,6 | 2,1 ± 1,2 |
| win | 4,4 ± 2,0 | 3,6 ± 1,6 | 3,7 ± 1,6 | 3,8 ± 2,2 | 1,4 ± 0,5 | 1,4 ± 0,5 |
| hst | 4,7 ± 2,9 | 4,2 ± 1,4 | 4,9 ± 3,0 | 4,0 ± 2,3 | 2,4 ± 1,7 | 1,9 ± 1,1 |
| hhu | 3,2 ± 1,1 | 3,0 ± 1,6 | 3,4 ± 2,3 | 2,8 ± 1,8 | 1,0 ± 0,0 | 1,0 ± 0,0 |
| hcl | 4,7 ± 2,6 | 3,6 ± 1,4 | 4,6 ± 1,9 | 3,7 ± 1,4 | 2,3 ± 1,6 | 2,3 ± 1,9 |
| vow | 3,5 ± 1,0 | 2,9 ± 1,0 | 2,6 ± 0,7 | 2,7 ± 1,3 | 3,7 ± 1,1 | 4,0 ± 1,8 |
| cra | 3,7 ± 1,6 | 2,9 ± 1,4 | 2,7 ± 1,5 | 2,1 ± 1,0 | 1,5 ± 1,1 | 1,4 ± 1,0 |
| ptu | 5,0 ± 1,2 | 4,3 ± 1,3 | 6,3 ± 1,6 | 6,4 ± 2,1 | 7,9 ± 3,2 | 6,0 ± 2,4 |
| vot | 1,7 ± 0,9 | 1,6 ± 0,7 | 2,5 ± 1,5 | 2,0 ± 1,1 | 1,0 ± 0,0 | 1,0 ± 0,0 |
| lym | 4,1 ± 2,8 | 4,1 ± 2,9 | 6,3 ± 2,5 | 4,5 ± 1,6 | 1,5 ± 0,7 | 1,6 ± 0,8 |
| veh | 5,5 ± 2,2 | 4,6 ± 1,8 > | 4,6 ± 1,8 | 3,7 ± 1,6 | 7,9 ± 4,0 | 3,9 ± 1,9 |
| hep | 3,3 ± 2,9 | 2,3 ± 1,5 | 4,2 ± 1,9 | 3,5 ± 1,9 | 1,5 ± 0,8 | 1,4 ± 0,7 |
| hco | 4,2 ± 2,0 | 4,0 ± 1,6 | 5,4 ± 3,0 | 3,7 ± 2,3 | 1,9 ± 0,9 | 1,7 ± 0,5 |
| aut | 4,9 ± 2,9 | 3,4 ± 1,6 | 2,6 ± 0,8 | 3,8 ± 4,0 | 2,5 ± 2,0 | 2,4 ± 1,7 |
| abp | 1,5 ± 0,5 | 1,5 ± 0,5 | 1,3 ± 0,7 | 2,8 ± 4,0 | 1,4 ± 0,5 | 1,4 ± 0,5 |
| ion | 7,1 ± 3,8 | 5,2 ± 1,9 | 4,1 ± 2,1 | 3,5 ± 1,6 | 3,4 ± 1,6 | 2,6 ± 1,2 |
| soy | 11,3 ± 5,1 | 8,7 ± 2,6 | 11,4 ± 2,7 | 8,8 ± 2,7 > | 16,9 ± 8,0 | 10,1 ± 4,7 |
| kvk | 4,2 ± 2,7 | 2,0 ± 0,0 | 5,7 ± 3,1 | 4,9 ± 1,7 | 3,5 ± 2,5 | 5,8 ± 1,9 |
| ann | 6,4 ± 4,1 | 5,5 ± 4,2 | 6,5 ± 2,2 | 4,4 ± 0,8 | 2,7 ± 1,3 | 3,2 ± 2,4 |
| lca | 2,0 ± 0,8 | 1,8 ± 0,6 | 4,8 ± 1,8 | 3,3 ± 1,3 | 1,1 ± 0,3 | 1,1 ± 0,3 |
| dnp | 6,1 ± 3,2 | 4,2 ± 3,6 | 8,1 ± 5,1 | 7,2 ± 4,7 | 1,1 ± 0,3 | 1,1 ± 0,3 |
| son | 7,0 ± 7,7 | 7,9 ± 6,2 | 12,9 ± 3,6 | 8,2 ± 4,3 | 4,4 ± 2,5 | 3,2 ± 2,1 |
| dna | 12,8 ± 6,1 | 6,4 ± 3,9 | 3,6 ± 1,0 | 2,9 ± 0,6 | 5,5 ± 1,4 | 4,9 ± 2,8 |
| spl | 16,7 ± 5,9 | 10,1 ± 5,1 | 4,5 ± 2,1 | 3,3 ± 2,1 | 4,5 ± 1,0 | 3,4 ± 0,7 |
| aud | 6,4 ± 2,1 | 5,3 ± 2,2 ≫ | 12,5 ± 6,6 | 8,4 ± 4,4 | 4,1 ± 1,7 | 3,2 ± 1,1 |
| rat | 8,0 ± 8,5 | 6,0 ± 4,7 | 3,9 ± 3,4 | 2,2 ± 0,9 | 4,0 ± 2,1 | 3,6 ± 1,8 |
| arr | 11,3 ± 7,0 | 7,7 ± 4,2 | 13,8 ± 4,5 | 13,1 ± 3,1 | 13,4 ± 6,4 | 11,6 ± 5,9 |

VNS-Variable Neighbourhood Search; VNS$_m$-VNS híbrida con memoria; > y ≫ mejora significativa en un 95 y 99 %; < y ≪ empeora significativamente.

Aplicando el test de Nemenyi sobre todos los resultados obtenidos con Naïve Bayes, se obtiene que las diferencias son estadísticamente significativas al nivel $\alpha = 0{,}05$ a favor de $\text{VNS}_m$. El análisis sobre el tamaño del problema revela que las diferencias son también estadísticamente favorables a $\text{VNS}_m$.

Los resultados obtenidos con IB1 muestran, según el test de Nemenyi, que sí existen diferencias significativas al nivel $\alpha = 0{,}05$ favorables a $\text{VNS}_m$. Analizando en función del tamaño de los problemas, se sigue que en los conjuntos pequeños esta diferencia es significativa al nivel $\alpha = 0{,}1$, mientras que en los grandes lo es al nivel $\alpha = 0{,}05$. Con el clasificador J48 se obtienen las mismas conclusiones.

### 5.4.6 Comparativa del rendimiento de $\text{VNS}_m$ con el AG y el FSSEBNA

Los resultados relativos al porcentaje de aciertos obtenidos con VNS y $\text{VNS}_m$ no presentan, en general, diferencias significativas. Las principales diferencias aparecen al analizar el tamaño de los subconjuntos de atributos y la convergencia. $\text{VNS}_m$ requiere menos iteraciones que VNS para encontrar subconjuntos de atributos de menor tamaño y con mayor porcentaje de aciertos (aunque, en general, estas diferencias no son significativas). Por ello, en las comparaciones que siguen se emplea $\text{VNS}_m$.

Como hasta ahora, empleamos bases de datos artificiales y reales para la comparación. En las Tablas 5.9, 5.10 y 5.11 se muestran los resultados obtenidos con las bases de datos artificiales y en las Tablas 5.12, 5.13 y 5.14 los obtenidos con bases de datos reales.

#### Problemas artificiales

La Tabla 5.9 almacena los porcentajes de acierto y tamaño de los subconjuntos encontrados aplicando el clasificador Naïve Bayes. En casi todos los casos, $\text{VNS}_m$ obtiene mejores porcentajes de aciertos y reduce en mayor medida que AG y FSSEBNA. Se detectan 2 y 7 diferencias estadísticamente significativas en el porcentaje de acierto a favor de $\text{VNS}_m$ cuando se com-

para, respectivamente, con AG y FSSEBNA. Si la comparación se hace con respecto al tamaño del subconjunto de atributos se detecta una diferencia a favor de $\text{VNS}_m$ al compararlo con AG. Las diferencias con FSSEBNA son estadísticamente significativas en casi todos los casos.

La aplicación del test de Nemenyi a todo el conjunto de datos, indica que, en porcentaje de aciertos, las diferencias entre $\text{VNS}_m$ y FSSEBNA son estadísticamente significativas a favor de $\text{VNS}_m$ al nivel $\alpha = 0{,}05$ y con el AG no existen diferencias significativas. En el tamaño de los subconjuntos de atributos, las diferencias entre $\text{VNS}_m$ y las estrategias AG y FSSEBNA son significativas al nivel $\alpha = 0{,}05$. El análisis detallado en función de las reglas de clasificación indica que $\text{VNS}_m$ mejora significativamente a FSSEBNA al nivel $\alpha = 0{,}05$ en ambos casos. $\text{VNS}_m$ es la estrategia que más reduce en ambos casos, pero, con la regla lineal, dicha reducción es estadísticamente significativa al nivel $\alpha = 0{,}05$ con respecto a AG y FSSEBNA, y, en el no lineal, sólo con FSSEBNA al nivel $\alpha = 0{,}05$.

El análisis en función del número de atributos indica que, para los problemas con 5 atributos relevantes, la diferencia es significativa al nivel $\alpha = 0{,}05$. En este caso el $\text{VNS}_m$ es el que obtiene el mayor porcentaje de aciertos. En el caso de la reducción se tiene que en ambos casos el $\text{VNS}_m$ es la estrategia que reduce en mayor medida, siendo esta diferencia significativa con $\alpha = 0{,}05$ tanto para el caso en que hay 5 como 10 atributos relevantes.

Los resultados obtenidos con el clasificador IB1 se muestran en la Tabla 5.10. Las diferencias entre $\text{VNS}_m$ y FSSEBNA son mayores que las que se presentan entre $\text{VNS}_m$ y AG cuando se analizan problema a problema. Así, sólo hay 3 diferencias estadísticamente significativas entre $\text{VNS}_m$ y AG y 10 entre $\text{VNS}_m$ y FSSEBNA. Todas estas diferencias son favorables a $\text{VNS}_m$, excepto una que es favorable a AG. Del análisis sobre el tamaño de los subconjuntos de atributos se extraen conclusiones similares. Sólo se detecta una diferencia estadísticamente significativa entre $\text{VNS}_m$ y AG, mientras que casi todas las diferencias son significativas entre $\text{VNS}_m$ y FSSEBNA.

Considerando todos los resultados, se sigue que $\text{VNS}_m$ es significativamente mejor que FSSEBNA al nivel $\alpha = 0{,}05$ cuando se analiza el porcentaje de aciertos, y significativamente mejor que AG y FSSEBNA al nivel $\alpha = 0{,}05$

cuando se analiza el tamaño de los subconjuntos de atributos.

Realizando el estudio en función de la regla de clasificación, $\text{VNS}_m$ es la estrategia que obtiene mejores porcentajes de aciertos en ambos casos, aunque sólo es significativa su mejora respecto a FSSEBNA al nivel $\alpha = 0{,}05$. $\text{VNS}_m$ también es la estrategia que reduce en mayor medida en los casos lineal y no lineal. En ambos casos, su mejora es significativa respecto a FSSEBNA al nivel $\alpha = 0{,}05$ y respecto a AG al nivel $\alpha = 0{,}1$.

Por último, analizando en función del número de atributos relevantes, se sigue que, en problemas con 5 atributos, $\text{VNS}_m$ mejora significativamente a AG y a FSSEBNA al nivel $\alpha = 0{,}05$, mientras que, en problemas con 10 atributos relevantes, la mejora sólo es estadísticamente significativa al nivel $\alpha = 0{,}05$ con AG.

El análisis del tamaño del subconjunto seleccionado indica que $\text{VNS}_m$ es estadísticamente mejor que FSSEBA al nivel $\alpha = 0{,}05$ en ambos casos, mientras que sólo es estadísticamente mejor al nivel $\alpha = 0{,}05$ que AG en problemas con 5 atributos relevantes.

Tabla 5.9: Comparativa entre la metaheurística $VNS_m$ y las estrategias AG y FSSEBNA con el clasificador Naïve Bayes. Porcentaje de acierto y reducción en conjuntos artificiales.

| conjuntos de datos | | | | Naïve Bayes | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | % aciertos | | | # atributos | | |
| l | r | $\sigma$ | d | $VNS_m$ | AG | FSSEBNA | $VNS_m$ | AG | FSSEBNA |
| 1 | 5 | 0 | 25 | 90,0 ± 2,5 | 89,2 ± 3,3 | 86,8 ± 3,2 < | 7,9 ± 2,7 | 7,6 ± 2,5 | 11,5 ± 2,8 < |
| | | | 50 | 88,8 ± 2,5 | 86,7 ± 3,8 | 83,5 ± 2,2 < | 8,6 ± 2,3 | 13,4 ± 6,9 | 20,6 ± 3,3 < |
| | | | 100 | 88,5 ± 3,5 | 84,5 ± 4,6 | 79,0 ± 1,5 < | 8,5 ± 1,4 | 21,2 ± 13,2 | 43,5 ± 3,2 ≪ |
| | | | 200 | 88,3 ± 2,9 | 79,2 ± 3,1 < | 72,6 ± 3,5 ≪ | 9,5 ± 2,6 | 26,1 ± 19,1 | 96,4 ± 7,2 ≪ |
| | | 0,1 | 25 | 90,8 ± 1,9 | 89,3 ± 1,3 | 87,8 ± 2,2 | 6,8 ± 1,5 | 8,3 ± 2,3 | 11,1 ± 2,5 |
| | | | 50 | 88,6 ± 5,3 | 86,2 ± 3,2 | 84,0 ± 2,8 | 7,2 ± 2,1 | 16,2 ± 5,6 | 20,8 ± 4,1 ≪ |
| | | | 100 | 88,2 ± 3,3 | 83,5 ± 2,9 < | 79,7 ± 2,5 < | 10,8 ± 2,7 | 22,0 ± 7,1 | 42,6 ± 6,9 ≪ |
| | | | 200 | 88,0 ± 4,5 | 80,2 ± 4,8 | 73,4 ± 3,1 < | 10,1 ± 5,2 | 24,1 ± 17,3 | 97,4 ± 7,7 ≪ |
| | 10 | 0 | 25 | 83,6 ± 3,1 | 83,7 ± 3,5 | 82,8 ± 4,2 | 13,2 ± 2,3 | 14,4 ± 3,1 | 13,5 ± 2,7 |
| | | | 50 | 79,9 ± 2,8 | 82,0 ± 2,6 | 81,3 ± 3,2 | 15,1 ± 4,0 | 19,9 ± 6,6 | 23,8 ± 2,2 < |
| | | | 100 | 78,0 ± 3,3 | 79,2 ± 4,3 | 74,7 ± 3,7 | 16,7 ± 5,9 | 22,0 ± 8,0 | 46,8 ± 3,1 ≪ |
| | | | 200 | 73,8 ± 6,8 | 72,4 ± 3,0 | 67,6 ± 2,4 | 19,1 ± 6,6 | 31,5 ± 18,3 | 99,3 ± 7,5 ≪ |
| | | 0,1 | 25 | 82,0 ± 2,5 | 83,3 ± 2,6 | 83,8 ± 2,8 | 12,2 ± 2,3 | 15,8 ± 1,8 | 13,8 ± 1,6 |
| | | | 50 | 81,1 ± 3,2 | 80,2 ± 3,6 | 80,4 ± 3,2 | 15,4 ± 4,0 | 19,3 ± 7,1 | 25,8 ± 2,4 ≪ |
| | | | 100 | 77,0 ± 5,1 | 76,6 ± 4,6 | 74,8 ± 3,0 | 17,9 ± 4,6 | 24,8 ± 5,5 | 45,5 ± 4,7 ≪ |
| | | | 200 | 73,2 ± 4,6 | 76,2 ± 4,7 | 67,4 ± 2,5 | 20,1 ± 8,8 | 28,3 ± 11,8 | 97,9 ± 8,5 ≪ |
| 2 | 5 | 0 | 25 | 89,8 ± 1,8 | 89,8 ± 1,9 | 87,7 ± 2,6 | 8,0 ± 2,7 | 7,5 ± 2,2 | 12,3 ± 2,2 < |
| | | | 50 | 88,8 ± 2,7 | 85,9 ± 2,1 | 85,6 ± 1,4 | 7,6 ± 2,0 | 11,8 ± 4,6 | 21,3 ± 3,8 < |
| | | | 100 | 88,6 ± 2,5 | 83,9 ± 3,8 | 78,1 ± 1,1 ≪ | 9,1 ± 4,1 | 12,8 ± 7,3 | 45,8 ± 2,5 ≪ |
| | | | 200 | 85,8 ± 3,8 | 80,4 ± 3,5 | 73,2 ± 3,4 | 11,9 ± 4,0 | 34,0 ± 18,1 | 92,3 ± 6,3 ≪ |
| | | 0,1 | 25 | 90,1 ± 2,3 | 89,8 ± 2,6 | 87,6 ± 2,6 | 7,3 ± 1,8 | 8,2 ± 3,2 | 11,7 ± 2,2 < |
| | | | 50 | 88,4 ± 2,7 | 86,5 ± 2,8 | 84,1 ± 2,0 | 8,7 ± 2,3 | 15,1 ± 4,8 < | 20,8 ± 3,3 ≪ |
| | | | 100 | 88,6 ± 3,9 | 84,9 ± 4,2 | 79,0 ± 2,9 | 7,9 ± 1,4 | 15,3 ± 6,3 | 47,4 ± 4,9 ≪ |
| | | | 200 | 85,1 ± 5,5 | 80,4 ± 2,3 | 72,5 ± 2,5 | 11,6 ± 6,5 | 24,9 ± 8,8 | 96,9 ± 9,1 ≪ |
| | 10 | 0 | 25 | 82,2 ± 2,9 | 81,9 ± 2,9 | 81,5 ± 3,1 | 13,3 ± 2,8 | 12,1 ± 2,4 | 13,7 ± 1,9 |
| | | | 50 | 80,6 ± 2,7 | 80,4 ± 3,1 | 79,9 ± 1,7 | 14,3 ± 2,8 | 13,0 ± 5,5 | 22,4 ± 3,0 < |
| | | | 100 | 75,2 ± 3,7 | 77,2 ± 4,9 | 74,2 ± 2,4 | 17,9 ± 6,8 | 17,2 ± 4,4 | 47,3 ± 4,5 ≪ |
| | | | 200 | 72,4 ± 3,9 | 73,5 ± 5,6 | 67,9 ± 2,7 | 17,7 ± 8,4 | 36,2 ± 25,3 | 95,9 ± 5,1 ≪ |
| | | 0,1 | 25 | 81,2 ± 4,6 | 81,4 ± 3,3 | 81,9 ± 2,6 | 12,0 ± 2,1 | 12,1 ± 3,1 | 14,0 ± 1,7 |
| | | | 50 | 74,5 ± 4,5 | 78,9 ± 3,2 | 77,5 ± 3,4 | 12,4 ± 5,1 | 17,3 ± 6,2 | 23,9 ± 3,0 ≪ |
| | | | 100 | 77,2 ± 5,1 | 78,3 ± 2,8 | 72,8 ± 3,3 | 17,1 ± 5,5 | 19,5 ± 9,7 | 49,1 ± 5,1 ≪ |
| | | | 200 | 71,7 ± 3,7 | 74,0 ± 4,6 | 65,8 ± 2,7 | 19,6 ± 9,3 | 31,2 ± 29,8 | 98,0 ± 6,7 ≪ |

$l$-regla de clasificación; $r$-número de atributos relevantes; $\sigma$-ruido Gaussiano;$VNS_m$-VNS com memoria; AG-Algoritmo Genético; FSSEBNA-Feature Subset Selection with EBNA; $>$ y $\gg$ mejora significativa en un 95 y 99 %; $<$ y $\ll$ empeora significativamente.

Tabla 5.10: Comparativa entre la metaheurística $VNS_m$ y las estrategias AG y FSSEBNA con el clasificador IB1. Porcentaje de acierto y reducción en conjuntos artificiales.

| conjuntos de datos | | | | IB1 | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | % aciertos | | | # atributos | | |
| l | r | $\sigma$ | d | $VNS_m$ | AG | FSSEBNA | $VNS_m$ | AG | FSSEBNA |
| 1 | 5 | 0 | 25 | 84,6 ± 3,5 | 79,8 ± 6,0 < | 80,8 ± 3,4 | 5,7 ± 1,1 | 7,5 ± 1,8 | 8,3 ± 1,8 |
| | | | 50 | 80,0 ± 4,0 | 77,3 ± 5,7 | 70,0 ± 2,6 ≪ | 7,1 ± 1,9 | 8,3 ± 4,1 | 19,7 ± 2,7 ≪ |
| | | | 100 | 77,7 ± 7,2 | 71,8 ± 6,2 | 62,3 ± 2,7 < | 9,6 ± 5,6 | 10,9 ± 8,5 | 47,2 ± 5,5 ≪ |
| | | | 200 | 74,5 ± 7,7 | 64,2 ± 4,8 | 57,8 ± 3,5 | 11,3 ± 7,6 | 31,1 ± 28,3 | 100,6 ± 6,2 ≪ |
| | | 0,1 | 25 | 84,4 ± 3,3 | 83,2 ± 3,6 | 80,4 ± 3,5 | 5,8 ± 0,6 | 7,1 ± 2,1 | 7,8 ± 1,6 |
| | | | 50 | 81,7 ± 4,5 | 77,9 ± 7,7 | 67,2 ± 3,1 ≪ | 7,0 ± 1,7 | 7,9 ± 4,0 | 22,2 ± 2,5 ≪ |
| | | | 100 | 79,6 ± 3,2 | 73,3 ± 6,9 | 61,9 ± 3,3 ≪ | 8,2 ± 2,2 | 12,1 ± 7,5 | 44,3 ± 5,8 ≪ |
| | | | 200 | 75,6 ± 9,1 | 65,4 ± 5,9 | 56,3 ± 2,6 < | 8,8 ± 2,3 | 27,4 ± 18,1 | 97,4 ± 5,7 ≪ |
| | 10 | 0 | 25 | 70,4 ± 5,4 | 72,4 ± 4,4 | 72,8 ± 2,8 | 9,9 ± 2,8 | 11,5 ± 2,4 | 13,2 ± 2,3 |
| | | | 50 | 66,2 ± 3,8 | 68,9 ± 2,7 | 66,5 ± 2,9 | 11,4 ± 5,5 | 10,4 ± 4,4 | 24,9 ± 4,0 |
| | | | 100 | 61,8 ± 5,2 | 63,6 ± 4,0 | 60,0 ± 2,9 | 19,7 ± 11,1 | 24,7 ± 13,7 | 52,1 ± 4,7 < |
| | | | 200 | 57,3 ± 5,6 | 61,6 ± 4,9 | 56,8 ± 5,2 | 16,3 ± 10,7 | 43,7 ± 31,3 | 96,4 ± 6,9 ≪ |
| | | 0,1 | 25 | 69,2 ± 6,6 | 72,2 ± 3,3 | 71,6 ± 2,3 | 10,2 ± 4,9 | 13,3 ± 2,4 | 13,6 ± 2,8 ≪ |
| | | | 50 | 66,1 ± 7,5 | 66,4 ± 5,5 | 65,4 ± 2,7 | 11,9 ± 2,8 | 11,9 ± 5,0 | 23,9 ± 3,5 < |
| | | | 100 | 63,2 ± 5,5 | 65,7 ± 4,1 | 58,7 ± 2,7 | 19,8 ± 6,1 | 17,2 ± 11,1 | 50,3 ± 5,7 ≪ |
| | | | 200 | 57,8 ± 3,5 | 61,8 ± 5,9 | 56,4 ± 2,4 | 19,7 ± 9,5 | 38,7 ± 35,8 | 97,4 ± 5,7 ≪ |
| 2 | 5 | 0 | 25 | 83,0 ± 3,5 | 81,4 ± 4,0 | 79,4 ± 4,5 | 6,1 ± 1,2 | 7,3 ± 2,9 | 8,6 ± 2,1 |
| | | | 50 | 80,2 ± 6,3 | 79,3 ± 4,5 | 70,3 ± 3,9 | 7,0 ± 2,6 | 7,2 ± 2,6 | 20,1 ± 4,7 < |
| | | | 100 | 76,6 ± 3,9 | 73,4 ± 4,1 | 63,4 ± 2,5 < | 8,1 ± 3,0 | 9,0 ± 4,2 | 50,2 ± 6,1 ≪ |
| | | | 200 | 76,1 ± 4,9 | 66,2 ± 4,5 ≪ | 59,0 ± 3,6 < | 7,8 ± 3,1 | 32,1 ± 27,5 | 95,7 ± 7,3 ≪ |
| | | 0,1 | 25 | 83,2 ± 4,5 | 82,6 ± 5,0 | 80,2 ± 3,2 | 5,9 ± 1,6 | 7,4 ± 2,2 | 8,6 ± 1,3 < |
| | | | 50 | 81,6 ± 4,1 | 79,8 ± 3,7 | 69,2 ± 3,4 < | 6,1 ± 1,5 | 8,2 ± 2,7 | 19,4 ± 2,9 ≪ |
| | | | 100 | 79,1 ± 3,7 | 74,5 ± 8,1 | 62,6 ± 3,4 ≪ | 6,1 ± 1,4 | 11,9 ± 10,8 | 48,8 ± 4,4 ≪ |
| | | | 200 | 75,4 ± 3,6 | 67,8 ± 6,0 | 58,4 ± 3,3 ≪ | 6,7 ± 2,1 | 28,6 ± 29,3 | 102,1 ± 5,9 ≪ |
| | 10 | 0 | 25 | 67,0 ± 7,6 | 69,0 ± 3,3 | 68,8 ± 3,9 | 12,8 ± 5,1 | 11,2 ± 2,7 | 12,1 ± 0,9 |
| | | | 50 | 65,6 ± 4,9 | 66,4 ± 4,0 | 63,2 ± 3,8 | 10,7 ± 5,2 | 13,1 ± 7,1 | 25,3 ± 5,2 < |
| | | | 100 | 62,9 ± 6,1 | 63,8 ± 6,0 ≫ | 60,1 ± 2,7 | 14,0 ± 5,7 | 23,6 ± 11,5 | 49,8 ± 3,0 ≪ |
| | | | 200 | 60,2 ± 3,7 | 57,8 ± 4,8 | 55,8 ± 3,8 | 16,3 ± 8,1 | 60,5 ± 36,2 | 98,7 ± 6,3 ≪ |
| | | 0,1 | 25 | 70,3 ± 4,7 | 67,8 ± 3,8 | 69,8 ± 3,4 | 10,1 ± 1,6 | 10,3 ± 3,4 | 11,8 ± 2,3 |
| | | | 50 | 64,3 ± 4,9 | 68,1 ± 3,1 | 62,6 ± 5,5 | 7,8 ± 3,7 | 12,8 ± 5,7 | 25,6 ± 3,1 ≪ |
| | | | 100 | 60,7 ± 3,9 | 64,4 ± 4,8 | 58,5 ± 2,7 | 21,1 ± 10,4 | 18,7 ± 10,3 | 49,5 ± 5,3 < |
| | | | 200 | 59,2 ± 5,8 | 62,8 ± 5,3 | 55,8 ± 2,2 | 16,8 ± 8,7 | 23,6 ± 27,6 < | 97,7 ± 4,9 ≪ |

$l$-regla de clasificación; $r$-número de atributos relevantes; $\sigma$-ruido Gaussiano;$VNS_m$-VNS com memoria; AG-Algoritmo Genético; FSSEBNA-Feature Subset Selection with EBNA; $>$ y $\gg$ mejora significativa en un 95 y 99 %; $<$ y $\ll$ empeora significativamente.

Tabla 5.11: Comparativa entre la metaheurística $VNS_m$ y las estrategias AG y FSSEBNA con el clasificador J48. Porcentaje de acierto y reducción en conjuntos artificiales.

| conjuntos de datos | | | | J48 | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | % aciertos | | | # atributos | | |
| l | r | $\sigma$ | d | $VNS_m$ | AG | FSSEBNA | $VNS_m$ | AG | FSSEBNA |
| 1 | 5 | 0 | 25 | 77,5 ± 3,9 | 77,2 ± 3,8 | 77,4 ± 3,0 | 6,3 ± 1,3 | 8,5 ± 2,4 | 10,1 ± 2,5 < |
| | | | 50 | 77,3 ± 4,7 | 74,0 ± 3,3 | 75,4 ± 3,3 | 8,0 ± 1,9 | 14,6 ± 5,6 < | 23,1 ± 2,9 ≪ |
| | | | 100 | 75,5 ± 4,0 | 74,4 ± 6,9 | 71,4 ± 3,7 | 7,6 ± 2,3 | 30,3 ± 13,3 | 47,9 ± 6,2 ≪ |
| | | | 200 | 74,3 ± 3,4 | 74,1 ± 4,1 | 71,3 ± 3,4 | 8,9 ± 1,9 | 41,4 ± 23,4 | 98,3 ± 6,5 ≪ |
| | | 0,1 | 25 | 76,0 ± 4,3 | 76,0 ± 4,3 | 75,0 ± 3,7 | 4,7 ± 1,1 | 8,7 ± 1,7 < | 10,4 ± 2,8 ≪ |
| | | | 50 | 76,1 ± 4,6 | 77,8 ± 4,0 | 74,6 ± 5,0 | 7,2 ± 2,1 | 13,7 ± 5,2 < | 23,0 ± 2,1 ≪ |
| | | | 100 | 73,9 ± 2,9 | 75,0 ± 3,2 | 71,9 ± 4,7 | 8,7 ± 2,4 | 21,7 ± 13,5 | 49,8 ± 5,1 ≪ |
| | | | 200 | 74,9 ± 4,4 | 72,5 ± 3,9 | 70,1 ± 3,4 | 9,2 ± 1,7 | 55,2 ± 29,9 | 98,3 ± 7,3 ≪ |
| | 10 | 0 | 25 | 67,4 ± 4,7 | 67,3 ± 3,4 | 67,6 ± 3,6 | 7,5 ± 3,1 | 11,9 ± 2,8 | 12,4 ± 1,8 |
| | | | 50 | 66,4 ± 3,1 | 66,7 ± 2,7 | 67,5 ± 4,4 | 8,0 ± 3,1 | 18,0 ± 4,2 < | 23,5 ± 3,2 ≪ |
| | | | 100 | 65,0 ± 3,6 | 64,2 ± 5,7 | 64,2 ± 2,3 | 10,7 ± 3,9 | 28,3 ± 14,2 | 51,6 ± 3,7 ≪ |
| | | | 200 | 62,4 ± 5,1 | 62,6 ± 4,0 | 61,0 ± 3,3 | 12,6 ± 2,4 | 59,0 ± 33,6 | 96,0 ± 6,7 ≪ |
| | | 0,1 | 25 | 65,5 ± 3,8 | 66,4 ± 2,6 | 67,2 ± 3,4 | 7,9 ± 2,4 | 10,5 ± 2,2 | 12,5 ± 2,7 |
| | | | 50 | 63,5 ± 4,3 | 63,7 ± 4,3 | 64,1 ± 4,2 | 9,4 ± 3,6 | 18,9 ± 7,8 | 22,0 ± 4,6 < |
| | | | 100 | 64,8 ± 4,0 | 63,0 ± 3,0 | 62,8 ± 3,2 | 11,6 ± 4,7 | 28,9 ± 17,9 < | 50,3 ± 3,5 ≪ |
| | | | 200 | 63,1 ± 4,3 | 62,4 ± 4,7 | 61,4 ± 4,5 | 12,9 ± 3,6 | 58,9 ± 30,3 ≪ | 100,0 ± 10,8 ≪ |
| 2 | 5 | 0 | 25 | 82,4 ± 2,7 | 81,1 ± 2,8 | 79,0 ± 4,1 < | 6,5 ± 1,4 | 8,2 ± 2,6 | 11,8 ± 2,0 < |
| | | | 50 | 80,7 ± 3,6 | 79,2 ± 3,4 | 79,0 ± 2,5 | 7,3 ± 1,8 | 18,6 ± 7,4 | 24,0 ± 4,5 ≪ |
| | | | 100 | 80,5 ± 3,4 | 78,6 ± 3,6 | 75,6 ± 3,7 | 8,4 ± 1,9 | 28,1 ± 16,8 | 49,8 ± 4,9 ≪ |
| | | | 200 | 79,8 ± 3,1 | 76,2 ± 4,6 | 74,4 ± 4,4 | 8,5 ± 2,0 | 53,7 ± 24,8 | 98,4 ± 7,2 ≪ |
| | | 0,1 | 25 | 79,0 ± 3,0 | 79,2 ± 3,7 | 78,2 ± 3,3 | 5,2 ± 2,2 | 9,1 ± 4,5 | 11,8 ± 2,9 |
| | | | 50 | 79,9 ± 3,9 | 78,9 ± 2,4 | 78,7 ± 2,0 | 6,0 ± 1,8 | 17,1 ± 6,0 | 22,2 ± 2,8 ≪ |
| | | | 100 | 78,4 ± 1,5 | 77,1 ± 4,0 | 76,4 ± 2,8 | 7,9 ± 1,7 | 31,4 ± 17,4 | 48,0 ± 3,7 ≪ |
| | | | 200 | 80,2 ± 3,8 | 75,6 ± 2,3 | 75,0 ± 2,7 < | 8,7 ± 1,9 | 48,5 ± 29,7 | 101,6 ± 7,0 ≪ |
| | 10 | 0 | 25 | 67,6 ± 3,6 | 67,7 ± 3,5 | 68,1 ± 2,7 | 7,5 ± 2,8 | 11,6 ± 2,5 | 12,8 ± 2,1 |
| | | | 50 | 67,6 ± 2,4 | 67,7 ± 4,5 | 64,6 ± 3,6 | 9,0 ± 2,2 | 15,0 ± 5,3 | 23,3 ± 2,4 < |
| | | | 100 | 66,1 ± 3,6 | 67,7 ± 5,5 | 64,0 ± 3,2 | 12,2 ± 3,7 | 25,5 ± 17,3 | 49,0 ± 4,8 ≪ |
| | | | 200 | 63,7 ± 4,9 | 63,9 ± 2,8 | 60,6 ± 3,9 | 12,7 ± 3,6 | 45,3 ± 30,8 | 102,9 ± 7,1 ≪ |
| | | 0,1 | 25 | 66,8 ± 2,5 | 68,0 ± 4,0 | 66,0 ± 2,1 | 8,1 ± 2,6 | 10,2 ± 2,7 | 13,0 ± 1,6 |
| | | | 50 | 64,8 ± 5,9 | 65,2 ± 2,9 | 64,3 ± 3,5 | 9,0 ± 2,4 | 13,6 ± 5,4 | 24,0 ± 3,5 ≪ |
| | | | 100 | 64,9 ± 5,0 | 64,1 ± 3,2 | 64,3 ± 2,8 | 9,7 ± 3,1 | 21,5 ± 13,0 | 48,2 ± 3,8 ≪ |
| | | | 200 | 63,4 ± 4,6 | 62,0 ± 5,6 | 59,5 ± 2,2 | 12,6 ± 3,4 | 45,5 ± 35,8 | 96,7 ± 6,1 ≪ |

$l$-regla de clasificación; $r$-número de atributos relevantes; $\sigma$-ruido Gaussiano;$VNS_m$-VNS com memoria; AG-Algoritmo Genético; FSSEBNA-Feature Subset Selection with EBNA; $>$ y $\gg$ mejora significativa en un 95 y 99 %; $<$ y $\ll$ empeora significativamente.

Los resultados obtenidos con el clasificador J48 pueden consultarse en la Tabla 5.11. Como en los casos anteriores las mayores diferencias de comportamiento se dan entre $\text{VNS}_m$ y FSSEBNA. Así, no hay diferencias estadísticamente significativas entre $\text{VNS}_m$ y AG cuando se analiza el porcentaje de acierto, y en 6 casos la reducción es significativa a favor del $\text{VNS}_m$. La comparativa con el FSSEBNA revela que en el acierto sólo hay 2 diferencias estadísticamente significativas favorables a $\text{VNS}_m$ y la reducción obtenida por $\text{VNS}_m$ es, en casi todos los problemas, mayor de forma signifivativa al nivel $\alpha = 0{,}05$.

Las conclusiones aplicando el test de Nemenyi son similares a las obtenidos con IB1, de modo que, en acierto y reducción, las diferencias entre $\text{VNS}_m$ y FSSEBNA son estadísticamente significativas a favor de $\text{VNS}_m$ al nivel $\alpha = 0{,}05$. Detallando el análisis en función de la regla de clasificación, se sigue que en el caso lineal, las diferencias en el acierto con AG son estadísticamente significativas a favor de $\text{VNS}_m$ al nivel $\alpha = 0{,}1$; en el caso no lineal, lo son al nivel $\alpha = 0{,}05$ a favor también de $\text{VNS}_m$. $\text{VNS}_m$ es la estrategia que significativamente reduce en mayor medida al nivel $\alpha = 0{,}05$ tanto en el caso lineal como en el no lineal. Los tamaños promedios sobre todos los problemas dan, respectivamente, subconjuntos de tamaño 9, 26 y 46 para $\text{VNS}_m$, AG y FSSEBNA. Analizando los problemas en función del número de atributos relevantes se sigue que, en porcentaje de aciertos, sólo son significativas las diferencias en problemas con 5 atributos relevantes. La reducción es estadísticamente significativa en ambos casos a favor de $\text{VNS}_m$ que es la estrategia que reduce en mayor medida.

### Problemas reales

Los resultados obtenidos al usar el clasificador Naïve Bayes se muestran en la Tabla 5.12. En porcentaje de aciertos, las diferencias estadísticamente significativas entre $\text{VNS}_m$ y AG se encuentran en el conjunto *spl*, y en los problemas *lym*, *aut* y *rat* con FSSEBNA. De las 4 diferencias significativas, 3 son desfavorables a $\text{VNS}_m$. El análisis sobre el tamaño del subconjunto de atributos revela un mayor número de diferencias significativas: 5 con AG y

15 con FSSEBNA. Todas son favorables a $VNS_m$.

Los resultados obtenidos al usar el clasificador IB1, pueden consultarse en la Tabla 5.13. En lo que al acierto se refiere, sólo para el problema *dna* se encuentran diferencias estadísticamente significativas entre $VNS_m$ y AG y FSSEBNA. El comportamiento en la reducción de atributos es similar al observado con Naïve Bayes. Así, la mayoría de las diferencias estadísticamente significativas se encuentran en los problemas de mayor tamaño.

Los resultados obtenidos con J48 se muestran en la Tabla 5.14. En porcentaje de aciertos, el test F no detecta diferencias estadísticamente significativas excepto en el problema *win* a favor de FSSEBNA; en tamaño de los subconjuntos de atributos, el comportamiento es similar al presentado al usar Naïve Bayes e IB1.

Tabla 5.12: Comparativa entre la metaheurística $VNS_m$ y las estrategias AG y FSSEBNA con el clasificador Naïve Bayes. Porcentaje de acierto y reducción en conjuntos reales.

| cdd | Naïve Bayes | | | | | |
|---|---|---|---|---|---|---|
| | % aciertos | | | # atributos | | |
| | $VNS_m$ | AG | FSSEBNA | $VNS_m$ | AG | FSSEBNA |
| pdi | 77,5 ± 2,0 | 77,8 ± 1,6 | 78,0 ± 1,9 | 3,8 ± 0,9 | 4,3 ± 0,7 | 4,1 ± 0,9 |
| gla | 57,1 ± 3,0 | 55,4 ± 4,8 | 53,5 ± 5,9 | 3,5 ± 1,5 | 3,5 ± 1,4 | 4,2 ± 1,7 |
| bca | 68,5 ± 3,1 | 69,9 ± 2,3 | 70,1 ± 3,4 | 3,4 ± 1,5 | 3,9 ± 1,4 | 4,7 ± 1,3 |
| bwi | 96,1 ± 1,2 | 95,9 ± 1,3 | 96,9 ± 1,0 | 4,2 ± 0,6 | 4,2 ± 1,3 | 5,7 ± 1,1 < |
| win | 96,1 ± 2,7 | 95,4 ± 2,0 | 95,6 ± 2,9 | 5,3 ± 1,3 | 5,5 ± 1,0 | 7,7 ± 1,8 |
| hst | 83,5 ± 1,7 | 82,5 ± 2,2 | 82,9 ± 3,4 | 6,5 ± 1,8 | 6,8 ± 1,2 | 8,1 ± 1,4 |
| hhu | 81,6 ± 2,9 | 81,9 ± 2,4 | 81,8 ± 2,7 | 4,8 ± 1,8 | 6,0 ± 1,6 | 7,2 ± 1,7 |
| hcl | 80,5 ± 4,3 | 80,1 ± 3,7 | 82,3 ± 3,2 | 5,6 ± 1,9 | 6,6 ± 2,0 < | 7,3 ± 2,1 |
| vow | 66,7 ± 3,1 | 67,7 ± 3,9 | 67,6 ± 3,1 | 6,9 ± 1,4 | 8,0 ± 1,4 | 7,6 ± 1,3 |
| cra | 85,5 ± 1,3 | 85,5 ± 1,2 | 85,4 ± 1,4 | 5,2 ± 1,4 | 5,2 ± 1,4 | 6,7 ± 1,9 |
| ptu | 42,4 ± 3,6 | 43,5 ± 2,5 | 43,3 ± 2,8 | 9,9 ± 2,6 | 10,1 ± 1,6 | 10,2 ± 1,9 |
| vot | 95,1 ± 0,9 | 95,4 ± 0,8 | 94,5 ± 1,0 | 2,2 ± 1,1 | 2,8 ± 1,1 | 4,6 ± 1,0 |
| lym | 77,6 ± 3,3 | 77,7 ± 3,6 | 80,0 ± 4,0 > | 5,5 ± 3,5 | 7,8 ± 2,7 | 9,2 ± 2,3 |
| veh | 59,3 ± 3,3 | 59,1 ± 2,8 | 59,8 ± 2,9 | 7,8 ± 2,6 | 9,5 ± 1,8 | 9,6 ± 1,7 |
| hep | 81,8 ± 4,8 | 84,0 ± 3,3 | 83,6 ± 4,3 | 3,5 ± 2,0 | 7,6 ± 2,1 | 10,1 ± 2,3 ≪ |
| hco | 82,8 ± 4,0 | 83,6 ± 3,1 | 83,6 ± 2,2 | 5,5 ± 1,7 | 5,6 ± 1,7 | 7,9 ± 2,0 |
| aut | 61,5 ± 6,1 | 63,7 ± 5,4 | 63,5 ± 4,8 > | 6,0 ± 1,6 | 7,6 ± 1,8 | 10,2 ± 1,9 < |
| abp | 96,5 ± 0,3 | 96,5 ± 0,4 | 96,5 ± 0,4 | 2,3 ± 0,8 | 7,1 ± 3,0 | 12,1 ± 2,3 < |
| ion | 91,1 ± 2,9 | 92,6 ± 2,8 | 91,9 ± 2,5 | 7,2 ± 1,5 | 12,8 ± 2,5 | 14,6 ± 3,0 < |
| soy | 89,9 ± 2,4 | 89,8 ± 2,0 | 90,6 ± 1,3 | 16,5 ± 2,4 | 21,4 ± 3,1 | 22,6 ± 2,8 < |
| kvk | 94,3 ± 0,4 | 93,5 ± 0,8 | 93,9 ± 0,7 | 5,0 ± 0,0 | 13,6 ± 2,9 ≪ | 14,7 ± 1,7 ≪ |
| ann | 90,1 ± 1,8 | 91,6 ± 1,3 | 92,2 ± 1,7 | 7,0 ± 3,1 | 18,7 ± 3,6 | 19,9 ± 2,9 < |
| lca | 46,3 ± 11,1 | 44,4 ± 9,5 | 47,5 ± 8,4 | 3,1 ± 1,0 | 7,9 ± 4,2 | 25,8 ± 3,3 ≪ |
| dnp | 82,8 ± 5,0 | 84,2 ± 6,1 | 87,0 ± 5,1 | 6,0 ± 3,6 | 22,5 ± 8,2 < | 27,9 ± 4,1 < |
| son | 74,5 ± 4,8 | 71,3 ± 4,7 | 75,0 ± 3,5 | 10,1 ± 6,2 | 16,7 ± 8,8 | 27,3 ± 3,7 < |
| dna | 94,8 ± 0,6 | 95,3 ± 0,5 | 95,6 ± 0,3 | 11,7 ± 3,1 | 34,0 ± 4,4 ≪ | 34,5 ± 3,5 ≪ |
| spl | 94,9 ± 0,6 | 95,7 ± 0,4 > | 95,7 ± 0,6 | 15,3 ± 3,4 | 35,2 ± 2,1 ≪ | 34,0 ± 2,9 < |
| aud | 69,2 ± 3,9 | 67,3 ± 3,0 | 68,5 ± 1,3 | 8,7 ± 2,0 | 16,6 ± 6,7 | 29,7 ± 2,5 ≪ |
| rat | 69,2 ± 6,2 | 69,3 ± 5,8 | 55,4 ± 5,7 < | 9,4 ± 5,6 | 4,1 ± 2,2 | 67,3 ± 6,2 ≪ |
| arr | 66,2 ± 2,0 | 64,6 ± 1,6 | 65,4 ± 1,9 | 12,6 ± 4,4 | 74,8 ± 30,6 | 138,4 ± 10,6 ≪ |

VNS-Búsqueda Dispersa; AG-Algoritmos Genético; FSSEBNA-Feature Subset Selection with EBNA; > y ≫ mejora significativa en un 95 y 99 %; < y ≪ empeora significativamente.

Tabla 5.13: Comparativa entre la metaheurística $VNS_m$ y las estrategias AG y FSSEBNA con el clasificador IB1. Porcentaje de acierto y reducción en conjuntos reales.

| cdd | IB1 | | | | | |
|---|---|---|---|---|---|---|
| | % aciertos | | | # atributos | | |
| | $VNS_m$ | AG | FSSEBNA | $VNS_m$ | AG | FSSEBNA |
| pdi | 68,5 ± 3,1 | 68,8 ± 2,2 | 68,1 ± 2,2 | 3,7 ± 0,9 | 4,2 ± 0,9 | 4,2 ± 0,8 |
| gla | 68,2 ± 2,9 | 67,6 ± 4,0 | 68,0 ± 3,0 | 5,5 ± 1,1 | 4,8 ± 0,6 | 5,3 ± 1,5 |
| bca | 62,1 ± 8,1 | 62,8 ± 9,6 | 62,3 ± 8,6 | 2,7 ± 1,2 | 3,1 ± 1,4 | 3,6 ± 1,5 |
| bwi | 94,7 ± 1,0 | 94,7 ± 1,1 | 94,6 ± 1,0 | 4,3 ± 1,3 | 4,4 ± 1,3 | 5,0 ± 1,1 |
| win | 93,8 ± 2,9 | 94,9 ± 2,6 | 95,4 ± 2,0 | 5,0 ± 1,6 | 6,2 ± 1,8 | 7,5 ± 1,1 |
| hst | 72,6 ± 5,0 | 75,9 ± 3,7 | 75,6 ± 3,3 | 5,9 ± 2,6 | 5,9 ± 2,0 | 7,0 ± 1,2 |
| hhu | 72,2 ± 12,5 | 74,4 ± 5,4 | 74,3 ± 4,5 | 3,9 ± 2,0 | 5,4 ± 2,2 | 5,2 ± 2,4 |
| hcl | 71,7 ± 8,6 | 74,4 ± 4,6 | 75,4 ± 4,0 | 6,2 ± 2,1 | 5,4 ± 1,6 | 6,8 ± 1,1 |
| vow | 94,2 ± 1,4 | 93,2 ± 1,1 | 94,4 ± 1,1 | 8,8 ± 0,8 | 8,6 ± 1,3 | 9,3 ± 0,8 |
| cra | 73,5 ± 13,0 | 81,1 ± 3,3 | 81,2 ± 2,2 | 3,5 ± 2,0 | 4,4 ± 1,6 | 6,5 ± 2,1 |
| ptu | 32,2 ± 5,5 | 31,6 ± 4,7 | 31,6 ± 4,8 | 10,0 ± 1,3 | 11,0 ± 1,9 | 10,2 ± 1,5 |
| vot | 94,5 ± 2,0 | 94,3 ± 1,9 | 94,0 ± 1,9 | 3,4 ± 1,9 | 4,7 ± 1,1 | 5,7 ± 1,6 |
| lym | 79,2 ± 3,6 | 75,3 ± 5,7 | 77,3 ± 6,6 | 7,4 ± 2,3 | 8,2 ± 2,0 | 9,7 ± 2,2 |
| veh | 68,6 ± 2,9 | 68,3 ± 1,9 | 68,8 ± 1,7 | 7,5 ± 1,7 | 8,4 ± 2,5 | 9,4 ± 1,8 |
| hep | 80,6 ± 5,6 | 79,1 ± 4,2 | 77,4 ± 6,4 | 4,5 ± 2,4 | 6,9 ± 2,1 | 8,5 ± 1,4 |
| hco | 79,9 ± 5,7 | 81,4 ± 3,6 | 81,0 ± 2,7 | 6,5 ± 3,1 | 6,5 ± 3,6 | 9,3 ± 2,4 |
| aut | 69,4 ± 5,7 | 70,1 ± 3,7 | 67,8 ± 4,5 | 4,6 ± 1,8 | 4,7 ± 2,7 | 12,3 ± 3,1 < |
| abp | 95,8 ± 0,8 | 96,4 ± 0,7 | 96,3 ± 0,4 | 3,3 ± 4,1 | 14,0 ± 3,4 | 17,6 ± 2,3 < |
| ion | 87,1 ± 1,9 | 87,2 ± 3,0 | 88,1 ± 2,0 | 5,4 ± 2,2 | 7,0 ± 3,3 | 12,1 ± 2,6 ≪ |
| soy | 88,6 ± 3,2 | 87,3 ± 1,2 | 90,3 ± 2,3 | 18,2 ± 3,4 | 21,7 ± 3,8 | 22,0 ± 1,6 |
| kvk | 94,7 ± 3,3 | 94,2 ± 2,7 | 95,4 ± 1,4 | 9,3 ± 2,5 | 17,6 ± 2,7 < | 16,6 ± 2,1 ≪ |
| ann | 96,5 ± 1,8 | 95,9 ± 1,4 | 95,0 ± 2,0 | 8,9 ± 0,9 | 16,1 ± 2,9 ≪ | 20,1 ± 3,0 < |
| lca | 47,5 ± 14,5 | 38,8 ± 7,1 | 44,4 ± 13,0 | 4,9 ± 1,7 | 13,4 ± 7,7 | 23,9 ± 4,7 < |
| dnp | 74,0 ± 9,2 | 71,7 ± 7,8 | 75,8 ± 4,2 | 9,5 ± 4,7 | 19,5 ± 9,4 | 29,0 ± 2,9 < |
| son | 80,5 ± 2,8 | 79,9 ± 4,0 | 81,9 ± 2,4 | 12,9 ± 4,6 | 20,6 ± 4,9 | 27,9 ± 3,5 ≪ |
| dna | 85,4 ± 1,8 | 80,1 ± 1,1 < | 76,0 ± 1,1 < | 6,3 ± 1,3 | 6,6 ± 1,6 | 27,4 ± 3,0 ≪ |
| spl | 84,0 ± 4,0 | 75,7 ± 1,9 | 76,7 ± 1,7 | 5,5 ± 1,2 | 25,7 ± 7,1 < | 28,0 ± 4,4 ≪ |
| aud | 68,5 ± 5,6 | 70,6 ± 5,5 | 70,4 ± 5,7 | 11,8 ± 2,4 | 33,4 ± 7,9 < | 35,0 ± 3,0 ≪ |
| rat | 68,4 ± 6,4 | 65,4 ± 5,6 | 64,8 ± 4,1 | 4,4 ± 2,1 | 26,0 ± 16,9 | 68,7 ± 6,2 ≪ |
| arr | 60,2 ± 2,3 | 55,1 ± 5,3 | 56,0 ± 3,3 | 22,2 ± 3,9 | 67,7 ± 30,1 | 136,0 ± 5,2 ≪ |

VNS-Búsqueda Dispersa; AG-Algoritmos Genético; FSSEBNA-Feature Subset Selection with EBNA; > y ≫ mejora significativa en un 95 y 99 %; < y ≪ empeora significativamente.

Tabla 5.14: Comparativa entre la metaheurística $VNS_m$ y las estrategias AG y FSSEBNA con el clasificador J48. Porcentaje de acierto y reducción en conjuntos reales.

| cdd | J48 | | | | | |
|---|---|---|---|---|---|---|
| | % aciertos | | | # atributos | | |
| | $VNS_m$ | AG | FSSEBNA | $VNS_m$ | AG | FSSEBNA |
| pdi | 73,2 ± 2,1 | 73,9 ± 2,1 | 73,0 ± 2,0 | 1,8 ± 1,0 | 2,8 ± 1,4 | 3,4 ± 1,0 |
| gla | 63,9 ± 5,8 | 65,7 ± 3,8 | 62,1 ± 5,8 | 4,2 ± 0,9 | 4,2 ± 0,6 | 4,8 ± 0,6 |
| bca | 71,3 ± 2,7 | 71,9 ± 2,7 | 72,0 ± 2,7 | 2,2 ± 1,2 | 2,3 ± 0,9 | 4,2 ± 1,2 |
| bwi | 94,2 ± 1,1 | 93,9 ± 1,2 | 94,2 ± 0,9 | 3,1 ± 1,4 | 3,0 ± 1,2 | 3,6 ± 1,3 |
| win | 89,6 ± 3,9 | 90,6 ± 2,7 | 91,9 ± 3,8 > | 2,4 ± 0,5 | 2,9 ± 1,0 | 5,9 ± 2,1 < |
| hst | 73,8 ± 6,2 | 75,5 ± 2,4 | 77,3 ± 4,4 | 2,4 ± 1,6 | 4,3 ± 1,5 | 6,1 ± 1,5 < |
| hhu | 78,8 ± 2,9 | 78,8 ± 3,4 | 80,5 ± 2,3 | 1,3 ± 0,5 | 2,6 ± 1,7 | 6,3 ± 1,4 ≪ |
| hcl | 74,6 ± 4,1 | 76,2 ± 3,2 | 77,3 ± 2,2 | 2,7 ± 1,6 | 4,6 ± 1,4 | 6,2 ± 1,3 |
| vow | 69,1 ± 2,9 | 69,1 ± 3,1 | 70,9 ± 2,4 | 6,9 ± 0,9 | 6,8 ± 1,2 | 7,1 ± 1,3 |
| cra | 85,5 ± 1,5 | 85,4 ± 1,1 | 85,3 ± 1,6 | 1,8 ± 1,6 | 6,3 ± 1,8 | 7,6 ± 1,6 < |
| ptu | 36,9 ± 3,2 | 38,2 ± 3,4 | 39,2 ± 3,4 | 7,3 ± 1,6 | 8,5 ± 2,1 | 9,1 ± 1,8 |
| vot | 95,6 ± 0,7 | 95,5 ± 0,7 | 95,4 ± 0,9 | 1,0 ± 0,0 | 1,6 ± 1,3 | 8,3 ± 1,3 ≪ |
| lym | 75,8 ± 3,3 | 75,4 ± 4,4 | 74,5 ± 6,5 | 2,3 ± 0,8 | 3,5 ± 1,8 | 8,8 ± 2,9 |
| veh | 67,4 ± 2,6 | 67,8 ± 2,5 | 68,7 ± 1,7 | 6,9 ± 2,8 | 10,6 ± 1,8 | 11,1 ± 1,7 |
| hep | 79,7 ± 3,1 | 78,1 ± 4,4 | 78,7 ± 4,2 | 1,8 ± 0,8 | 3,9 ± 2,8 | 9,3 ± 2,7 ≪ |
| hco | 84,5 ± 2,8 | 85,1 ± 2,4 | 85,2 ± 2,0 | 2,8 ± 0,6 | 5,0 ± 2,0 | 10,0 ± 2,7 |
| aut | 62,0 ± 4,7 | 64,8 ± 5,5 | 63,6 ± 6,5 | 3,5 ± 1,8 | 7,5 ± 2,5 | 12,0 ± 2,6 < |
| abp | 97,1 ± 0,3 | 97,0 ± 0,3 | 97,0 ± 0,3 | 2,5 ± 1,0 | 12,6 ± 2,9 ≪ | 14,6 ± 2,5 ≪ |
| ion | 88,2 ± 1,6 | 88,3 ± 2,1 | 87,8 ± 2,3 | 3,8 ± 1,3 | 9,0 ± 3,9 | 15,9 ± 3,3 ≪ |
| soy | 87,3 ± 1,9 | 88,7 ± 2,8 | 88,9 ± 1,0 | 15,5 ± 3,0 | 21,3 ± 2,7 | 20,8 ± 2,1 |
| kvk | 97,1 ± 1,3 | 98,2 ± 0,7 | 98,6 ± 0,4 | 8,5 ± 1,7 | 24,7 ± 3,9 ≪ | 26,2 ± 2,3 ≪ |
| ann | 86,3 ± 4,0 | 89,7 ± 2,2 | 89,1 ± 2,9 | 4,3 ± 1,6 | 18,2 ± 4,0 ≪ | 19,7 ± 2,7 ≪ |
| lca | 37,5 ± 8,8 | 39,4 ± 5,1 | 42,5 ± 6,5 | 1,3 ± 0,7 | 6,5 ± 4,6 < | 28,5 ± 4,8 ≪ |
| dnp | 74,9 ± 2,7 | 75,1 ± 3,4 | 73,4 ± 5,7 | 1,3 ± 0,7 | 3,7 ± 4,1 | 25,0 ± 3,9 ≪ |
| son | 68,0 ± 2,8 | 69,1 ± 5,0 | 68,5 ± 5,5 | 5,5 ± 1,8 | 19,0 ± 8,8 < | 27,2 ± 4,8 ≪ |
| dna | 92,9 ± 0,8 | 93,0 ± 0,7 | 93,0 ± 0,7 | 8,6 ± 1,2 | 28,9 ± 4,6 ≪ | 32,4 ± 3,7 ≪ |
| spl | 92,3 ± 0,6 | 92,5 ± 0,4 | 92,5 ± 0,8 | 7,9 ± 0,9 | 33,2 ± 5,3 ≪ | 31,5 ± 4,4 ≪ |
| aud | 70,3 ± 3,2 | 71,6 ± 1,9 | 72,8 ± 4,0 | 6,0 ± 1,4 | 26,0 ± 5,0 ≪ | 35,5 ± 6,0 ≪ |
| rat | 71,3 ± 5,1 | 70,7 ± 4,8 | 70,5 ± 4,9 | 5,0 ± 1,8 | 36,9 ± 18,1 | 66,6 ± 7,8 ≪ |
| arr | 66,5 ± 3,3 | 64,6 ± 3,1 | 64,7 ± 1,8 | 12,8 ± 2,9 | 114,0 ± 30,9 < | 140,7 ± 6,6 ≪ |

VNS-Búsqueda Dispersa; AG-Algoritmos Genético; FSSEBNA-Feature Subset Selection with EBNA; > y ≫ mejora significativa en un 95 y 99 %; < y ≪ empeora significativamente.

### 5.4.7 Tiempos de ejecución

Finalmente, analizamos los tiempos de ejecución de las estrategias de selección de atributos presentadas en este capítulo. Los resultados correspondientes a los problemas artificiales se hallan en las Tablas 5.15, 5.16 y 5.17, y los correspondientes a los problemas reales se detallan en las Tablas 5.18, 5.19 y 5.20. A continuación, se describen en detalle los resultados.

#### Problemas artificiales

Con el clasificador Naïve Bayes se obtuvieron los tiempos de ejecución que aparecen en la Tabla 5.15. Como puede observarse, la búsqueda SFFS es la que menos tiempo tarda, seguida de $\text{VNS}_m$, AG y FSSEBNA. Las diferencias entre $\text{VNS}_m$ y SFFS, $\text{VNS}_m$ y AG y $\text{VNS}_m$ y FSSEBNA son estadísticamente significativas en 13, 20 y 26 conjuntos de datos, respectivamente. Las diferencias significativas son favorables a $\text{VNS}_m$ cuando se compara esta estrategia con GA y FSSEBNA, y desfavorables cuando se compara con SFFS.

Analizando los resultados en función de la regla de clasificación, se encuentra que $\text{VNS}_m$ es estadísticamente más eficiente que AG al nivel $\alpha = 0{,}1$ en el caso no lineal. El resto de las diferencias son estadísticamente significativas al nivel $\alpha = 0{,}05$ con ambas reglas. Considerando los problemas con 5 atributos relevantes, se sigue que las diferencias encontradas entre las diferentes estrategias son estadísticamente significativas al nivel $\alpha = 0{,}05$; en problemas con 10 atributos relevantes todas las diferencias son significativas excepto entre $\text{VNS}_m$ y AG.

Los tiempos requeridos al usar el clasificador IB1, mostrados en la Tabla 5.16, revelan un comportamiento similar al detectado al usar Naïve Bayes. Destacamos que en este caso el número de diferencias significativas entre $\text{VNS}_m$ y SFFS, $\text{VNS}_m$ y AG y $\text{VNS}_m$ y FSSEBNA son, respectivamente, 4, 20 y 25.

Por último, consideramos el clasificador J48 en los resultados mostrados en la Tabla 5.17. Con SFFS sólo hay 4 diferencias estadísticamente significativas, mientras son estadísticamente diferentes casi todas las que se presentan con las estrategias AG y FSSEBNA.

Tabla 5.15: Comparativa entre la metaheurística $VNS_m$, SFFS, AG y FSSEBNA con el clasificador Naïve Bayes. Tiempo de ejecución en conjuntos de datos artificiales.

| conjuntos de datos | | | | Naïve Bayes | | | |
|---|---|---|---|---|---|---|---|
| | | | | tiempo de ejecución | | | |
| l | r | $\sigma$ | d | $VNS_m$ | SFFS | AG | FSSEBNA |
| 1 | 5 | 0 | 25 | 101,0 ± 68,8 | 26,9 ± 7,3 | 636,2 ± 15,4 $\ll$ | 1281,9 ± 20,5 $\ll$ |
| | | | 50 | 262,9 ± 154,8 | 63,0 ± 15,1 $>$ | 1309,4 ± 24,8 $\ll$ | 2435,9 ± 26,8 $\ll$ |
| | | | 100 | 499,5 ± 136,3 | 165,5 ± 63,8 $>$ | 2703,0 ± 71,3 $\ll$ | 4778,6 ± 31,4 $\ll$ |
| | | | 200 | 1475,8 ± 1061,6 | 428,5 ± 131,2 | 5508,4 ± 174,6 $\ll$ | 9583,4 ± 75,2 $\ll$ |
| | | 0,1 | 25 | 72,5 ± 30,0 | 25,4 ± 8,9 | 623,2 ± 8,8 $\ll$ | 1268,7 ± 28,8 $\ll$ |
| | | | 50 | 179,2 ± 58,7 | 69,5 ± 23,9 | 1306,8 ± 26,5 $\ll$ | 2465,9 ± 33,3 $\ll$ |
| | | | 100 | 837,7 ± 377,8 | 170,2 ± 61,9 | 2824,4 ± 190,8 $\ll$ | 4839,7 ± 40,3 $\ll$ |
| | | | 200 | 1753,5 ± 2210,7 | 365,6 ± 127,8 | 5567,4 ± 204,7 | 9634,0 ± 158,7 $\ll$ |
| | 10 | 0 | 25 | 236,3 ± 65,1 | 35,4 ± 18,4 $>$ | 633,4 ± 17,9 $\ll$ | 1303,7 ± 23,7 $\ll$ |
| | | | 50 | 688,3 ± 373,6 | 79,3 ± 51,7 | 1311,3 ± 33,1 | 2481,9 ± 19,0 $\ll$ |
| | | | 100 | 3157,8 ± 4118,8 | 183,8 ± 96,7 | 2683,8 ± 77,9 | 4822,4 ± 46,5 |
| | | | 200 | 5886,7 ± 4190,5 | 309,5 ± 208,4 | 5488,9 ± 120,9 | 9676,5 ± 91,1 |
| | | 0,1 | 25 | 221,1 ± 87,1 | 31,2 ± 12,2 $\gg$ | 638,8 ± 18,2 $\ll$ | 1327,2 ± 41,5 $\ll$ |
| | | | 50 | 833,8 ± 344,3 | 62,4 ± 29,9 $>$ | 1310,0 ± 31,3 | 2495,1 ± 23,8 $\ll$ |
| | | | 100 | 2365,1 ± 1233,4 | 168,9 ± 67,8 $\gg$ | 3136,5 ± 69,4 | 4865,3 ± 40,5 $\ll$ |
| | | | 200 | 6410,4 ± 5907,9 | 289,7 ± 209,6 | 5507,0 ± 100,0 | 9636,8 ± 56,9 |
| 2 | 5 | 0 | 25 | 111,9 ± 81,7 | 27,3 ± 5,9 $\gg$ | 630,6 ± 14,7 $\ll$ | 1268,1 ± 16,5 $\ll$ |
| | | | 50 | 201,4 ± 86,4 | 67,2 ± 19,7 $>$ | 1286,9 ± 16,2 $\ll$ | 2439,2 ± 30,6 $\ll$ |
| | | | 100 | 696,5 ± 648,0 | 124,2 ± 52,6 | 2671,6 ± 40,6 $\ll$ | 4775,2 ± 57,2 $\ll$ |
| | | | 200 | 2709,3 ± 3087,7 | 328,4 ± 274,1 | 5510,4 ± 130,1 | 9593,2 ± 60,9 $<$ |
| | | 0,1 | 25 | 85,9 ± 50,2 | 25,6 ± 5,0 | 629,7 ± 14,2 $\ll$ | 1262,9 ± 15,7 $\ll$ |
| | | | 50 | 232,8 ± 135,6 | 70,8 ± 18,1 | 1324,0 ± 37,5 $\ll$ | 2439,1 ± 21,1 $\ll$ |
| | | | 100 | 463,6 ± 139,5 | 138,7 ± 50,3 $>$ | 2709,7 ± 70,7 $\ll$ | 7161,8 ± 2552,5 $\ll$ |
| | | | 200 | 2423,2 ± 2630,7 | 268,6 ± 130,3 | 5505,8 ± 97,7 | 9577,0 ± 62,3 $<$ |
| | 10 | 0 | 25 | 199,5 ± 61,1 | 35,8 ± 15,7 $>$ | 630,9 ± 17,0 $\ll$ | 1311,0 ± 20,1 $\ll$ |
| | | | 50 | 656,6 ± 315,5 | 76,4 ± 51,5 | 1282,4 ± 21,9 $<$ | 2478,7 ± 34,9 $\ll$ |
| | | | 100 | 2128,5 ± 1514,6 | 165,2 ± 98,7 $\gg$ | 2657,1 ± 39,1 $<$ | 4819,9 ± 46,4 $\ll$ |
| | | | 200 | 4798,8 ± 3845,5 | 307,3 ± 184,9 | 5448,9 ± 117,9 | 9621,1 ± 62,2 |
| | | 0,1 | 25 | 187,8 ± 67,0 | 27,7 ± 16,6 $>$ | 634,1 ± 14,2 $\ll$ | 1308,7 ± 26,3 $\ll$ |
| | | | 50 | 553,2 ± 380,2 | 60,0 ± 29,9 $>$ | 1290,3 ± 33,8 $\ll$ | 2469,6 ± 19,6 $\ll$ |
| | | | 100 | 2375,2 ± 1497,3 | 116,1 ± 104,0 | 2694,0 ± 96,0 | 4829,9 ± 49,8 |
| | | | 200 | 6467,5 ± 5798,2 | 269,8 ± 198,9 | 5456,3 ± 140,0 | 9588,7 ± 92,3 |

$l$-regla de clasificación; $r$-número de atributos relevantes; $\sigma$-ruido Gaussiano;VNS-Variable Neighbourhood Search; $VNS_m$-VNS híbrida con memoria; SFFS-Sequential Forward Floating Selection; AG-Algoritmo genético; FSSEBNA-Feature Subset Selection by Estimation of Bayesian Network; $>$ y $\gg$ mejora significativa en un 95 y 99 %; $<$ y $\ll$ empeora significativamente.

Tabla 5.16: Comparativa entre la metaheurística $VNS_m$, SFFS, AG y FSSEBNA con el clasificador IB1. Tiempo de ejecución en conjuntos de datos artificiales.

| conjuntos de datos | | | | IB1 | | | |
|---|---|---|---|---|---|---|---|
| | | | | tiempo de ejecución | | | |
| l | r | $\sigma$ | d | $VNS_m$ | SFFS | AG | FSSEBNA |
| 1 | 5 | 0 | 25 | 13,7 ± 4,1 | 5,4 ± 1,5 ≫ | 139,5 ± 17,1 ≪ | 231,8 ± 12,8 ≪ |
| | | | 50 | 46,7 ± 24,8 | 11,5 ± 7,8 | 275,2 ± 34,6 ≪ | 465,8 ± 58,4 ≪ |
| | | | 100 | 173,4 ± 193,6 | 27,3 ± 17,0 | 553,5 ± 79,0 | 866,9 ± 52,8 ≪ |
| | | | 200 | 551,6 ± 755,0 | 39,0 ± 30,3 | 1060,0 ± 117,1 | 1723,2 ± 18,6 |
| | | 0,1 | 25 | 12,1 ± 2,1 | 5,7 ± 1,5 ≫ | 140,3 ± 16,2 ≪ | 240,1 ± 21,4 ≪ |
| | | | 50 | 43,0 ± 21,0 | 12,5 ± 3,8 | 276,2 ± 34,2 ≪ | 449,0 ± 31,7 ≪ |
| | | | 100 | 127,9 ± 69,2 | 28,1 ± 16,9 | 498,6 ± 18,5 ≪ | 910,6 ± 83,2 ≪ |
| | | | 200 | 331,4 ± 129,7 | 55,3 ± 33,9 | 1009,0 ± 36,4 ≪ | 1835,0 ± 151,4 ≪ |
| | 10 | 0 | 25 | 32,7 ± 21,6 | 6,9 ± 2,7 | 141,7 ± 17,8 ≪ | 253,2 ± 24,7 ≪ |
| | | | 50 | 120,8 ± 97,3 | 13,5 ± 8,9 | 263,7 ± 30,5 | 469,9 ± 43,4 < |
| | | | 100 | 616,2 ± 549,9 | 18,1 ± 21,7 | 587,4 ± 66,4 | 905,3 ± 77,4 |
| | | | 200 | 1065,0 ± 1262,1 | 23,0 ± 28,2 | 1107,1 ± 136,2 | 1844,0 ± 153,6 |
| | | 0,1 | 25 | 29,0 ± 19,2 | 4,5 ± 2,5 > | 136,7 ± 12,7 ≪ | 252,9 ± 22,7 ≪ |
| | | | 50 | 97,3 ± 36,8 | 9,8 ± 5,5 > | 273,3 ± 38,0 ≪ | 462,1 ± 37,2 ≪ |
| | | | 100 | 529,2 ± 274,2 | 19,7 ± 20,4 ≫ | 551,6 ± 75,8 | 887,1 ± 65,8 < |
| | | | 200 | 1480,1 ± 1179,4 | 21,5 ± 18,5 | 1108,4 ± 143,3 | 1838,5 ± 157,2 |
| 2 | 5 | 0 | 25 | 12,0 ± 3,8 | 6,2 ± 1,6 | 141,5 ± 17,3 ≪ | 227,6 ± 3,4 ≪ |
| | | | 50 | 37,3 ± 31,4 | 12,1 ± 3,7 | 250,0 ± 13,6 ≪ | 462,8 ± 37,6 ≪ |
| | | | 100 | 119,6 ± 81,5 | 25,3 ± 11,9 | 521,4 ± 55,4 ≪ | 904,5 ± 73,6 ≪ |
| | | | 200 | 253,0 ± 241,1 | 53,6 ± 24,0 | 1069,4 ± 106,0 ≪ | 1866,6 ± 151,4 ≪ |
| | | 0,1 | 25 | 12,6 ± 5,9 | 5,1 ± 1,5 | 145,4 ± 21,3 ≪ | 234,6 ± 20,1 ≪ |
| | | | 50 | 34,7 ± 19,7 | 12,4 ± 4,8 | 277,3 ± 38,6 ≪ | 456,0 ± 38,8 ≪ |
| | | | 100 | 68,0 ± 36,8 | 22,2 ± 5,6 | 494,0 ± 21,6 ≪ | 874,2 ± 48,0 ≪ |
| | | | 200 | 175,8 ± 150,1 | 50,2 ± 21,0 | 1056,7 ± 124,0 ≪ | 1743,5 ± 97,2 ≪ |
| | 10 | 0 | 25 | 40,3 ± 22,3 | 2,7 ± 2,1 | 144,0 ± 18,7 ≪ | 236,5 ± 14,9 ≪ |
| | | | 50 | 102,7 ± 89,7 | 7,0 ± 8,0 | 265,0 ± 30,2 | 442,6 ± 27,3 ≪ |
| | | | 100 | 353,7 ± 219,7 | 17,6 ± 22,0 > | 519,0 ± 55,6 | 897,5 ± 76,3 ≪ |
| | | | 200 | 965,1 ± 773,5 | 33,1 ± 34,0 | 1020,7 ± 40,2 | 1808,8 ± 140,3 |
| | | 0,1 | 25 | 34,3 ± 16,9 | 5,2 ± 3,6 | 135,0 ± 12,5 ≪ | 236,2 ± 14,9 ≪ |
| | | | 50 | 51,5 ± 34,1 | 6,5 ± 7,1 | 252,4 ± 21,6 ≪ | 459,0 ± 40,5 ≪ |
| | | | 100 | 755,9 ± 677,7 | 21,5 ± 20,8 | 555,6 ± 72,4 | 917,6 ± 117,4 |
| | | | 200 | 903,1 ± 691,0 | 32,5 ± 34,0 | 1137,2 ± 139,2 | 1744,7 ± 96,1 |

$l$-regla de clasificación; $r$-número de atributos relevantes; $\sigma$-ruido Gaussiano;VNS-Variable Neighbourhood Search; $VNS_m$-VNS híbrida con memoria; SFFS-Sequential Forward Floating Selection; AG-Algoritmo genético; FSSEBNA-Feature Subset Selection by Estimation of Bayesian Network; > y ≫ mejora significativa en un 95 y 99 %; < y ≪ empeora significativamente.

Tabla 5.17: Comparativa entre la metaheurística $VNS_m$, SFFS, AG y FSSEBNA con el clasificador J48. Tiempo de ejecución en conjuntos de datos artificiales.

| conjuntos de datos | | | | J48 | | | |
|---|---|---|---|---|---|---|---|
| | | | | tiempo de ejecución | | | |
| l | r | $\sigma$ | d | $VNS_m$ | SFFS | AG | FSSEBNA |
| 1 | 5 | 0 | 25 | 5,9 ± 2,3 | 2,6 ± 1,1 | 67,8 ± 7,9 ≪ | 104,5 ± 4,1 ≪ |
| | | | 50 | 28,8 ± 11,2 | 6,0 ± 2,1 | 131,7 ± 10,7 ≪ | 203,7 ± 6,8 ≪ |
| | | | 100 | 51,9 ± 27,7 | 15,3 ± 5,8 | 272,3 ± 11,6 ≪ | 397,9 ± 15,3 ≪ |
| | | | 200 | 138,5 ± 62,8 | 36,9 ± 14,4 | 560,6 ± 23,8 ≪ | 831,7 ± 32,7 ≪ |
| | | 0,1 | 25 | 6,0 ± 6,3 | 2,3 ± 0,8 | 67,3 ± 7,2 ≪ | 105,8 ± 3,7 ≪ |
| | | | 50 | 24,9 ± 19,1 | 5,8 ± 1,2 > | 132,5 ± 8,7 ≪ | 204,2 ± 6,5 ≪ |
| | | | 100 | 78,1 ± 41,5 | 12,1 ± 3,0 | 273,9 ± 13,4 ≪ | 398,8 ± 16,3 ≪ |
| | | | 200 | 220,3 ± 167,2 | 30,4 ± 10,7 | 566,0 ± 20,9 < | 832,5 ± 23,5 ≪ |
| | 10 | 0 | 25 | 14,7 ± 10,0 | 2,7 ± 1,5 > | 74,6 ± 7,6 ≪ | 116,7 ± 3,5 ≪ |
| | | | 50 | 32,9 ± 26,0 | 5,3 ± 3,3 | 143,9 ± 11,5 ≪ | 225,4 ± 9,9 ≪ |
| | | | 100 | 170,3 ± 164,8 | 12,9 ± 8,0 | 297,4 ± 7,4 | 441,2 ± 14,3 |
| | | | 200 | 391,7 ± 174,8 | 46,3 ± 17,6 | 599,1 ± 15,4 | 908,0 ± 37,0 ≪ |
| | | 0,1 | 25 | 14,3 ± 11,3 | 2,9 ± 1,1 | 74,1 ± 10,2 ≪ | 114,4 ± 5,6 ≪ |
| | | | 50 | 43,0 ± 32,1 | 6,6 ± 4,5 > | 144,8 ± 11,3 ≪ | 224,4 ± 9,2 ≪ |
| | | | 100 | 187,9 ± 127,1 | 17,3 ± 9,6 | 293,8 ± 13,9 | 436,0 ± 16,5 < |
| | | | 200 | 435,4 ± 286,4 | 49,0 ± 32,9 | 602,3 ± 21,3 | 905,7 ± 44,4 |
| 2 | 5 | 0 | 25 | 7,7 ± 6,4 | 2,9 ± 0,8 | 65,8 ± 8,9 ≪ | 99,1 ± 4,4 ≪ |
| | | | 50 | 19,3 ± 12,6 | 6,4 ± 1,9 | 128,6 ± 11,2 ≪ | 193,5 ± 7,4 ≪ |
| | | | 100 | 65,0 ± 71,5 | 15,6 ± 4,6 | 268,2 ± 12,3 ≪ | 379,5 ± 8,2 ≪ |
| | | | 200 | 164,0 ± 134,7 | 40,0 ± 15,5 | 542,7 ± 20,7 ≪ | 777,2 ± 28,9 ≪ |
| | | 0,1 | 25 | 5,7 ± 4,4 | 2,3 ± 1,3 | 66,2 ± 8,0 ≪ | 98,8 ± 3,8 ≪ |
| | | | 50 | 13,0 ± 5,5 | 5,1 ± 2,1 | 127,1 ± 10,0 ≪ | 190,6 ± 7,1 ≪ |
| | | | 100 | 58,6 ± 45,8 | 12,9 ± 3,9 | 267,0 ± 13,7 ≪ | 375,2 ± 13,4 ≪ |
| | | | 200 | 144,3 ± 131,0 | 33,6 ± 12,6 | 547,2 ± 16,8 < | 785,0 ± 22,7 ≪ |
| | 10 | 0 | 25 | 14,8 ± 12,7 | 3,6 ± 1,1 | 74,0 ± 9,5 ≪ | 116,7 ± 5,5 ≪ |
| | | | 50 | 42,8 ± 22,1 | 7,9 ± 3,2 | 139,3 ± 10,1 ≪ | 228,5 ± 11,5 ≪ |
| | | | 100 | 188,1 ± 137,9 | 20,9 ± 5,7 | 290,6 ± 15,4 | 448,6 ± 22,6 < |
| | | | 200 | 397,8 ± 254,5 | 47,3 ± 19,7 | 599,2 ± 17,0 | 936,5 ± 45,9 |
| | | 0,1 | 25 | 13,4 ± 9,5 | 2,4 ± 1,2 ≫ | 73,3 ± 9,8 ≪ | 115,7 ± 3,7 ≪ |
| | | | 50 | 41,1 ± 35,1 | 7,0 ± 4,7 | 141,6 ± 10,6 ≪ | 224,2 ± 8,3 ≪ |
| | | | 100 | 116,1 ± 105,0 | 17,0 ± 8,7 | 290,0 ± 10,2 < | 444,2 ± 25,9 ≪ |
| | | | 200 | 344,2 ± 194,2 | 55,6 ± 40,7 | 593,4 ± 24,8 | 930,5 ± 54,5 < |

$l$-regla de clasificación; $r$-número de atributos relevantes; $\sigma$-ruido Gaussiano;VNS-Variable Neighbourhood Search; $VNS_m$-VNS híbrida con memoria; SFFS-Sequential Forward Floating Selection; AG-Algoritmo genético; FSSEBNA-Feature Subset Selection by Estimation of Bayesian Network; > y ≫ mejora significativa en un 95 y 99 %; < y ≪ empeora significativamente.

### Problemas reales

Los resultados en todas las tablas tienen una lectura similar a los resultados analizados en los conjuntos artificiales. El orden de las estrategias que emplean, en promedio, de menos a más tiempo son SFFS, $\text{VNS}_m$, AG y FSSEBNA.

En todos los clasificadores, las diferencias entre $\text{VNS}_m$ y SFFS son estadísticamente significativas para unos pocos casos repartidos entre los distintos tipos de conjuntos que consideramos. Con el resto de estrategias, las diferencias estadísticamente significativas son mayores. En los conjuntos de mayor tamaño no hay diferencias significativas empleando el clasificador Naïve Bayes.

Tabla 5.18: Comparativa entre $VNS_m$, SFFS, AG y FSSEBNA con el clasificador Naïve Bayes. Tiempo de ejecución en conjuntos de datos reales.

| cdd | Naïve Bayes | | | |
|---|---|---|---|---|
| | tiempos de ejecución | | | |
| | $VNS_m$ | SFFS | AG | FSSEBNA |
| pdi | 8,4 ± 2,1 | 5,3 ± 1,5 > | 26,8 ± 0,5 ≪ | 26,8 ± 0,5 ≪ |
| gla | 2,2 ± 0,7 | 1,3 ± 0,4 ≫ | 12,6 ± 1,0 ≪ | 11,9 ± 0,4 ≪ |
| bca | 0,5 ± 0,1 | 0,4 ± 0,1 | 1,9 ± 0,4 ≪ | 1,6 ± 0,0 ≪ |
| bwi | 3,5 ± 0,9 | 2,0 ± 0,4 | 16,4 ± 0,4 ≪ | 15,7 ± 0,3 ≪ |
| win | 5,2 ± 2,3 | 1,9 ± 0,4 | 46,6 ± 3,2 ≪ | 68,1 ± 2,0 ≪ |
| hst | 5,5 ± 2,1 | 1,5 ± 0,6 | 42,0 ± 2,8 ≪ | 59,6 ± 2,1 ≪ |
| hhu | 3,1 ± 1,8 | 1,1 ± 0,4 | 35,0 ± 2,7 ≪ | 46,4 ± 3,2 ≪ |
| hcl | 4,4 ± 1,9 | 1,8 ± 0,8 > | 45,3 ± 3,7 ≪ | 61,1 ± 3,0 ≪ |
| vow | 152,5 ± 42,7 | 70,0 ± 16,5 | 1179,3 ± 25,4 ≪ | 1670,4 ± 59,8 ≪ |
| cra | 10,8 ± 9,8 | 3,9 ± 2,6 | 125,9 ± 5,3 ≪ | 202,2 ± 8,2 ≪ |
| ptu | 2,5 ± 0,5 | 1,5 ± 0,2 | 21,1 ± 8,8 | 23,3 ± 0,8 ≪ |
| vot | 0,7 ± 0,2 | 0,6 ± 0,1 | 9,0 ± 2,0 ≪ | 12,1 ± 0,3 ≪ |
| lym | 1,5 ± 1,1 | 0,7 ± 0,1 | 12,2 ± 3,8 < | 16,1 ± 1,5 ≪ |
| veh | 117,9 ± 58,6 | 27,6 ± 15,7 | 730,1 ± 26,1 ≪ | 1484,5 ± 49,9 ≪ |
| hep | 1,3 ± 0,6 | 0,7 ± 0,1 | 17,8 ± 3,7 ≪ | 30,3 ± 2,1 ≪ |
| hco | 4,4 ± 1,9 | 1,5 ± 0,5 | 45,6 ± 3,2 ≪ | 78,1 ± 3,2 ≪ |
| aut | 8,8 ± 4,3 | 3,5 ± 1,5 | 84,0 ± 8,4 ≪ | 155,2 ± 8,5 ≪ |
| abp | 43,0 ± 28,0 | 29,3 ± 10,3 | 762,1 ± 23,6 ≪ | 1392,8 ± 64,1 ≪ |
| ion | 49,3 ± 18,8 | 13,0 ± 2,5 > | 296,6 ± 9,1 ≪ | 551,5 ± 18,0 ≪ |
| soy | 30,1 ± 10,5 | 8,0 ± 1,2 > | 58,8 ± 18,3 | 75,0 ± 1,8 ≪ |
| kvk | 11,9 ± 0,6 | 10,9 ± 0,2 | 104,0 ± 5,8 ≪ | 160,6 ± 2,1 ≪ |
| ann | 45,3 ± 41,5 | 8,3 ± 3,5 | 125,7 ± 8,4 | 224,1 ± 6,5 ≪ |
| lca | 0,9 ± 0,1 | 0,8 ± 0,1 | 6,8 ± 3,7 | 6,7 ± 0,1 ≪ |
| dnp | 1,7 ± 1,0 | 1,0 ± 0,2 | 12,8 ± 8,6 | 9,7 ± 0,2 ≪ |
| son | 147,8 ± 182,4 | 8,9 ± 4,8 | 303,9 ± 8,3 | 577,2 ± 6,2 < |
| dna | 84,2 ± 41,7 | 32,8 ± 5,5 | 143,0 ± 12,7 | 221,3 ± 4,2 < |
| spl | 139,8 ± 70,2 | 34,0 ± 7,6 | 210,1 ± 128,9 | 226,6 ± 3,4 |
| aud | 8,9 ± 3,5 | 3,5 ± 0,8 | 46,9 ± 22,9 | 49,7 ± 1,9 ≪ |
| rat | 171,1 ± 166,8 | 14,5 ± 6,6 | 334,3 ± 17,6 | 634,5 ± 25,5 < |
| arr | 3211,0 ± 2290,2 | 339,8 ± 147,3 | 3848,9 ± 230,8 | 6540,6 ± 54,4 |

VNS-Variable Neighbourhood Search; $VNS_m$-VNS híbrida con memoria; SFFS-Sequential Forward Floating Selection; AG-Algoritmo genético; FSSEBNA-Feature Subset Selection with EBNA;> y ≫ mejora significativa en un 95 y 99 %; < y ≪ empeora significativamente.

## 5.5 Conclusiones

En el presente trabajo se han propuesto y analizado dos estrategias de Búsqueda por Entorno Variable para el problema de la selección de atributos. Una de las estrategias es un híbrido entre VNS y Búsqueda Tabú. De la experiencia computacional podemos destacar las siguientes conclusiones:

- Del análisis de la aportación del VNS al clasificador base concluimos que, en los problemas artificiales considerados, VNS permite seleccionar un subconjunto de atributos con el que se mejora los resultados obtenidos con el clasificador sin selección de atributos. En los problemas reales, la eficacia es similar, aumentando así la eficiencia del clasificador.

- Comparando VNS y SFFS (estrategia usada como búsqueda local en VNS) se concluye que VNS mejora la selección del subconjunto de atributos útil para la tarea de clasificación. Esto se refleja, en los conjuntos artificiales, en la mejora en el porcentaje de aciertos obtenido. Casi todas diferencias son estadísticamente significativas y favorables a VNS. Además, en los conjuntos reales, también se obtuvieron mejoras significativas por parte del VNS usando los clasificadores Naïve Bayes e IB1. No obstante, habría que aumentar la experiencia para poder afianzar estas conclusiones, ya que VNS selecciona subconjuntos de tamaño mayor o igual al seleccionado por SFFS. Esto sucede en todas las pruebas realizadas.

- Al comparar VNS y $\text{VNS}_m$ se sigue que ambas estrategias presentan un rendimiento similar en cuanto a porcentaje de acierto, ya que, de todos los casos estudiados, sólo en los problemas artificiales usando Naïve Bayes se encontraron diferencias estadísticamente significativas. En tamaño de los subconjuntos de atributos, también el rendimiento es similar, pues sólo para el caso de los problemas artificiales con Naïve Bayes, se encuentran diferencias significativas claras; usando el clasificador J48, se encuentran diferencias significativas en los problemas artificiales cuando se analizan conjuntamente los resultados sobre

Tabla 5.19: Comparativa entre $VNS_m$, SFFS, AG y FSSEBNA con el clasificador IB1. Tiempo de ejecución en conjuntos de datos reales.

| cdd | IB1 | | | |
|---|---|---|---|---|
| | tiempos de ejecución | | | |
| | $VNS_m$ | SFFS | AG | FSSEBNA |
| pdi | 4,0 ± 1,4 | 2,0 ± 0,7 | 15,3 ± 2,5 ≪ | 14,5 ± 1,2 ≪ |
| gla | 1,0 ± 0,2 | 0,7 ± 0,1 | 4,4 ± 1,0 < | 3,5 ± 0,3 ≪ |
| bca | 0,7 ± 0,3 | 0,6 ± 0,1 | 5,3 ± 1,2 < | 4,2 ± 0,1 ≪ |
| bwi | 5,1 ± 1,7 | 2,6 ± 0,8 | 26,3 ± 4,0 ≪ | 23,4 ± 1,9 ≪ |
| win | 1,4 ± 0,6 | 0,7 ± 0,1 | 13,7 ± 3,7 < | 14,2 ± 0,6 ≪ |
| hst | 2,5 ± 1,2 | 0,8 ± 0,3 | 22,7 ± 4,7 ≪ | 28,6 ± 1,2 ≪ |
| hhu | 1,9 ± 1,0 | 0,9 ± 0,3 | 24,5 ± 5,6 < | 27,3 ± 0,9 ≪ |
| hcl | 2,8 ± 1,2 | 0,9 ± 0,4 | 26,4 ± 5,6 < | 31,7 ± 1,7 ≪ |
| vow | 25,3 ± 6,3 | 17,9 ± 3,4 | 245,1 ± 24,8 ≪ | 297,6 ± 14,7 ≪ |
| cra | 7,1 ± 5,8 | 3,1 ± 2,2 | 145,8 ± 17,1 ≪ | 218,1 ± 11,0 ≪ |
| ptu | 8,6 ± 2,4 | 1,7 ± 0,9 > | 42,7 ± 10,7 < | 63,7 ± 1,1 ≪ |
| vot | 2,4 ± 1,2 | 1,4 ± 0,7 | 53,1 ± 8,8 ≪ | 78,9 ± 1,8 ≪ |
| lym | 2,0 ± 0,6 | 0,8 ± 0,3 ≫ | 14,1 ± 4,9 | 17,5 ± 0,4 ≪ |
| veh | 43,0 ± 22,7 | 14,8 ± 3,0 | 291,4 ± 30,4 ≪ | 499,9 ± 32,8 ≪ |
| hep | 1,3 ± 0,6 | 0,6 ± 0,2 | 14,6 ± 4,6 < | 19,7 ± 0,4 ≪ |
| hco | 7,2 ± 5,5 | 2,6 ± 1,0 > | 61,0 ± 9,2 ≪ | 97,8 ± 1,3 ≪ |
| aut | 3,7 ± 5,0 | 1,3 ± 0,3 | 28,9 ± 6,1 ≪ | 43,6 ± 1,9 ≪ |
| abp | 711,0 ± 1500,0 | 104,6 ± 99,2 | 6267,3 ± 179,3 ≪ | 12074,1 ± 122,9 ≪ |
| ion | 10,4 ± 5,3 | 3,2 ± 1,1 | 91,6 ± 9,9 ≪ | 154,5 ± 12,2 ≪ |
| soy | 237,1 ± 67,5 | 51,0 ± 13,4 ≫ | 267,5 ± 33,1 | 442,2 ± 4,5 ≪ |
| kvk | 1733,5 ± 749,5 | 368,8 ± 242,0 | 5679,8 ± 161,8 ≪ | 9644,6 ± 218,5 ≪ |
| ann | 111,2 ± 31,4 | 51,1 ± 11,3 > | 341,6 ± 33,6 ≪ | 573,1 ± 11,3 ≪ |
| lca | 1,2 ± 0,2 | 0,8 ± 0,1 | 9,0 ± 6,1 | 7,0 ± 0,1 ≪ |
| dnp | 5,4 ± 3,5 | 1,0 ± 0,3 | 18,9 ± 6,7 | 23,0 ± 0,2 ≪ |
| son | 33,9 ± 23,8 | 4,6 ± 2,3 | 63,0 ± 9,8 < | 104,7 ± 8,8 ≪ |
| dna | 1313,6 ± 347,9 | 512,0 ± 139,0 > | 8504,0 ± 114,4 ≪ | 15425,3 ± 127,6 ≪ |
| spl | 1314,6 ± 687,9 | 409,0 ± 118,9 | 9083,4 ± 217,2 ≪ | 15667,5 ± 97,7 ≪ |
| aud | 37,0 ± 20,7 | 8,3 ± 3,0 > | 71,7 ± 17,6 | 95,0 ± 0,7 ≪ |
| rat | 14,9 ± 8,9 | 6,2 ± 3,4 | 122,8 ± 20,1 ≪ | 214,8 ± 11,0 ≪ |
| arr | 2964,3 ± 915,2 | 187,5 ± 92,3 > | 1167,0 ± 157,6 | 1999,2 ± 99,2 |

VNS-Variable Neighbourhood Search; $VNS_m$-VNS híbrida con memoria; SFFS-Sequential Forward Floating Selection; AG-Algoritmo genético; FSSEBNA-Feature Subset Selection with EBNA;> y ≫ mejora significativa en un 95 y 99 %; < y ≪ empeora significativamente.

Tabla 5.20: Comparativa entre $VNS_m$, SFFS, AG y FSSEBNA con el clasificador J48. Tiempo de ejecución en conjuntos de datos reales.

| cdd | J48 | | | |
|---|---|---|---|---|
| | tiempos de ejecución | | | |
| | $VNS_m$ | SFFS | AG | FSSEBNA |
| pdi | 0,8 ± 0,2 | 0,6 ± 0,1 | 4,2 ± 0,3 ≪ | 3,9 ± 0,2 ≪ |
| gla | 1,2 ± 0,2 | 0,8 ± 0,1 > | 5,7 ± 1,2 < | 4,6 ± 0,2 ≪ |
| bca | 0,5 ± 0,1 | 0,5 ± 0,1 | 2,8 ± 0,6 ≪ | 2,3 ± 0,1 ≪ |
| bwi | 1,1 ± 0,5 | 0,8 ± 0,1 | 5,8 ± 0,6 ≪ | 5,0 ± 0,2 ≪ |
| win | 0,7 ± 0,1 | 0,7 ± 0,1 | 11,0 ± 2,7 < | 11,9 ± 0,7 ≪ |
| hst | 0,9 ± 0,5 | 0,5 ± 0,1 | 13,9 ± 2,9 ≪ | 17,1 ± 1,0 ≪ |
| hhu | 0,6 ± 0,1 | 0,5 ± 0,1 | 11,6 ± 2,5 ≪ | 12,7 ± 1,1 ≪ |
| hcl | 1,0 ± 0,6 | 0,6 ± 0,2 | 14,5 ± 3,6 < | 16,2 ± 1,0 ≪ |
| vow | 29,8 ± 10,0 | 13,3 ± 4,5 | 189,9 ± 15,8 ≪ | 230,8 ± 16,6 ≪ |
| cra | 1,1 ± 1,3 | 0,7 ± 0,2 | 36,3 ± 6,4 ≪ | 50,2 ± 3,0 ≪ |
| ptu | 6,0 ± 2,1 | 1,7 ± 0,5 | 37,2 ± 7,7 < | 57,9 ± 1,5 ≪ |
| vot | 0,6 ± 0,0 | 0,5 ± 0,0 ≫ | 13,4 ± 3,0 ≪ | 17,9 ± 1,0 ≪ |
| lym | 0,7 ± 0,2 | 0,6 ± 0,1 | 10,3 ± 3,8 | 12,2 ± 0,8 ≪ |
| veh | 14,4 ± 7,1 | 4,9 ± 1,1 | 120,0 ± 11,3 ≪ | 199,1 ± 5,7 ≪ |
| hep | 0,6 ± 0,2 | 0,6 ± 0,1 | 10,4 ± 3,6 < | 13,6 ± 0,7 ≪ |
| hco | 1,1 ± 0,2 | 1,0 ± 0,1 | 22,8 ± 3,9 ≪ | 32,1 ± 1,7 ≪ |
| aut | 2,3 ± 1,4 | 1,3 ± 0,3 | 29,5 ± 8,6 < | 40,5 ± 2,1 ≪ |
| abp | 11,0 ± 5,6 | 9,5 ± 3,1 | 280,4 ± 12,1 ≪ | 462,3 ± 22,1 ≪ |
| ion | 3,8 ± 1,7 | 1,9 ± 0,5 | 53,4 ± 5,4 ≪ | 88,5 ± 5,1 ≪ |
| soy | 90,7 ± 43,9 | 25,9 ± 5,7 | 120,6 ± 16,3 | 177,6 ± 4,1 |
| kvk | 56,4 ± 19,3 | 12,3 ± 1,4 > | 296,2 ± 12,0 ≪ | 413,4 ± 8,7 ≪ |
| ann | 13,6 ± 11,8 | 6,5 ± 2,6 | 88,8 ± 9,6 ≪ | 152,9 ± 5,2 ≪ |
| lca | 0,7 ± 0,1 | 0,7 ± 0,1 | 9,6 ± 5,3 | 8,4 ± 0,1 ≪ |
| dnp | 0,9 ± 0,2 | 0,8 ± 0,1 | 12,0 ± 4,8 | 13,5 ± 0,5 ≪ |
| son | 7,0 ± 3,8 | 3,4 ± 1,5 | 50,6 ± 6,4 ≪ | 80,8 ± 2,8 ≪ |
| dna | 98,4 ± 47,9 | 45,4 ± 7,5 | 292,8 ± 25,3 < | 420,7 ± 3,2 ≪ |
| spl | 79,2 ± 29,4 | 44,8 ± 7,3 ≫ | 515,9 ± 108,2 ≪ | 693,5 ± 43,0 ≪ |
| aud | 8,3 ± 2,5 | 5,7 ± 1,6 | 52,1 ± 12,8 < | 68,0 ± 1,1 ≪ |
| rat | 12,8 ± 6,0 | 4,8 ± 2,0 | 91,0 ± 9,3 ≪ | 167,7 ± 6,2 ≪ |
| arr | 1162,2 ± 796,6 | 79,0 ± 23,2 | 1063,2 ± 40,1 | 1842,6 ± 82,1 |

VNS-Variable Neighbourhood Search; $VNS_m$-VNS híbrida con memoria; SFFS-Sequential Forward Floating Selection; AG-Algoritmo genético; FSSEBNA-Feature Subset Selection with EBNA;> y ≫ mejora significativa en un 95 y 99 %; < y ≪ empeora significativamente.

todas las bases de datos, pero no cuando se hace de forma detallada. Por ello, concluimos que en términos generales las diferencias en el porcentaje de aciertos y número de atributos no muestran evidencias claras que indiquen que el rendimiento sea distintos aún cuando en casos particulares existen diferencias estadísticamente significativas. Las diferencias entre ambas estrategias se da en la convergencia. La versión híbrida converge antes.

- Comparando los resultados de $\text{VNS}_m$ con los obtenidos por AG y FSSEBNA, concluimos que $\text{VNS}_m$ es una estrategia competitiva que obtiene resultados similares o mejores que los de las otras estrategias, usando un subconjunto de atributos de menor tamaño. En todos los casos, $\text{VNS}_m$ es la estrategia que reduce en mayor medida, mientras que su porcentaje de aciertos, dependiendo del conjunto de datos, mejora o no a las otras dos estrategias, si bien estas diferencias no son estadísticamente significativas.

- Estudiando el tiempo de ejecución y comparándolo con el de las otras estrategias, concluimos que, como era de esperar, $\text{VNS}_m$ es menos eficiente que SFFS, pues hace uso de esta estrategia como búsqueda local. Por otro lado, $\text{VNS}_m$ es más eficiente que AG y FSSEBNA. Por ello, podemos decir que el $\text{VNS}_m$ tiene mantiene un compromiso entre la eficiencia y eficacia que es competitivo frente al presentado por el resto de estrategias analizadas en el capítulo.

En el futuro inmediato, se ampliará la experiencia computacional y se estudiará en mayor profundidad las diferencias existentes entre la metaheurística propuesta y otras presentes en la literatura.

# Capítulo 6

# Conclusiones y trabajos futuros

## 6.1 Conclusiones

En este trabajo se presenta, en primer lugar, una propuesta de implementación de la metaheurística Búsqueda Dispersa para resolver el problema de selección de atributos. Diseñamos dos métodos de combinación de soluciones: la Combinación Voraz y la Combinación Voraz Reducida. Con el propósito de hacer uso de ambas combinaciones de forma simultánea e incrementar así la capacidad de exploración de la estrategia, desarrollamos una paralelización de la Búsqueda Dispersa. Dicha paralelización consiste en ejecutar cada método de combinación de soluciones en un procesador distinto.

Posteriormente se proponen dos estrategias de la Búsqueda por Entorno Variable, una de ellas híbrida con la Búsqueda Tabú para el problema de la selección de atributos. Además, se compara con la estrategia SFFS, considerada en la búsqueda local. De las experiencia computacional podemos destacar las siguientes conclusiones:

- Los problemas artificiales permiten estudiar en mejor medida el rendimiento de las distintas estrategias que se estén comparando. El conocimiento de las dificultades del problema, así como la variación gradual de dichas dificultades, permite extraer conclusiones sobre la idoneidad

de la aplicación de cada estrategia en problemas de características similares.

- La utilización de los problemas reales del repositorio de la UCI dificultan que se obtengan conclusiones claras sobre el rendimiento de las estrategias debido al desconocimiento de las dificultades de estos problemas.
- Ninguna estrategia es superior a las otras en los problemas reales, con lo que sería conveniente centrarse en un problema real concreto.
- Tanto la Búsqueda Dispersa como las Búsquedas por Entorno Variable presentadas en este trabajo demuestran que son estrategias competitivas en los distintos problemas estudiados.

Las conclusiones específicas de la Búsqueda Dispersa son:

- La Búsqueda Dispersa mostró ser una estrategia robusta frente a los cambios en los valores de los parámetros de entrada en los conjuntos de datos reales sobre los que se probó.
- Las distintas versiones secuenciales presentan un comportamiento similar y la versión paralela encuentra, en los conjuntos probados de mayor tamaño, subconjuntos de menor tamaño con una tasa de error similar al encontrado por las versiones secuenciales.
- De la comparativa entre la Búsqueda Dispersa, el AG y FSSEBNA concluimos que la Búsqueda Dispersa es una estrategia competitiva. Sin embargo es preferible usarlo con el Naïve Bayes o J48.
- La Búsqueda Dispersa es una estrategia que emplea un tiempo pequeño, con lo que puede aplicarse a problemas de mayor tamaño que el AG y FSSEBNA.

Las conclusiones específicas de la Búsqueda por Entorno variable son:

- En los conjuntos de datos artificiales el VNS mostró ser una estrategia capaz de seleccionar subconjuntos de atributos útiles para la tarea de clasificación, mejorando la tasa de acierto del clasificador base.

- La introducción de memoria permite que la estrategia VNS converja antes sin perjuicio de la eficacia de la tarea.
- El $\text{VNS}_m$ mantiene un compromiso entre la eficiencia y eficacia y es competitivo frente a las otras estrategias presentadas.

## 6.2 Trabajos futuros

De esta forma, las líneas a seguir son las siguientes:

- Como trabajo futuro común a las estrategias presentadas en este trabajo tenemos el ampliar la experiencia computacional con problemas artificiales, añadiendo nuevas complicaciones como, por ejemplo, la adición de atributos redundantes, pesos asociados a los atributos relevantes, etc.
- Aplicar las estrategias Búsqueda Dispersa y Búsqueda por Entorno Variable a un problema real en el que a día de hoy se está trabajando.
- Estudiar posibles mejoras en ambas estrategias teniendo e cuenta los últimos avances en el área.
- Mejorar la experiencia computacional de la Búsqueda Dispersa para estudiar los valores de los parámetros más adecuados. Esta mejora consistiría tanto en aumentar los problemas considerados así como los clasificadores a considerar.

# Apéndice A

# Definiciones

A lo largo del presente trabajo se considerarán diversos términos generales que pasamos a definir. En la tesis doctoral de Ron Kohavi [83] puede verse algunas de estas definiciones de ámbito general en el aprendizaje automático:

**Definición A.1** ***Atributo (Variable, Característica, Campo)****: Descripción parcial de una instancia.*

*Los valores que adopte dependerá del dominio definido por el tipo de atributo.*

**Definición A.2** ***Clasificador****: Correspondencia entre instancias no etiquetadas y las clases (discreta). Función que asigna a una instancia una clase.*

**Definición A.3** ***Conjunto de Datos****: Conjunto de ejemplos sobre los cuales se llevará a cabo la tarea.*

**Definición A.4** ***Dimensión****: Número de atributos que describen una instancia.*

**Definición A.5** ***Esquema****: Descripción de los atributos de un conjunto de datos y sus propiedades.*

**Definición A.6** ***Inductor o Algoritmo de Inducción****: Algoritmo que, dado un conjunto de instancias, produce un modelo que generaliza más allá de*

*estas instancias. En nuestro caso dicho modelo corresponderá a un clasificador.*

**Definición A.7** ***Instancia (Ejemplo, Caso, Objeto)****: Un objeto del universo a partir del cual un modelo será aprendido, o sobre el cual un modelo será aplicado. Las instancias se describen, en la mayoría de las tareas de aprendizaje automátivo, por un vector de atributos.*

Antes de continuar conviene aclarar los tipos de atributos con los que nos podemos encontrar ya que según el tipo deberá recibir tratamientos diferentes. Dorian [117] distingue entre medidas escalares y medidas no escalares. Los tipos de medidas escalares que podemos encontrar son:

**Nominal.** Consiste simplemente en una etiqueta, con lo que ofece una cantidad de información pobre.

**Categórica.** También es una etiqueta; sin embargo cada etiqueta identifica grupos de individuos y no uno solo. Aunque también puede ser un número no deja de ser una mera etiqueta.

**Ordinal.** Añade un cierto tipo de orden según el valor de la etiqueta. La cantidad de información que ofrece es por tanto superior con respecto a los dos casos anteriores.

**Intervalo.** Es un dato cuantitativo en que el origen de la escala es arbitrario, de modo que a misma diferencia de valores de un intervalo en dos unidades de medida distintas puede llevar a distintos cocientes entre el valor máximo y mínimo de entre ambos intervalos.

**Indice.** Es similar al caso anterior pero el origen no es arbitrario sino absoluto.

Las medidas no escalares son aquellas que junto al valor de la medida ha de informarse de la escala contra la cual fue realizada.

A efectos prácticos suele diferenciarse entre variables categóricas y variables continuas. Las nominales están limitadas a un determinado grupo finito

o numerable de valores discretos entre los que no se puede establecer una relación de orden mientras que las continuas son aquellas en las que sí se puede establecer una relación de orden.

En caso de que el valor de una variable de una instancia determinada sea desconocida se dice que tiene el valor perdido.

Algunos de los motivos más comunes del desconocimiento de un valor son la imposibilidad de realizar la medida correspondiente y el incorrecto funcionamiento del aparato de medida. A esta clase de datos se le suelen dar ciertos tratamientos que están fuera de los propósitos de este capítulo.

**Definición A.8** ***Matriz de confusión****: Matriz que muestra información relativa a las predicciones realizadas por el clasificador. El tamaño suele ser de $c \times c$, donde $c$ es el número de clases o etiquetas existentes. Da una información más detallada sobre la clasificación de las instancias pues informa no sólo del número de instancias mal clasificadas sino que detalla el número de instancias de cada clase que han sido clasificadas en cada una de las clases existentes.*

**Definición A.9** ***Modelo****: Estructura y correspondiente interpretación que resume o parcialmente resume el conjunto de datos para una descripción o predicción.*

**Definición A.10** ***Patrón****: Es una característica local de un conjunto de ejemplos, presente parcial o completamente en los atributos o ejemplos (o incluso en ambos).*

**Definición A.11** ***Precisión****: El índice de predicciones correctas (o incorrectas) realizadas por el modelo sobre un conjunto de instancias.*

# Apéndice B

# Conjuntos de datos

Los experimentos realizados se llevaron a cabo sobre conjuntos de datos artificiales, en el que las dificultades del problema son conocidas, y conjuntos de datos reales que son frecuentemente usados en minería de datos.

## B.1 Conjuntos de datos artificiales

Consideremos que tenemos un conjunto $\mathcal{E}$ formado por $n$ ejemplos que pueden describirse mediante la dupla $\langle \mathbf{e}_i, c_i \rangle,\ i = 1, \ldots, n$, donde cada $\mathbf{e}_i$ es un vector que viene descrito por $d$ variables de carácter cuantitativo y su correspondiente $c_i$ es un atributo cualitativo $\mathcal{C} \in \{+1, -1\}$ que describe la clase asociada al vector. Con $e_{ij}$ designaremos el elemento $j$ del ejemplo $i$.

### Características y generación de los problemas

Los problemas fueron definidos en función de 5 parámetros [80], cuyos valores se variaron con objeto de poder obtener problemas de distintas dificultades. Así un problema viene definido mediante la tupla $(n, d, r, l, \sigma)$, donde $n$ representa el número de ejemplos, $d$ el número total de atributos, $r$ el número de atributos relevantes, $l$ el tipo de regla de clasificación y $\sigma$ la tasa de ruido asociada a los atributos.

Un atributo consideraremos que es relevante para la tarea de clasificación si está presente en la definición de la regla de clasificación. En este caso consideraremos dos tipos de reglas de clasificación; una lineal ($l = 1$) y otra no lineal ($l = 2$). Para un ejemplo $i$ del conjunto $\mathcal{E}$, la regla de clasificación [119] viene definida por la siguiente expresión:

$$c_i \begin{cases} +1 & \text{si } p_i > \mu \\ \text{-1} & \text{en otro caso} \end{cases} \tag{B.1}$$

con

$$p_i = \prod_{k=1}^{l} \Big( \sum_{j=1}^{r} (w_{jk} \cdot e_{ij}) + b_k \Big).$$

donde $b_k$ es un término independiente generado aleatoriamente de forma que se generen todos los monomios hasta grado $l$, $\mu$ la mediana de $p_i$ y $w_{jk}$ los elementos de una matriz aleatoria $r \times 2$ con coeficientes en el rango $[-2, -1] \cup [+1, +2]$. El motivo de elegir este rango es el evitar que se generen coeficientes con valores próximos a 0, falseando así el subconjunto de atributos relevantes dado por el parámetro $r$. Cada elemento $e_{ij}$ de los $d$ atributos de los $n$ ejemplos fue generado mediante una distribución uniforme en el rango $[0, 1]$. La etiqueta de cada ejemplo $c_i$ fue asignada considerando la ecuación B.1, con todos los atributos relevantes teniendo el mismo peso. Los problemas constaban de 500 ejemplos con un número de atributos que variaban desde los 25 hasta los 200. El número de atributos relevantes se fijó a 5 y 10. Con objeto de aumentar la dificultad, se añadió un ruido gaussiano $\mathcal{N}(\mu, \sigma)$ con media $\mu = 0$ y varianza $\sigma = 0$ y 0,10. En la Tabla B.1 se resumen los valores considerados.

## B.2 Conjuntos de datos reales

Los conjuntos reales, a excepción de *ratio*, fueron recopilados por Merz y Murphy y están disponibles en el repositorio de aprendizaje automático de la Universidad de California Irvine [113]. Son frecuentemente utilizados

Tabla B.1: Características principales de los conjuntos de datos artificiales.

| *car.* | `valores` | |
|---|---|---|
| $n$ | 500 | |
| $d$ | $25, 50, 100, 200$ | |
| $r$ | 5 | 10 |
| $\sigma$ | 0 | 0,1 |
| $l$ | 1 | 2 |

por la comunidad de aprendizaje automático y minería de datos. En dicha página puede verse una descripción detallada de cada uno de los problemas considerados en esta memoria. El conjunto *ratio* fue recopilado por Joaquín Pacheco, de la Universidad de Burgos.

Se seleccionaron conjuntos de datos con distinto nivel de dificultad, de modo que se muestran ordenados en orden creciente del número de atributos, agrupándose en problemas pequeños (hasta 20 atributos), medianos (de 21 a 50 atributos) y grandes (más de 50 atributos). La línea simple separa los tres grupos considerados. Aquellos conjuntos de datos que coinciden en el número de atributos son ordenados en orden creciente del número de ejemplos.

La Tabla B.2 está organizada como sigue; en la primera columna se muestra el nombre real del conjunto de datos. La segunda representa el identificador asociado que será el que se utilice en la tablas de resultados. A continuación se detalla el número de ejemplos, y en las siguientes tres columnas la información relativa a la dimensión del problema. Dicha información es el número total de atributos, el número de atributos del total que son numéricos y categóricos. Le sigue la información del número de clases y por último si hay o no valores perdidos.

Tabla B.2: Características principales de los conjuntos de datos reales ordenados en función del número de atributos, y en caso de igualdad, en orden ascendente del número de instancias.

| datos | id. | #insts. | #atrs. | | | #clases | vp. |
|---|---|---|---|---|---|---|---|
| | | | #total | #num. | #nom. | | |
| pima diabetes | pdi | 768 | 8 | 8 | 0 | 2 | NO |
| glass | gla | 214 | 9 | 9 | 0 | 7 | NO |
| breast cancer | bca | 286 | 9 | 0 | 9 | 2 | SI |
| breast wisconsin | bwi | 699 | 9 | 9 | 0 | 2 | SI |
| wine | win | 178 | 13 | 13 | 0 | 3 | NO |
| heart-statlog | hst | 270 | 13 | 13 | 0 | 2 | NO |
| heart-hungary | hhu | 294 | 13 | 6 | 7 | 5 | SI |
| heart-cleveland | hcl | 303 | 13 | 6 | 7 | 5 | SI |
| vowel | vow | 990 | 13 | 10 | 3 | 11 | NO |
| credit rating | cra | 690 | 15 | 6 | 9 | 2 | SI |
| primary-tumor | ptu | 339 | 17 | 0 | 17 | 22 | SI |
| vote | vot | 435 | 17 | 0 | 16 | 2 | SI |
| lymphography | lym | 148 | 18 | 3 | 15 | 4 | NO |
| vehicle | veh | 846 | 18 | 18 | 0 | 4 | NO |
| hepatitis | hep | 155 | 19 | 6 | 13 | 2 | SI |
| horse colic | hco | 368 | 22 | 7 | 15 | 2 | SI |
| autos | aut | 205 | 25 | 15 | 10 | 7 | SI |
| sick-euthyroid | tse | 3163 | 25 | 7 | 18 | 2 | SI |
| allbp | abp | 3772 | 30 | 7 | 23 | 3 | SI |
| ionosphere | ion | 351 | 34 | 34 | 0 | 2 | NO |
| soybean | soy | 683 | 35 | 0 | 35 | 19 | SI |
| kr vs kp | kvk | 3196 | 36 | 0 | 36 | 2 | NO |
| anneal | ann | 898 | 38 | 6 | 32 | 6 | SI |
| lung-cancer | lca | 32 | 56 | 56 | 0 | 3 | SI |
| dna promoter | dnp | 106 | 58 | 0 | 58 | 2 | NO |
| sonar | son | 208 | 60 | 60 | 0 | 2 | NO |
| dna-nom | dna | 3186 | 60 | 0 | 60 | 3 | NO |
| splice | spl | 3190 | 62 | 0 | 62 | 3 | NO |
| audiology | aud | 226 | 69 | 0 | 69 | 24 | SI |
| ratio | rat | 198 | 142 | 142 | 0 | 2 | NO |
| arrhythmia | arr | 452 | 279 | 279 | 0 | 16 | SI |
| isolet | iso | 1559 | 617 | 617 | 0 | 26 | NO |

datos-nombre del conjunto de datos; id.-identificador asociado a un conjunto de datos; #insts.-número de instancias; #atrs.-número de atributos; #total-número total de atributos; #num.-número de atributos numéricos; #nom.-número de atributos nominales; #clases-número de clases; vp.-presencia de valores perdidos

# Bibliografía


[1] E. H. Aarts and J. K. Lenstra. *Local Search in Combinatorial Optimization*. Wiley, 1996.

[2] D. W. Aha and D. K. amd M. K. Albert. Instanced-based learning algorithms. *Machine Learning*, 6(37–66), 1991.

[3] H. Almuallim and T. G. Dietterich. Learning with many irrelevant features. In A. P. . T. M. Press, editor, *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 547–552, 1992.

[4] E. Alpaydin. Combined 5x2 cv for f test for comparing supervised classification learning algorithms. *Neural Computation*, 11:1885–1892, 1999.

[5] A. Bahamonde, G. F. Bayón, J. Díez, J. R. Quevedo, O. Luaces, J. J. del Coz, J. Alonso, and F. Goyache. Feature subset selection for learning preferences: a case study. In *21st International Conference on Machine Learning, ICML 2004*, pages 49–56, 2004.

[6] R. S. Barr, B. L. Golden, J. P. Kelly, M. G. C. Resende, and W. R. Stewart. Designing and reporting on computational experiments with heuristic methods. *Journal of Heuristics*, 1:9–32, 1995.

[7] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4):537–550, 1994.

[8] T. Bayes. An essay towards solving a problem in the doctrine of chances. *Philosophical Trans. Royal Society of London*, 53:370–418, 1763.

[9] D. Bell and H. Wang. A formalism for relevance and its application in feature subset selection. *Machine Learning*, 41(2):175–195, 2000.

[10] M. Birrattari, L. Paquete, T. Stützle, and K. Varrentrapp. Classification of metaheuristics and design of experiments for the analysis of components. Technical report, Darmstadt University of Technology, 2001.

[11] A. L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, pages 245–271, 1997.

[12] C. G. E. Boender, A. H. G. R. Kan, L. Stougie, and G. T. Timmer. A stochastic method for global optimization. *Mathematical Programming*, 22:125–140, 1982.

[13] G. Brassard and P. Bratley. *Fundamentals of Algorithms*. Prentice-Hall, 1996.

[14] V. Campos, F. Glover, M. Laguna, and R. Martí. An experimental evaluation of a scatter search for the linear ordering problem. *Journal of Global Optimization*, 21:397–414, 2001.

[15] J. G. Carbonell. Learning by analogy: Formulating and generalizing plans from past experience. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*, pages 137–161. Springer, Berlin, Heidelberg, 1984.

[16] C. Cardie and N. Howe. Empirical methods in information extraction. In D. Fischer, editor, *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 65–79. Morgan Kaufmann, 1997.

[17] C. Cardie93. Using decision trees to improve case-based learning. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 25–32, 1993.

[18] R. Caruana and D. Freitag. Greedy attribute selection. In *Proceedings of International Conference on Machine Learning*, pages 28–36. AAAI Press / The MIT Press, 1994.

[19] R. Caruana and D. Freitag. How useful is relevance? In *Working Notes of the AAAI Fall Symposium on Relevance*, pages 25–29, 1994.

[20] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, IT-13(1):21–27, 1967.

[21] M. Dash. Feature selection via set cover. In *Proceedings of IEEE Knowledge and Data Engineering Exchange Workshop*, pages 165–171, 1997.

[22] M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1(3), 1997.

[23] M. Dash and H. Liu. Hybrid search of feature subsets. In *Pacific Rim International Conference on Artificial Intelligence*, pages 238–249, 1998.

[24] L. Davis, editor. *handbook of Genetic Algorithms*. Van Nostrand Reinhold, 1991.

[25] I. J. G. del Amo, M. García-Torres, B. Melián-Batista, J. A. Moreno-Pérez, J. M. Moreno-Vega, and R. Rivero-Martín. Data mining with scatter search. *Lecture Notes in Computer Science*, 3643:199–204, 2005.

[26] J. Demšar. Statisticals comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[27] P. A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice Hall International, 1982.

[28] T. Dietterich. Approximate statistical test for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1924, 1998.

[29] J. Doak. An evaluation of feature selection methods and their application to computer security. Technical report, University of California, Department of Computer Science, 1992.

[30] P. Domingos. Contextsensitive feature selection for lazy learners. *Artificial Intelligence Review*, 11:227–253, 1997.

[31] M. Dorigo. *Optimization, Learning and Natural Algorithms*. PhD thesis, DEI, Politecnico di Milano, 1992.

[32] M. Dorigo and G. D. Caro. *The Ant Colony Optimization Metaheuristic*, pages 11–32. New Ideas in Optimization. McGraw-Hill, 1999.

[33] M. Dorigo, V. Maniezzo, and A. Colorni. Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics - Part B*, 26(1):29–41, 1996.

[34] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Jonh Wiley Sons, 1973.

[35] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons, inc, second edition, 2001. hardcopy.

[36] S. Dudani. The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-6(4):325–327, 1975.

[37] O. J. Dunn. Multiple comparisons among means. *Journal of the American Statistical Association*, 56:52–64, 1961.

[38] R. Etxebarria and P. Larrañaga. Global optimization with bayesian network. In *Proceedings of the II Symposium on Arti cial Intelligence, CIMAF99,*, pages 332–339, 1999.

[39] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *Ai Magazine*, 17:37–54, 1996.

[40] T. A. Feo and M. G. C. Rosende. Greedy randomized adaptive search procedures. *Journal of Global Optimizations*, 6:109–133, 1995.

[41] E. Fix and J. Hodges. Discriminatory analysis, nonparametric discrimination consistency properties. Technical report, US Air Force, School of Aviation Medicine, 1951.

[42] E. Fix and J. Hodges. Discriminatory analysis, nonparametric discrimination: Small sample performance. Technical report, US Air Force, School of Aviation Medicine, 1952.

[43] D. Foley. Considerations of sample and feature size. *IEEE Transactions Information Theory*, 18:618–626, 1972.

[44] I. Foroutan and J. Sklansky. Feature selection for automatic classification of non-gaussian data. *IEEE Transactions on Systems, Man and Cybernetics*, 17(2):187–198, 1987.

[45] M. Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32:675–701, 1937.

[46] M. Friedman. A comparison of alternative tests of significance for the problem of m rankings. *Annals of Mathematical Statistics*, 11:86–92, 1940.

[47] F. García-López, B. Melián, J. A. Moreno-Pérez, and J. M. Moreno-Vega. Parallelization of the scatter search. *Parallel Computing*, pages 575–589, 2003.

[48] F. C. García-López, M. García-Torres, B. Melián-Batista, J. A. Moreno-Pérez, and J. M. Moreno-Vega. Búsqueda dispersa y algoritmo genético para el problema de la selección de variables. In *Tercer Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB 2004)*, pages 609–616, 2004.

[49] F. C. García-López, M. García-Torres, B. Melián-Batista, J. A. Moreno-Pérez, and J. M. Moreno-Vega. Parallel scatter search for solving machine learning problems. In *Informs Annual Meeting*, 2004.

[50] F. C. García-López, M. García-Torres, B. Melián-Batista, J. A. Moreno-Pérez, and J. M. Moreno-Vega. *Parallel Metaheuristics: A New Class of Algorithms*, chapter Parallel Scatter Search, pages 223–246. Wiley, 2005.

[51] F. C. García-López, M. García-Torres, B. Melián-Batista, J. A. M. Pérez, and J. M. Moreno-Vega. Solving the feature selection problem by a parallel scatter search. *European Journal of Operations Research*, 169(2):477–489, 2006.

[52] F. C. García-López, M. García-Torres, J. A. Moreno-Pérez, and J. M. Moreno-Vega. Búsqueda dispersa para el problema de la selección de variables. In *X Conferencia de la Asociación Española para la Inteligencia Artificial. V Jornadas de Transferencia Tecnológica de Inteligencia Artificial (CAEPIA/TTIA 2003)*, volume 1, pages 405–414, 2003.

[53] F. C. García-López, M. García-Torres, J. A. Moreno-Pérez, and J. M. Moreno-Vega. Scatter search for the feature selection problem. *Lecture Notes in Artificial Intelligence*, 3040:517–525, 2004.

[54] M. García-Torres, F. C. García-López, B. Melián-Batista, J. A. Moreno-Pérez, and J. M. Moreno-Vega. Solving feature subset selection problem by a hybrid metaheuristic. In *First International Workshop in Hybrid Metaheuristics at ECAI 2004 (HM 2004)*, pages 59–69, 2004.

[55] J. H. Gennari, P. Langley, and D. Fisher. Models of incremental concept formation. *Artificial Intelligence*, 40(11-61), 1989.

[56] F. Glover. Heuristics for integer programming using surrogate constrints. *Decision Sciences*, 8:156–166, 1977.

[57] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 5:533–549, 1986.

[58] F. Glover. Tabu search for nonlinear and parametric optimization (with links to genetic algorithms). *Discrete Applied Mathematics*, 49:231–255, 1994.

[59] F. Glover. Ejection chains, reference structures and alternating path methods for traveling salesman problems. *Discrete Applied Mathematics*, 65:223–253, 1996.

[60] F. Glover. A template for scatter search and path relinking. *Artificial Intelligence, Lecture Notes in Computer Science*, 1363:13–54, 1998.

[61] F. Glover and M. Laguna. *Modern Heuristic Techniques for Combinatorial Problems*, chapter Tabu Search, pages 71–140. Blackwell Scientific Publishing, 1993.

[62] F. Glover and M. Laguna. *Tabu Search*. Klwer, 1997.

[63] F. Glover, M. Laguna, and R. Martí. Fundamentals of scatter search and path relinking. *Control and Cybernetics*, 29(3):653–684, 2000.

[64] D. E. Goldberg. *Genetics Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.

[65] L. K. Grover. Local search and the local structure of np-complete problems. *Operational Research Letter*, 12:235–243, 1992.

[66] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[67] P. Hansen and N. Mladenović. Variable neighborhood search for the $p$-median. *Location Science*, 5:207–226, 1997.

[68] P. Hansen and N. Mladenović. *Metaheuristics: Advances and Trends in Local Search Paradigms for Optimization*, chapter An Introduction to Variable Neighborhood Search, pages 433–458. Kluwer, 1999.

[69] P. Hansen and N. Mladenović. Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130:449–467, 2001.

[70] Y. Hochberg. A sharper bonferroni procedure for multiple tests of significance. *Biometrika*, 75:800–803, 1988.

[71] J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. University of Michigan Press, 1975.

[72] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6:65–70, 1979.

[73] R. C. Holte. Very simple classification rules perform well on most commonly use datasets. *Machine Learning*, 11:63–91, 1993.

[74] M. Ichino and J. Sklansky. Feature selection for linear classifier. In *Proceedings of the Seventh International Conference on Pattern Recognition*, pages 124–127. Morgan Kaufmann, 1984.

[75] M. Ichino and J. Sklansky. Optimun feature selection by zero-one programming. *IEEE Transactions on Systems, Man and Cybernetics*, 14(5):737–746, 1984.

[76] I. Inza, P. Larrañaga, R. Etxeberria, and B. Sierra. Feature subset selection by bayesian networks based optimization. *Artificial Intelligence*, 123(1-2):157–184, 2000.

[77] I. Inza, P. Larrañaga, and B. Sierra. Feature subset selection by bayesian networks: a comparison with genetic and sequential algorithms. *International Journal of Approximate Reasoning*, 27(2):143–164, 2001.

[78] G. H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Machine Learning: Proceedings of the Eleventh International Conference*, pages 121–1129. ICML94, Morgan Kaufmann, 1994.

[79] D. S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9:256–278, 1974.

[80] K. Jong, J. Mary, A. Cornuéjols, E. Marchiori, and M. Sebag. Ensemble feature ranking. In *PKDD'04: Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Datab*

*ases*, pages 267–278, New York, NY, USA, 2004. Springer-Verlag New York, Inc.

[81] K. Kira and L. A. Rendell. The feature selection problem. In *In Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 129–134. Menlo Park: AAAI Press / The MIT Press, 1992.

[82] S. Kirkpatrick, C. D. Gelatt, and M. P. vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[83] R. Kohavi. *Wrappers for Performance Enhancement and Oblivious Decision Graphs*. PhD thesis, Stanford University, Computer Science department, 1995.

[84] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[85] D. Koller and M. Sahami. Toward optimal feature selection. In *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 284–292, 1996.

[86] I. Kononenco. Estimating attributes: Analysis and extension of relief. In *In Proceedings of the European Conference on Machine Learning*, pages 171–182. Springer-Verlag, 1994.

[87] M. Laguna and R. Martí. *Scatter Search: Metodology and Implementations in C*. Kluwer Academic Press, 2003.

[88] P. Langley. Selection of relevants features in machine learning. In *Proceedings of the AAAI Fall Symposium on Relevance*, 1994.

[89] P. Langley, W. Iba, and K. Thompson. An analysis of bayesian classifiers. In *In Proceedings of the 10th National Conference on Artificial Intelligence*, pages 223–228, 1992.

[90] P. Langley and S. Sage. Oblivious decision trees and abstract cases. In *Working Notes of the AAAI-94 Workshop on Case-Based Reasoning*. AAAI Press, 1994.

[91] P. Larrañaga and J. A. Lozano. *Estimaation Distribution of Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic Publisher, 2001.

[92] M. Lau and M. Schultz. A feature selection method for gene expression data with thousands of features. http://zoo.cs.yale.edu/classes/cs490/02-03b/manfred.lau/, 2003.

[93] H. Liu, H. Motoda, and M. Dash. A monotonic measure for optimal feature slection. In C.Ñedellec and C. Rouveirol, editors, *Machine Learning: ECML-98*, pages 101–106. Springer-Verlag, 1998.

[94] H. Liu and R. Setiono. Feature selection and classification - a probabilistic wrapper approach. In *Proceedings of the Ninth International Conference on Industrial and Engineering Applications of AI and ES*, pages 419–424, 1996.

[95] H. Liu and R. Setiono. A probabilistic approach to feature selection - a filter solution. In *Proceedings of International on Machine Learning*, pages 319–327. Morgan Kaufmann, 1996.

[96] H. Liu and R. Setiono. Scalable feature selection for large sized databases. In *Proceedings of the Fourth World Congress on Expert Systems*, volume 1. Morgan Kaufmann, 1998.

[97] H. Liu and L. Yu. Feature selection for data mining. Technical report, Department of Computer Science and Engneering, 2003.

[98] M. Los and C. Lardinois. Combinatorial programming, statistical optimization and the optimal transportation network problem. *Transportation Research*, 2:89124, 1982.

[99] B. Melián, J. A. Moreno-Pérez, and J. M. Moreno-Vega. Metaheurísticas: una visión global. *Revista Iberoamericana de Inteligencia Artificial*, 2(19):7–28, 2003.

[100] N. Metropolis, A. W. Rosenbluth, M.Ñ. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculation by fast computing machines. *Journal of Chemistry Physics*, 21:1087–1091, 53.

[101] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Verlag, 1992.

[102] R. S. Michalski. A theory and methodology of inductive learning. *Artidicial Intelligence*, 20:111–116, 1983.

[103] D. Michie, D. J. Spiegelhalter, C. C. Taylor, and J. Campbell, editors. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, 1994.

[104] M. Mitchel. *An Introduction to Genetic Algorithms*. MIT Press, 1996.

[105] T. M. Mitchell. *Machine Learning*. Series in Computer Science. McGraw-Hill, 1997.

[106] M. Modrzejewski. Feature selection using rough sets theory. In *Proceedings of the European Conference on Machine Learning*, pages 213–226, 1993.

[107] A. W. Moore and M. S. Lee. Efficient algorithms for minimizing cross validation error. In *International Conference on Machine Learning*, pages 190–198, 1994.

[108] J. A. Moreno-Pérez, M. García-Torres, B. Melián-Batista, J. M. Moreno-Vega, and R. Rivero-Martín. Búsquedas dispersa y de entorno variable en minería de datos. In *III Taller Nacional de Minería de Datos y Aprendizaje (TAMIDA 2005)*, pages 309–316, 2005.

[109] A.Ñ. Mucciardi and E. E. Gose. A comparison of seven techniques for choosing subsets of pattern recognition. *IEEE Transactions on Computers*, 20:1023–1031, 1971.

[110] H. Mühlenbein and G. Paa$\beta$. From recombination of genes to the estimation of distribution i. binary parameters. *Lecture Notes in Computer Science 1411: Parallel problem Solving from Nature - PPSN IV*, pages 178–187, 1996.

[111] P. M. Narenda and K. Fukunaga. A branch and bound algorithm for feature subset selection. *IEEE Trans. on Computer*, 26(9):917–922, 1977.

[112] P. B. Nemenyi. *Distribution-free Multiple Comparisons*. PhD thesis, Princeton University, 1963.

[113] D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz. UCI repository of machine learning databases. http://www.ics.uci.edu/∼mlearn/MLRepository.html, 1998.

[114] A. L. Oliveira and A. S. Vincentelli. Constructive induction using a non-greedy strategy for feature selection. In *Proceedings of the Ninth International Conference on Machine Learning*, pages 355–360. Morgan Kaufmann, 1992.

[115] M. Pirlot. General local search methods. *European Journal of Operational Research*, 92(3):493–511, 1996.

[116] P. Pudil, J.Novovicova, and J. Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15:1119–1125, 1994.

[117] D. Pyle. *Data Preparation for Data Mining*. Morgan Kaufmann Publishers, 1999.

[118] C. E. Queiros and E. S. Gelsema. On feature selection. In *Proceedings of the Seventh International Conference on Pattern Recognition*, pages 128–130, 1984.

[119] J. R. Quevedo, A. Bahamonde, and O. Luaces. A simple and efficient method for variable ranking according to their usefulness for learning.

Technical report, Artificial Intelligence Center, University of Oviedo at Gijón, 2006.

[120] J. R. Quinlan. Discovering rules by induction from large collections of examples. In D. Michie, editor, *Expert Systems in the Micro-Electronic Age*, pages 168–201. Edinburgh University Press, 1979.

[121] J. R. Quinlan. Learning efficient classification procedures and their application to chess end games. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*, pages 463–482. Springer, 1984.

[122] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[123] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[124] J. R. Quinlan. *c4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[125] J. Reunanen. Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research*, 3:1371–1382, 2003.

[126] A. M. Ribas, E. A. Voltas, J. B. Alonso, L. B. Muñoz, U. C. García, R. G. Mestre, J. M. G. Illa, B. L. Ibáñez, M. M. Muñoz, and M. S. Marré. *Aprendizaje Automático*. Edicions UPC, 1994.

[127] J. Rissanen. Modelling by the shortest data description. *Automatica*, 14:465–471, 1978.

[128] M. Scherf and W. Brauer. Improving rbf networks by the feature selection approach eubafes. In *Proceedings of the 7th International Conference on Artificial Neural Networks*, pages 391–396, 1997.

[129] J. C. Schlimmer. Efficiently inducing determinations: A complete and systematic search algorithm that uses optimal pruning. In *In Proceedings og the Tenth Inetrnational Conference on Machine Learning*, pages 284–290. ICML93, 1993.

[130] D. Schuurmans and F. Southey. Metric-based methods for adaptive model selection and regularization. *Machine Learning*, 48:51–84, 2002.

[131] J. Segen. Feature selection and constructive inference. In *In Proceedings of the Seventh International Conference on Pattern Recognition*, pages 1344–1346, 1984.

[132] J. Sheinvald, B. Dom, and W.Ñiblack. A modelling approach to feature selection. In *In Proceedings of the Tenth International Conference on Pattern Recognition*, pages 535–539, 1990.

[133] W. Siedlecki and J. Sklansky. On automatic feature selection. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(2):197–220, 1988.

[134] W. Siedlecki and J. Sklansky. A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters*, 10:335–347, 1989.

[135] H. A. Simon. Why should machines learn? In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*, pages 25–37. Springer, Berlin, Heidelberg, 1984.

[136] T. Stützle. *Local Search Algorithms for Combinatorial Problems-Analysis, Improvements and New Applications*. PhD thesis, Darmstadt University of Technology, Department of Computer Science, 1998.

[137] H. Vafaie and I. F. Imam. Feature selection methods: Genetic algorithms vs. greedy-like search. In *Proceedings of the International Conference on Fuzzy and Intelligent Control Systems*, 1994.

[138] V. Vapnik. *Statistical Learning Theory*. John Wiley, 1998.

[139] J. M. M. Vega and J. A. M. Perez. *Heurísticas en Optimización*. Colección Textos Universitarios. Gobierno de Canarias, 1999.

[140] C. Voudouris and E. P. K. Tsang. Guided local search. Technical report, University of Essex, Dept. of Computer Science, 1995.

[141] C. Voudouris and E. P. K. Tsang. Guided local search. *European Journal of Operational Research*, 113:469–499, 1999.

[142] H. Wang, D. Bell, and F. Murtagh. Axiomatic approach to feature subset selection based on relevance. *IEEE Transactions on Pattern Analysis and machine Intelligence*, 21(3):271–277, 1999.

[143] K. Wang and S. Sundaresh. Selecting features by vertical compactness of data. In D. Heckerman, H. Mannila, and D. Pregibon, editors, *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pages 275–278. AAAI Press, 1997.

[144] D. Wettschereck and T. G. Dietterich. An experimental comparison of the nearest-neighbor and nearest-hyperrectangle algorithms. *Machine Learning*, 19(1):5–27, 1995.

[145] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics*, 1:80–83, 1945.

[146] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, 2000.

[147] L. Xu, P. Yan, and T. Chang. Best first strategy for feature selection. In *In Proceedings of the Ninth International Conference on Pattern Recognition*, pages 706–708, 1988.

[148] M. Yagiura and T. Ibaraki. *handbook of Applied Optimization*, pages 104–123. Oxford University Press, 2002.

[149] J. Yang and V. Honavar. Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems*, 13(2):44–49, 1998.

[150] L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. *J. Mach. Learn. Res.*, 5:1205–1224, 2004.

# Índice alfabético