Universidad de Huelva

Departamento de Tecnologías de la Información

Nuevas propuestas en el ámbito de los operadores adaptativos para Sistemas Difusos Lingüísticos Evolutivos Multiobjetivo

Memoria para optar al grado de doctor
presentada por:

Antonio Ángel Márquez Hernández

Fecha de lectura: 5 de febrero de 2016

Bajo la dirección de los doctores:

Antonio Peregrín Rubio

Francisco Alfredo Márquez Hernández

Huelva, 2016

UNIVERSIDAD DE HUELVA

Universidad de Huelva

Nuevas Propuestas en el Ámbito de los Operadores Adaptativos para Sistemas Difusos Lingüísticos Evolutivos Multiobjetivo

Tesis Doctoral

Antonio Ángel Márquez Hernández

Huelva, Noviembre de 2015

Departamento de Tecnologías de la Información

UNIVERSIDAD DE HUELVA

Universidad de Huelva

Nuevas Propuestas en el Ámbito de los Operadores Adaptativos para Sistemas Difusos Lingüísticos Evolutivos Multiobjetivo

MEMORIA QUE PRESENTA

Antonio Ángel Márquez Hernández

PARA OPTAR AL GRADO DE DOCTOR EN INFORMÁTICA

Noviembre de 2015

DIRECTORES

Dr. Antonio Peregrín Rubio
Dr. Francisco Alfredo Márquez Hernández

Programa de Doctorado
Tecnologías Informáticas Avanzadas

Departamento de Tecnologías de la Información

La memoria titulada "*Nuevas Propuestas en el Ámbito de los Operadores Adaptativos para Sistemas Difusos Lingüísticos Evolutivos Multiobjetivo*", que presenta D. Antonio Ángel Márquez Hernández para optar al grado de doctor, ha sido realizada dentro del programa de doctorado "*Tecnologías Informáticas Avanzadas*" del Departamento de Tecnologías de la Información de la Universidad de Huelva bajo la dirección de los doctores D. Antonio Peregrín Rubio y D. Francisco Alfredo Márquez Hernández.

Huelva, Noviembre de 2015

El Doctorando

Fdo: Antonio Ángel Márquez Hernández.

El Director

Fdo: Dr. Antonio Peregrín Rubio

El Director

Fdo: Dr. Francisco Alfredo Márquez Hernández

Agradecimientos

"Lo que sabemos es una gota de agua, lo que ignoramos es el océano"

Isaac Newton

Decir que la elaboración de una tesis doctoral es un proceso largo, intermitente en intensidad, difícil y frustrante en muchos momentos, es algo obvio para cualquiera que haya alcanzado el grado de Doctor. Decir que el esfuerzo que supone culminar esta tarea resulta esencial para madurar, no sólo profesionalmente sino como persona, es seguramente otra reflexión que muchos habrán hecho en circunstancias similares. Pero lo que sin lugar a dudas es más cierto cuando se refiere a trabajos que se extienden en el tiempo, es que uno jamás podría haber llegado al resultado final sin la complicidad, paciencia, ánimo y ayuda de las personas que te acompañan durante el viaje. El apoyo de todos y cada uno de ellos, expresado en una multitud de formas diferentes, ha sido fundamental para llegar hasta aquí, de ahí que quiera dedicar unas líneas para poder expresar mi más sincero agradecimiento.

En primer lugar, gracias a mi director de tesis Antonio Peregrín, que con su tiempo, esfuerzo, paciencia (mucha paciencia) y sobre todo su comprensión en los momentos difíciles me ha ayudado a culminar este trabajo. Bien está lo que bien acaba, y más cuando lo que hemos puesto es tan sólo un punto y seguido.

Gracias también a mi hermano y co-director de tesis Paco Márquez por su inestimable ayuda, no sólo profesional sino (mucho más importante) personal. Sin su cariño, ánimo, apoyo y optimismo en los momentos de crisis no hubiera podido finalizar este viaje.

Gracias a toda mi familia por su constante preocupación, sobre todo a mis padres Paco y Charo (qué raro me resulta llamaros así, acostumbrado siempre a llamaros Papá y Mamá) por querer siempre lo mejor para sus hijos, por enseñarme que la familia es lo más importante y por sacrificarse durante toda su vida por mí y por mis hermanos. Quiero que sepáis que estoy muy orgulloso de vosotros y que sois para mí un ejemplo a seguir.

Gracias también a mis otros dos hermanos Silvia e Ismael por ser como sois y por haberme alegrado estos años de tesis con el nacimiento de mis dos sobrinas Manuela y Lucía. Gracias también a mis sobrinos mayores Paco y Adrian y mis sobrinos pequeños Leo y Hugo (por cierto Gloria ¿de verdad que no te animas a darle una hermanita a tus dos hijos? La última vez me quedé con las ganas de ser el padrino…) por la alegría que me dais cada vez que os veo y gracias también a mis cuñadas Mari Carmen e Inma (Javi tu también) por haber dado unos sobrinos tan guapos.

Y termino con la que he dejado para el final pero que para mí es el comienzo, el punto medio y el todo. Mi verdadera razón para empezar, mi verdadera razón para seguir y mi verdadera razón para por fin terminar eres tú, Rocío. Este trabajo ha coincidido con el principio de nuestra relación, y en estos años has tenido que soportar los momentos de crisis – y también los de euforia- que inevitablemente suceden. Gracias Rocío, por tu infinito apoyo, tu comprensión, tu paciencia, tu amor en definitiva y perdóname por no haberte dedicado, a ti y a tu hija Celia, todo el tiempo que hubiera querido.

Estoy convencido de que al final todos los sacrificios han merecido la pena: este trabajo está dedicado, nunca mejor dicho, a ti.

Índice

Índice de figuras

Índice de Tablas

Tabla de Acrónimos

AE	Algoritmo Evolutivo	2
AG	Algoritmo Genético	2
AGMO	Algoritmo Genético Multiobjetivo	3
AEMO	Algoritmo Evolutivo Multiobjetivo	12
BC	Base de Conocimiento	1
BD	Base de Datos	1
BR	Base de Reglas	1
COR	Metodología de Cooperación entre Reglas	41
ECM	Error Cuadrático Medio	19
ID	Interfaz de Defuzzificación	2
MI	Mecanismo de Inferencia	1
ODA	Operador Difuso Adaptativo	4
SBRD	Sistema Basado en Reglas Difusas	1
SI	Sistema de Inferencia	2
SDE	Sistema Difuso Evolutivo	2
SDEMO	Sistema Difuso Evolutivo Multiobjetivo	3
WM	Método ad-hoc guiado por ejemplos de Wang y Mendel	46

Introducción

A Planteamiento

La mayoría de los sistemas que se desean modelar hoy día comparten una serie de características, tales como necesidad de una importante precisión o la carencia de conocimiento formal sobre su funcionamiento, que dificultan su modelado con técnicas tradicionales, y en muchos de ellos interviene información compleja, incompleta e imprecisa que, aunque un experto es capaz de procesar, resulta difícil definir y representar en un sistema. Por ello, para diseñar y modelar un sistema de forma robusta y precisa, con rendimiento e interpretabilidad altos, se hace necesario el uso de nuevas herramientas que permitan tratar este tipo de información.

En este sentido, en las últimas décadas la construcción de modelos basados en la Lógica Difusa [Zad65, Zad73] ha tenido una gran difusión debido a su flexibilidad para ser aplicados en distintos problemas gracias a que proporcionan más generalidad, poder expresivo y tolerancia frente a la imprecisión que otros métodos. En particular, los sistemas difusos más estudiados y utilizados son los Sistemas Basados en Reglas Difusas (SBRDs), que han sido aplicados con éxito a campos tan diferentes como el *modelado difuso de sistemas* [BD95, Ped96, SY93], *control difuso* [DHR93, Hir93, Wan94], *clasificación* y problemas de *minería de datos* [INN04, Kun00].

Los SBRDs cuentan con un conjunto de reglas que permiten representar conceptos imprecisos e incluir conocimiento previo de una forma comprensible para los usuarios del sistema. Específicamente, un SBRD se compone de una Base de Conocimiento (BC), que incluye la información en forma de reglas difusas *IF-THEN* en la Base de Reglas (BR) y la correspondencia de los valores difusos en la Base de Datos (BD), y un Mecanismo de Inferencia (MI), que

incluye una interfaz de fuzzificación, un Sistema de Inferencia (SI) y una Interfaz de Defuzzificación (ID). El diseño de este tipo de sistemas es realizado por un experto mediante la definición de las variables lingüísticas asociadas al problema, el conjunto de reglas difusas y el SI.

Para facilitar la tarea de diseño se han desarrollado además diferentes métodos de aprendizaje automático los cuales tienen como objetivo obtener SBRDs a partir de un conjunto de pares de datos entrada-salida, y que reflejan el comportamiento del sistema real que se desea modelar. Dicho aprendizaje se puede enfocar y resolver como un problema de optimización o búsqueda, por lo que, a la hora de determinar el conjunto de reglas y operadores difusos que mejor representan el comportamiento del sistema, uno de los métodos utilizados y que mejores resultados ofrece son los Algoritmos Evolutivos (AEs) y más concretamente los Algoritmos Genéticos (AGs), gracias a su capacidad para incluir conocimiento a priori [CHHM01], y su habilidad para encontrar soluciones casi óptimas en espacios de búsqueda complejos.

La hibridación entre la Lógica Difusa [YF94] y los AEs ha dado lugar a los Sistemas Difusos Evolutivos (SDEs) [Her08, FAN+13], y en particular a los Sistemas Basados en Reglas Difusas Evolutivos (SBRDEs), donde el proceso de diseño del SBRD se considera como un problema de optimización o búsqueda y el AE es utilizado para resolver este problema.

En el diseño de un algoritmo de aprendizaje de SBRDs existen dos objetivos principales que permiten asegurar la calidad del modelo difuso obtenido y que se deben maximizar [CCHM03a, CCHM03b, GAH11]: la precisión y la interpretabilidad. Estas dos propiedades son contradictorias y dependen directamente del proceso de aprendizaje y del tipo de SBRD utilizado. De esta forma, dependiendo del objetivo principal que se desee satisfacer y de acuerdo a la estructura de regla considerada, los SBRDs pueden dividirse en dos categorías. La primera está formada por los SBRDs lingüísticos (también conocidos como de tipo Mamdani [MA75, Mam74]), que permite obtener modelos claramente interpretables por el ser humano. Sin embargo, presenta un problema fundamental, que es su falta de precisión al modelar sistemas complejos. La segunda comprende los modelos difusos precisos aunque menos interpretables debido a la estructura de sus reglas, los cuales se implementan mediante los SBRDs aproximativos [ACCH01, BD95, CFM96, CH97, Koc96] y los de tipo Takagi-Sugeno-Kang (TSK) [SK88, TS85]. La primera de estas dos categorías es la que más desarrollo y éxito ha tenido en el campo de la Inteligencia Artificial al ser sus modelos más interpretables y comprensibles para los humanos.

Aunque inicialmente los investigadores se centraron en encontrar modelos precisos en lugar de modelos interpretables, esta tendencia fue cambiando, de forma que cada vez son más los investigadores que apuestan por modelos interpretables a la vez que precisos [AM11]. Esta interpretabilidad es mayor si el modelizado es de tipo lingüístico, por lo que en esta memoria de tesis nos centramos en este tipo de modelado.

Una buena forma de optimizar estos criterios de interpretabilidad y precisión al mismo tiempo durante el proceso de diseño es mediante el uso de Algoritmos Genéticos Multiobjetivo (AGMOs), ya que permiten obtener un conjunto de soluciones no dominadas para cada uno de los objetivos considerados. Por ello, en las últimas décadas, los SDEs se han centrado en desarrollar modelos que permiten manejar múltiples objetivos, extendiéndose el uso de AGMOs para tal fin, mejorando la precisión de éstos y tratando de evitar el descenso de la interpretabilidad del sistema [ADH+09, ADLM09, AGHAF07, ANHI11, BLMS09, CHP+07, Cor11, GAH09, GAH10, IN07, INN11], dando lugar esta hibridación a los Sistemas Difusos Evolutivos Multiobjetivos (SDEMOs) [FAN+13].

Con el fin de medir la interpretabilidad de los SBRDs, aunque en la literatura han aparecido diversos índices y taxonomías que los clasifican [AMGR09, GAH11, MF08, ZG08], actualmente los investigadores concuerdan en distinguir dos formas de manejar la interpretabilidad:

- Controlar la complejidad del modelo global por medio de medidas como el número de reglas, de variables, de etiquetas por regla, etc.

- Medir la interpretabilidad de las particiones difusas usando medidas de interpretabilidad semántica (cobertura, distinguibilidad, consistencia, etc.).

Si bien se ha trabajado en los últimos años en esta faceta y se han producido mejoras en la comprensión y cuantificación de la interpretabilidad, en realidad sigue siendo un problema abierto, especialmente el de su aspecto semántico [ADLM11, AMGR09, GAH10].

Actualmente, aunque el área de investigación sobre SDEMOs se puede considerar madura, aún quedan problemas sin resolver, como ocurre por ejemplo cuando nos encontramos en presencia de problemas de alta dimensionalidad [Jin00]. Habitualmente se ha optado por aprender o ajustar los componentes de la BC, esto es, BR y BD, al ser el elemento fundamental que define el comportamiento del sistema. Sin embargo, la parametrización del MI y más concretamente del SI y la ID que lo componen es otra posibilidad

interesante [AFHMP07, CHMP04, HMP03, MPH07] poco explotada con SDEMOs, los cuales pueden encontrar una forma de defuzzificar la contribución de cada regla adecuadamente [CHMP04] así como mejorar la cooperación entre las reglas [AFHMP07] conservando la BC original, a diferencia de las otras alternativas.

El hecho de que esta técnica sea complementaria a las basadas en las mejoras de la BC, la hace aún más interesante, ya que la combinación de ambas permite lograr mejores modelos y afrontar desde su perspectiva particular algunos de los retos vigentes señalados en la literatura del área [FAN+13, FLdJH15, Her08]).

B Objetivos

El objetivo general del trabajo desarrollado en esta memoria es afrontar desde el punto de vista de los operadores difusos adaptativos del MI, algunos de los problemas relevantes para el área de los sistemas difusos, esto es, proponer modelos con buen equilibrio entre precisión e interpretabilidad, avanzar en el uso de medidas y diseño de índices que permitan cuantificar y optimizar mejor la interpretabilidad, y proponer métodos para poder abordar el diseño de sistemas difusos partiendo de conjuntos de datos de mayor tamaño.

Para desarrollar este objetivo general, definimos los siguientes objetivos particulares:

- *Mejorar la interpretabilidad y la precisión mediante la cooperación entre el MI y la BR.* Se pretende desarrollar un modelo evolutivo basado en AGMO para el aprendizaje cooperativo y simultáneo de los operadores difusos adaptativos (ODAs) que intervienen en el SI y la ID, junto con la propia BR, para mostrar que es posible alcanzar una sinergia positiva entre ambas partes del modelo.

- *Proponer mecanismos que permitan cuantificar y optimizar cómo afecta a la interpretabilidad el uso de operadores adaptativos.* Se pretende estudiar qué medidas serían adecuadas y combinarlas en un índice que se pueda emplear para su optimización junto con la precisión mediante un AGMO, y así obtener no sólo sistemas difusos más interpretables

sino también más precisos gracias a la reducción de la complejidad del modelo.

- *Incrementar la escalabilidad y mejorar el rendimiento y la eficiencia de la inferencia adaptativa.* Se trata de desarrollar propuestas de diseño sobre los operadores adaptativos aprendidos mediante AGMOs que puedan utilizarse con conjuntos de entrenamiento más grandes y complejos, en definitiva, con conjuntos de reglas mayores.

C Resumen

La memoria se ha organizado en cuatro capítulos con los que se pretende alcanzar los objetivos anteriormente propuestos, una sección de comentarios finales y cuatro apéndices. La estructura de cada una de estas partes se introduce brevemente a continuación.

El Capítulo 1 contiene la introducción a este trabajo de forma que oriente al lector sobre el tipo de problemas que pretendemos resolver. Para ello se introducen los SDEs y los SDEMOs, describiéndolos con detalle y mostrando una taxonomía que los clasifica en función de diferentes criterios. Se presenta también una revisión de los distintos enfoques existentes en la literatura para abordar el equilibrio entre interpretabilidad y precisión con SDEMOs y se explica cómo se puede conseguir este objetivo a través de los operadores del MI. Finalmente, se introducen los nuevos problemas en SDEs y SDEMOs centrándose finalmente en aquellos relacionados con el aprendizaje de los operadores del SI y la ID.

En el Capítulo 2 se presenta una propuesta multiobjetivo en la que el aprendizaje de los operadores del MI y la BR se realiza de forma conjunta y cooperativa con objeto de obtener sistemas más precisos e interpretables gracias a la simbiosis derivada del aprendizaje simultáneo de ambas partes del modelo. Para ello se introducen los operadores difusos adaptativos y el algoritmo de aprendizaje de la BR utilizados en la propuesta, y se propone el modelo de aprendizaje evolutivo multiobjetivo específicamente adaptado. Finalmente se desarrolla un estudio experimental y se muestran los resultados obtenidos validados mediante el uso de test estadísticos no paramétricos.

En el Capítulo 3 se presenta una propuesta para cuantificar la pérdida de interpretabilidad que se produce en los sistemas difusos con defuzzificación adaptativa, y se presenta un modelo multiobjetivo que optimiza precisión e interpretabilidad usando dicha medida. Para ello se propone un mecanismo de mejora de la interpretabilidad así como las métricas necesarias para implementar dicho mecanismo. Finalmente se realiza un estudio experimental, y al igual que en el capítulo anterior, los resultados obtenidos en el estudio experimental son validados mediante el uso de test estadísticos no paramétricos.

En el Capítulo 4 se presenta una propuesta para incrementar la escalabilidad y el rendimiento de los sistemas difusos lingüísticos que tienen un SI adaptativo, a fin de poder ser usados en problemas que presentan alta dimensionalidad. Para ello se propone un nuevo modelo de operador de conjunción adaptativo cuyo ajuste evolutivo multiobjetivo es más rápido y eficiente. Se lleva a cabo un estudio experimental dividido en tres partes en el que se compara la nueva propuesta: en la primera se compara la solución más precisa alcanzada por cada modelo, en la segunda se comparan los tiempos de ejecución de cada uno de ellos y en la tercera y última se comparan los frentes de Pareto obtenidos por cada modelo. Tal y como se hizo en los capítulos anteriores, los resultados del estudio experimental son validados con test estadísticos no paramétricos que confirman las ventajas de las propuestas realizadas.

En la sección de Comentarios Finales se resumen los resultados obtenidos en esta memoria así como las principales conclusiones a las que hemos llegado en la investigación. Esta sección incluye las líneas futuras como posible continuación de este trabajo.

Se incluyen cuatro apéndices dedicados respectivamente a introducir los AEs y la Lógica Difusa, los métodos de aprendizaje de BR utilizados en los estudios experimentales de esta tesis, una descripción general de los test estadísticos utilizados en esta memoria y una descripción de los distintos tipos de SBRDs conocidos, explicando en profundidad los SBRD lingüísticos, sus componentes y el MI adaptativo. La memoria termina con la recopilación bibliográfica de las contribuciones más destacadas en la materia estudiada referenciadas a lo largo de la memoria.

Capítulo 1

Sistemas Difusos Lingüísticos Evolutivos Multiobjetivo

Hoy en día, la mayoría de los problemas son tan complejos y ambiguos que se hace necesario el uso de modelos matemáticos inteligentes capaces de extraer el conocimiento para facilitar su compresión y modelado. En este sentido muchos investigadores del área de la Inteligencia Artificial se han centrado en las últimas décadas en el desarrollo de métodos de aprendizaje automáticos basándose en modelos de toma de decisiones, neuronales, evolutivos y lingüísticos que han permitido resolver problemas tales como la clasificación (predicción de categorías), la optimización (maximización o minimización de un valor) y la regresión (predicción de un valor).

Si bien es verdad que en la literatura existen numerosas técnicas y métodos de aprendizaje automático para modelar este tipo de problemas, no es menos cierto, que las técnicas de aprendizaje aplicadas a regresión, lo son en menor medida aun cuando este tipo de problemas son muy importantes y están ampliamente extendidos, por lo que en esta memoria nos vamos a centrar en este tipo de problemas.

Para solventar los citados problemas de regresión, esto es, para predecir el valor de una variable o variables continuas en función de otras, se han desarrollado diferentes técnicas, tanto matemáticas (mediante recursos estadísticos) como de aprendizaje automático (mediante algoritmos de aprendizaje), siendo éstas últimas las que proporcionan una mayor legibilidad y eficiencia.

El aprendizaje automático [YL14] contiene dos áreas principales: el aprendizaje supervisado y el aprendizaje no supervisado. Mientras que en el

aprendizaje no supervisado no existe un conocimiento a priori, el aprendizaje supervisado parte de unos datos de entrenamiento que permiten realizar una predicción basada en el comportamiento de esos datos. De entre todas las técnicas de aprendizaje supervisado (redes neuronales [Ata15, Roj96, SS15, WCWW15, ZLSJ15], redes bayesianas [Hül08, JLM14, Sun13, YL14], árboles de decisión [Kot13, Qui86, Shi14, UAG99]) existentes, los SBRDs son los que han tenido una mayor difusión y éxito debido a su flexibilidad para ser aplicados en distintos problemas gracias a que proporcionan más capacidad de generalización, poder expresivo y robustez frente a la imprecisión e incertidumbre que las otras técnicas.

1.1. Sistemas Difusos Evolutivos

El Modelado Difuso o Modelado de Sistemas mediante SBRDs [MAS+15] permite tratar la imprecisión e incertidumbre de los datos gracias al uso de un lenguaje descriptivo basado en la Lógica Difusa [Zad65, Zad73] que permite representar dicho conocimiento mediante un conjunto de reglas de una forma comprensible para los humanos. Para determinar o aprender el conjunto de reglas y operadores difusos que mejor representan el comportamiento del sistema, debemos optimizar y buscar aquella combinación de reglas y operadores que mejores resultado nos proporcione. En este sentido, los AEs y más concretamente los AGs son considerados tanto desde un punto de vista teórico como práctico, una de las técnicas de búsqueda más versátiles y robustas, y que por ello mejores resultados ofrecen gracias a su habilidad para encontrar soluciones casi óptimas en espacios de búsqueda complejos, así como a su capacidad para incluir conocimiento experto previo y trabajar con diferentes tipos de estructuras solución (modelos) [CHHM01].

En el caso de los SBRDs, este conocimiento experto previo puede ser en forma de variables lingüísticas, granularidad, parámetros de la inferencia, reglas difusas, número de reglas, etc., los cuales pueden ser codificados en el cromosoma y/o considerados en la búsqueda evolutiva definiendo diferentes mecanismos para tenerlos en cuenta. Estas capacidades han hecho que a lo largo de estos últimos años se haya extendido el uso de los AEs en el desarrollo de una amplia gama de métodos para diseñar SBRDs orientados a obtener de un modo automático la totalidad o una parte de la BC o los operadores que intervienen en el proceso de inferencia del SBRD, como se ilustra en la Figura 1.1. El aprendizaje de los distintos componentes de un SBRDs mediante AEs ha dado lugar a los denominados SDEs [CGH+04, FAN+13, Her08].

Un SDE [FAN+13, Her08] llamado en inglés *Evolutionary Fuzzy System* (EFS) es básicamente un Sistema Difuso mejorado por un proceso de aprendizaje evolutivo. El aprendizaje evolutivo es un campo del aprendizaje automático que tiene como objetivo resolver problemas de optimización. Este tipo de algoritmos realizan una exploración y una explotación del espacio de búsqueda mediante una simulación de la evolución [YL14], partiendo de una población inicial aleatoria que evoluciona mediante la selección de un conjunto de individuos (cada uno de los cuales representa una solución del espacio de búsqueda), a los que se les aplican unos operadores genéticos (mutación y cruce) de forma probabilística para obtener nuevas generaciones. Los AEs abarcan varias ramas [BS02], que incluye AGs [SD08], Programación Genética [GT12], Programación Evolutiva [FF96, Fog12] y Estrategias de Evolución [BFK13], entre otras.

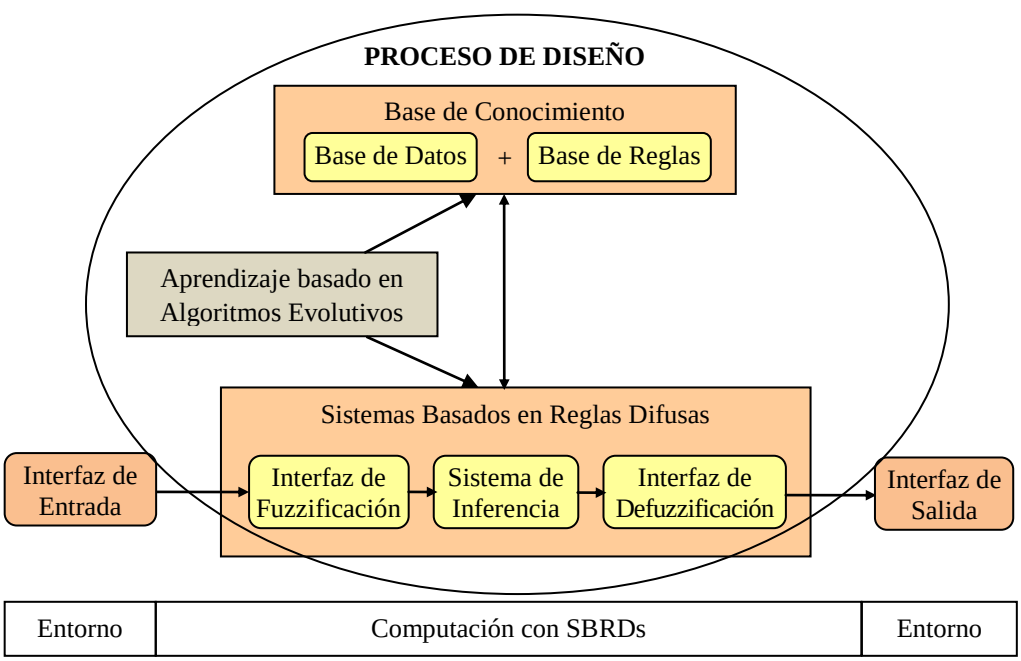


Figura 1.1: Sistemas Difusos Evolutivos

A la hora de diseñar un SBRD existen dos objetivos principales (y opuestos) que permiten asegurar la calidad del modelo difuso obtenido y que se deben maximizar: la precisión y la interpretabilidad. La presencia de múltiples

objetivos en un problema provoca que no exista una solución óptima única, sino que exista un conjunto de soluciones óptimas (conocidas como soluciones Pareto óptimas), al no haber ninguna que domine a las demás soluciones en todas las funciones objetivo a optimizar. En este caso los AEs tradicionales no son una herramienta adecuada, ya que al permitir optimizar sólo una de estas magnitudes, provocan que se obtengan bien sistemas muy precisos o bien sistemas muy interpretables. La aparición en las últimas décadas de los Algoritmos Evolutivos Multiobjetivo (AEMOs) y más concretamente los AGMOs [Deb01, PR10, WFP14, IMTN14] ha provocado que se popularice y extienda su uso en el aprendizaje automático de SBRDs, al permitir esta técnica obtener un conjunto de soluciones no dominadas en el que cada solución del conjunto representa un equilibrio entre los objetivos considerados. De esta forma, en la última década, a fin de considerar más de un objetivo en lugar de uno solo en el proceso de diseño y optimización, los SDEs se han extendido a un enfoque multiobjetivo mediante el uso de los citados AEMOs, dando lugar a los SDEMOs [FAN+13].

1.2. Taxonomía de los Sistemas Difusos Evolutivos

Como hemos comentado anteriormente, desde el punto de vista de la optimización, encontrar un modelo difuso apropiado es equivalente a codificar éste como una estructura de parámetros y encontrar los valores de los mismos que nos den el óptimo para una determinada función de aptitud (fitness). Los parámetros del modelo proporcionan el espacio de búsqueda que se transforma de acuerdo a una representación genética.

De este modo, el primer paso en el diseño de un SDE es decidir qué partes del sistema difuso van a estar sujetos a la optimización por parte del AE, codificando estas partes en sus cromosomas, y cómo se va a realizar dicha optimización (mediante un aprendizaje o mediante un ajuste de dichos componentes).

Por otro lado se debe decidir qué tipo de AE se va a utilizar, monoobjetivo o multiobjetivo, en función de si sólo se pretender mejorar la precisión del sistema o, por el contrario, se debe conseguir un equilibrio entre la precisión y la interpretabilidad (y/o otros posibles objetivos).

Por último hay que decidir qué tipo de representación se ha de utilizar para los conjuntos difusos, teniendo en cuenta que en los últimos años se han diseñado nuevas representaciones para ellos.

Por tanto, los SDEs se pueden clasificar atendiendo a diferentes criterios, en función de si se considera qué partes del SBRDs se aprenden y/u optimizan [Her08, Cor11], qué tipo de criterio de optimización multiobjetivo se utiliza en función del problema multiobjetivo abordado [FAN+13] o qué tipo de representación se utiliza para los conjuntos difusos. Todos estos criterios de clasificación vienen descritos en la taxonomía realizada por [FLdJH15] mostrada en la Figura 1.2 que será descrita en las siguientes subsecciones. En primer lugar, se clasificarán los modelos de acuerdo a los componentes del SBRD involucrados en el proceso de aprendizaje evolutivo (Sección 1.2.1), para posteriormente centrarnos en la optimización multi-objetivo (Sección 1.2.2). Finalmente, comentamos de forma breve la construcción parametrizada para nuevas representaciones difusas (Sección 1.2.3).

1.2.1. Aprendizaje y Ajuste Evolutivo de los Componentes de los SBRDs

Como se ha indicado anteriomente, para modelar un problema mediante conjuntos difusos debemos decidir si se desea crear un SBRD (mediante el diseño de métodos para aprender los componentes de la BC y/o del MI), o bien mejorar uno ya existe (desarrollando enfoques que ajusten sus componentes). Por lo tanto, el primer factor que debe tenerse en cuenta para clasificar los SDEs según este criterio es la existencia o no de una BC previa (incluyendo la BD y la BR). De esta forma, los SDEs se pueden dividir entre aquellos que realizan un ajuste evolutivo por existir una BC previa y aquellos que realizan un aprendizaje evolutivo de la BC o un aprendizaje evolutivo de los componentes de la BC junto con los parámetros del MI. Estas tres propuestas que se describen a continuación, se pueden realizar bien a través de un enfoque mono-objetivo estándar o bien mediante multiobjetivo.

1.2.1.1. Aprendizaje Evolutivo de la BC

Esta categoría engloba a todas las propuestas que realizan el aprendizaje sobre la BR, sobre la BD, o bien sobre todos los componentes de la BC simultáneamente. Las siguientes cuatro posibilidades de aprendizaje de la BC pueden ser consideradas:

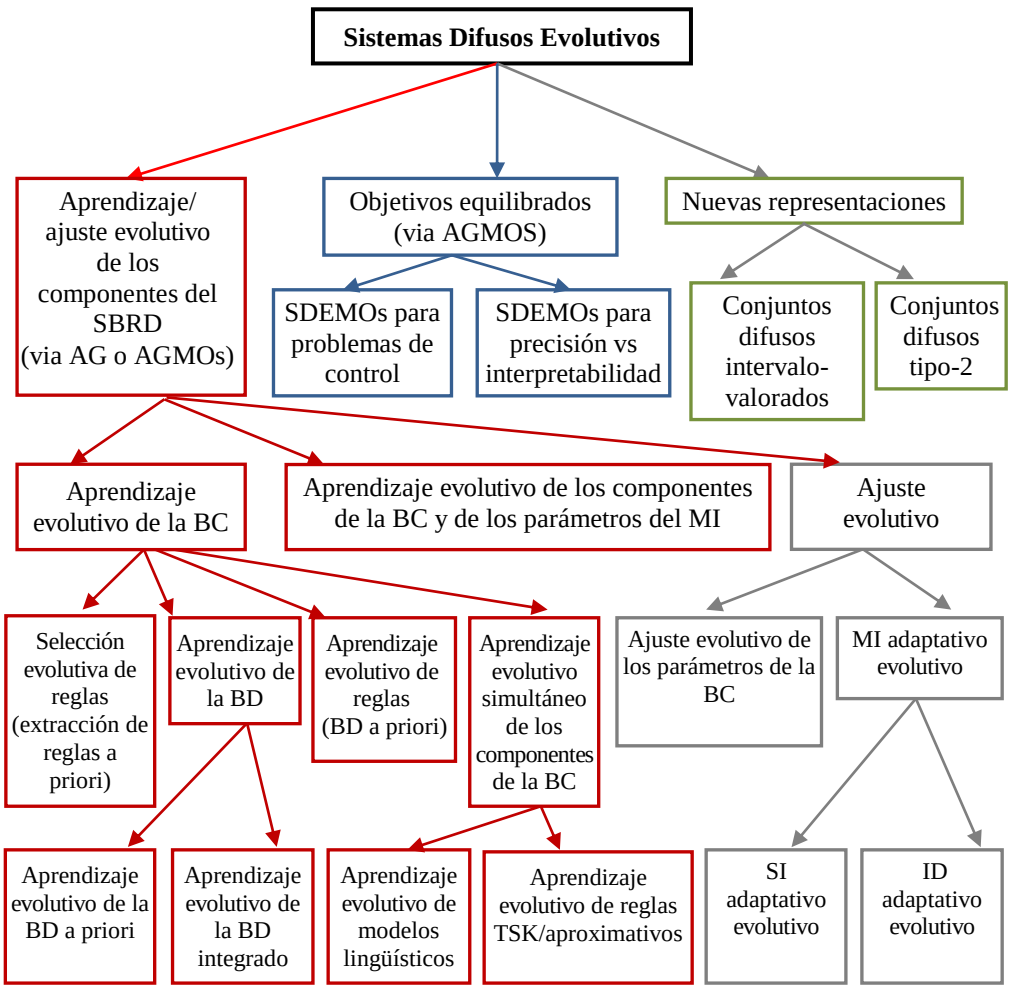


Figura 1.2: Taxonomía de los SDEs

Aprendizaje Evolutivo de Reglas. La mayoría de los métodos propuestos para aprender de forma automática la BC a partir de información numérica se centran en el aprendizaje de la BR utilizando una BD predefinida [Bon96, GP99, INM99]. Para predefinir la BD lo habitual es escoger un número impar de términos lingüísticos para cada variable lingüística (normalmente el mismo para todas las variables) y realizar una distribución uniforme de dichos términos en el universo de discurso de dichas variables. La propuesta pionera

de este enfoque se puede encontrar en [Thr91]. En [dJGHM07, KA05] podemos encontrar otras propuestas que se centran en la extracción de algunas reglas descriptivas en problemas de minería de datos (reglas de asociación, descubrimiento de subgrupos, etc.)

1. *Selección Evolutiva de Reglas*. Una BR con un excesivo número de reglas es poco interpretable y puede contener reglas irrelevantes, redundantes, erróneas y/o conflictivas, que perturban el funcionamiento del SBRD. Para eliminar esas reglas se puede utilizar un proceso evolutivo de selección de reglas y así obtener un subconjunto optimizado a partir del conjunto original [AGHAF07, GAH10, HLV98a, IY04]. En [INYT95] podemos encontrar la primera contribución en este área y en [IMT97] la primera mediante selección evolutiva de reglas multiobjetivo.

2. *Aprendizaje Evolutivo de la BD*. Implica el aprendizaje de la forma de las funciones de pertenencia y otros componentes de la BD como las funciones de escala o la granularidad de las particiones difusas (aprendizaje evolutivo de la BD a priori) [BLMS09]. Otro opción es integrar el aprendizaje de la BR en el aprendizaje evolutivo de la BD, de forma que cada vez que se aprenda una BD en una iteración del algoritmo, se use un algoritmo para derivar la BR y se evalúe la BC completa así obtenida (aprendizaje evolutivo de la BD embebido) [CHV01].

3. *Aprendizaje Evolutivo simultáneo de los componentes de la BC*. Otras propuestas tratan de aprender los dos componentes de los BC (BR y BR) de forma simultánea [BAB01, CH97, CH01, MMH97, MV97, PKL94]. Al trabajar de esta manera, se tiene la posibilidad de generar una BC de mejor calidad, pero como contrapartida se trabaja con un espacio de búsqueda más grande que hace que el proceso de aprendizaje sea más complicado y lento. En [HM95] se puede encontrar un trabajo que es una referencia de este tipo de proceso de aprendizaje.

1.2.1.2. Aprendizaje Evolutivo de los Componentes de la BC y de los Parámetros del MI (Modelos Híbridos)

Esta categoría engloba las propuestas que son un modelo híbrido entre el MI adaptativo y el aprendizaje de los componentes de la BC. Este tipo de enfoque trata de encontrar un nivel alto de colaboración entre el MI a través de

su parametrización y el aprendizaje de los componentes de la BC, incluyendo ambos en un proceso de aprendizaje simultáneo. En [MPH07] podemos encontrar una propuesta que combina el diseño de la BC con elementos del MI, para aprender de forma conjunta la BR lingüística y operadores paramétricos de la inferencia y defuzzificación en un solo paso.

1.2.1.3. Ajuste Evolutivo

Si ya existe una BC previa, podemos mejorar el funcionamiento del SBRD, aplicando un proceso de ajuste evolutivo a posteriori para mejorar la definición preliminar de la BD o los parámetros del SI sin realizar ningún cambio sobre la BR existente previamente. Por tanto, de acuerdo con los componentes del SBRD que se desee ajustar tenemos tres posibilidades de adaptación:

1. *Ajuste Evolutivo de los parámetros de la BC*. Se realiza un proceso de ajuste a posteriori considerando toda la BC obtenida para ajustar los parámetros de la función de pertenencia, sin modificar la BR existente. El proceso de ajuste puede modificar la forma de las funciones de pertenencia [AAFGH07, GAH10, AGH11], el número de términos lingüísticos [CCdJH05] y los parámetros de las funciones de pertenencia [GAH09, ANHI11]. El citado ajuste de los parámetros de las funciones de pertenencia consiste en alterar los valores de los distintos parámetros que las definen para realizar desplazamientos y/o ensanchamientos de los conjuntos difusos. Esto se puede conseguir bien ajustando directamente cada uno de los parámetros, usando diferentes factores de escala lineales o aplicando desplazamientos laterales y/o de amplitud. Por ejemplo, si consideramos la siguiente función de pertenencia triangular:

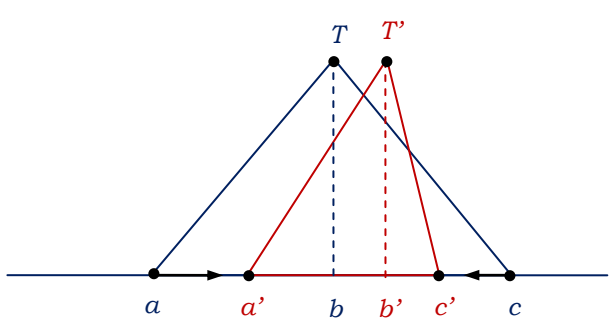


Figura 1.3: Ajuste de las funciones de pertenencia

alterar los parámetros *a*, *b* y *c* supone variar la forma del conjunto difuso asociado a la función de pertenencia, afectando así al comportamiento del SBRD. En [Kar91] podemos encontrar la primera propuesta en este área.

2. *Sistemas de Inferencia Adaptativos Evolutivos*. Este enfoque usa expresiones parametrizadas en el SI (se denomina SI adaptativo), para conseguir una mayor cooperación entre las reglas difusas y por tanto modelos más precisos, sin perder interpretabilidad las reglas lingüísticas. En [AFHMP07, CBFO06, CBM07], se pueden encontrar propuestas en este área centradas en regresión y clasificación.

3. *Métodos de Defuzzificación Adaptativos Evolutivos*. Este enfoque usa parámetros en la ID a fin de mejorar la precisión del sistema. La técnica más utilizada, debido a su buen comportamiento, eficiencia y fácil implementación, consiste en aplicar primero la función de defuzzificación a todo el conjunto de reglas inferido (para conseguir un valor característico de cada regla) y posteriormente calcular el valor final mediante un operador de media ponderada aplicado sobre cada uno de dichos valores [CHMP04, KCL02].

1.2.2. Enfoques para la Optimización de varios Objetivos

Tradicionalmente, los esfuerzos en el desarrollo de SDEs han estado dirigidos a la mejora de la exactitud/precisión de los SBRDs de una manera única. Sin embargo en la mayoría de los problemas del mundo real se necesitan optimizar más de un objetivo de forma simultánea. En orden a considerar más de un objetivo durante el proceso de diseño u optimización, los SDEs han sido extendidos a un enfoque multiobjetivo dando lugar , como se indicó anteriormente, a los SDEMOs [FAN+13].

En los últimos años han aparecido multitud de trabajos relacionados con los SDEMOs en los que se propone su uso para una amplia gama de tipos de problemas diferentes. Para clasificar estos trabajos realizamos una división de este tipo de técnicas en dos niveles, basándonos en primer lugar en el tipo de problema multiobjetivo abordado, es decir, el tipo de objetivo optimizado, y en segundo lugar en el tipo de componentes del SBRDs optimizado durante el proceso evolutivo. Ambos criterios afectan al tipo y a la complejidad del espacio de búsqueda y por tanto a la forma en la que se aplican los SDEMOs.

Como podemos observar, el segundo nivel presenta una clara correspondencia con los tipos descritos anteriormente para SDEs en el apartado anterior, razón por la cual aquí solo presentaremos una breve descripción de cada categoría (una descripción más detallada de esta clasificación puede consultarse en [FAN+13] o en su página web asociada (http://sci2s.ugr.es/moefs-review/)).

Por tanto, centrándonos en la naturaleza del problema multiobjetivo abordado, podemos distinguir dos categorías principales: la primera incluye aquellas contribuciones en los que los SDEMOs son aplicados a los problemas de control multiobjetivo y la segunda aquellas en las que los SDEMOs son diseñados para generar SBRDs que presentan distintos niveles de equilibrio entre precisión e interpretabilidad.

1. *Problemas de Control.* La calidad y rendimiento de un sistema de control depende de su precisión. En el diseño de un controlador encontramos un par de problemas. El primero aparece si los procesos se describen de forma imprecisa. Por otro lado, encontramos el problema de cómo diseñar modelos adaptables, es decir, que sean capaces de activar un proceso de aprendizaje y ajuste cuando los parámetros del sistema real cambian. Además en el diseño de un sistema de control a menudo es necesario optimizar múltiples objetivos, como por ejemplo restricciones de tiempo o requisitos de robustez y estabilidad, que pueden estar en conflicto entre ellos. Todas estas consideraciones han conducido a la aplicación de los SDEMOs en el diseño de Controladores basados en Lógica Difusa.

2. *Equilibrio entre Precisión e Interpretabilidad.* El interés de los investigadores en la obtención de modelos lingüísticos más interpretables [GAH11] ha crecido de manera significativa en los últimos años. El problema es que la precisión y la interpretabilidad representan objetivos generalmente contradictorios, por lo que una buena forma de afrontar este problema es usando AGMOs [CLV07] ya que conducen a un conjunto de modelos difusos con diferentes equilibrios entre ambos objetivos en lugar de a una única solución. Esto ha dado lugar a que se extienda el uso de los SDEMOs para afrontar este tipo de problema. Esta categoría es la más numerosa y la que más contribuciones presenta, al ser la interpretabilidad uno de las cualidades más importantes de los SBRDs.

Cabe señalar que la mayoría de las contribuciones en este grupo utilizan un modelo difuso lingüístico, a priori el más interpretable, sin embargo hay también un número limitado de trabajos que afrontan la interpretabilidad incluso en los SBRDs de tipo TSK [JGSRB01, WKJ+05a, WKJ+05b].

De estas dos categorías presentados nosotros estamos especialmente interesados en la segunda por lo que la abordaremos más detenidamente en las siguientes subsecciones.

1.2.2.1. Precisión e Interpretabilidad de los Sistemas Difusos Lingüísticos Evolutivos

El modelado de los SDEMOs está caracterizado principalmente por dos propiedades, que permiten asegurar la calidad del modelo difuso obtenido:

Precisión: Es la capacidad de representar fielmente un sistema real. Un sistema será mejor cuanto mayor similaridad exista entre la respuesta del sistema real y el modelo difuso.

Interpretabilidad: Es la capacidad para expresar el comportamiento de un sistema real de una manera comprensible. Un sistema será mejor cuanto más simple y más fácil de entender sea para los humanos.

Mientras que la definición de la precisión en una determinada aplicación es sencilla y las medidas para medirlas están claras y perfectamente establecidas, siendo lo habitual usar el Error Cuadrático Medio (ECM) para tal fin, la definición de la interpretabilidad es más compleja, al existir múltiples aspectos que pueden ser tenidos en cuenta para medirla [ADLM11, AMGR09, GAH10].

La interpretabilidad es una propiedad subjetiva difícil de definir que depende de muchos factores, algunos relacionados con la complejidad del modelo y otros relacionados con su semántica, tales como el número de reglas, el número de características (variables de entrada), el número de términos lingüísticos o granularidad de las etiquetas, la forma de los conjuntos difusos de las etiquetas (funciones de pertenencia), pero también de criterios tan variados como la distinguibilidad, la normalidad o la completitud, número de reglas que se activan simultáneamente con cada ejemplo, etc.

Debido a su subjetividad, el problema principal es encontrar una definición universalmente reconocida y una manera objetiva de medir esta característica. En la última década varios trabajos han analizado el problema de la interpretabilidad [Cor11] en busca de medidas que podrían ser universalmente aceptadas por la comunidad científica. En la literatura especializada, podemos encontrar diferentes medidas propuestas [AMGR09, BLMS09, GAH10, GAH11, MF08, Nau03, ZG08] y técnicas [AAFHO07, GAH09, GH03, GH04, GRP+07, IM96, IMT95, INYT95] para obtener modelos difusos lingüísticos más interpretables. Debido a la gran cantidad de medidas diferentes que pueden ser tenidas en cuenta para medir la interpretabilidad, en los últimos años se han propuesto diferentes taxonomías [AMGR09, GAH11, MF08, ZG08] que resultan interesantes para ayudar a entender mejor la interpretabilidad dentro del área de los sistemas difusos.

Así por ejemplo, Zhou y Gan [ZG08] proponen una taxonomía en términos de interpretabilidad de bajo nivel y de alto nivel.

- *Interpretabilidad de bajo nivel*: se alcanza a nivel de conjuntos difusos mediante la optimización de las funciones de pertenencia en términos de criterios semánticos sobre dichas funciones (interpretabilidad basada en la semántica), tales como distinguibilidad, normalidad, cobertura y complementariedad.

- *Interpretabilidad de alto nivel*: se obtiene a nivel de reglas difusas mediante la reducción de la complejidad global en términos de algún criterio, tales como un moderado número de variables y reglas (interpretabilidad basada en la complejidad), completitud y consistencia de las reglas (interpretabilidad basada en la semántica).

Mientras que Alonso y otros [AMGR09] proponen una taxonomía en términos de interpretabilidad descriptiva y explicativa:

- *Interpretabilidad descriptiva o Global* (legibilidad de la estructura del sistema o complejidad del modelo): El sistema es visto como un todo que describe su comportamiento global (generalmente medida como número de reglas, variables, etiquetas por variable, etc.)

- *Interpretabilidad explicativa o Local* (comprensión del modelo): Una medida semántica de la interpretabilidad a nivel individual (SI y método de defuzzificación utilizado, operador de conjunción, reglas activadas simultáneamente, reglas ponderadas, etc.)

Por otro lado, Gacto y otros [GAH11] proponen otra clasificación basada en un doble eje (complejidad frente a interpretabilidad semántica y BR frente a particionamiento difuso) dando lugar a la aparición de la siguiente taxonomía:

- *Complejidad a nivel de BR* (número de reglas y de variables en cada regla)

- *Complejidad a nivel de Partición Difusa o BD* (número de funciones de pertenencia o granularidad de las etiquetas y de variables)

- *Interpretabilidad semántica a nivel de BR* (consistencia de las reglas, reglas disparadas al mismo tiempo, estructura de reglas)

- *Interpretabilidad semántica a nivel de Partición Difusa o BD* (completitud, normalidad, distinguibilidad y complementariedad)

Con independencia de las taxonomías anteriores y aunque no hay una medida de interpretabilidad universalmente reconocida, en la actualidad los investigadores concuerdan en considerar dos tipos de medidas de interpretabilidad, unas basados en la complejidad y otras basadas en la semántica:

- *Medidas basadas en la complejidad*: Este enfoque engloba las medidas que se utilizan para disminuir la complejidad del modelo difuso obtenido (número de reglas, número de antecedentes o variables en una regla, número de etiquetas por regla, etc.).

- *Medidas de interpretabilidad semántica*: Este enfoque engloba las medidas que se utilizan para preservar la semántica asociada con las funciones de pertenencia (distinguibilidad, cobertura, etc.) y reglas (consistencia, etc.)

Generalmente al evaluar la interpretabilidad de un modelo difuso, los índices se centran en el primer tipo de medidas, al ser más claras y fáciles de cuantificar que las segundas. Por esta razón en esta memoria nos centraremos en las medidas de complejidad para medir la interpretabilidad.

Un estudio completo sobre las medidas de interpretabilidad de SBRDs lingüística puede verse en [GAH11].

1.2.2.2. El Enfoque Multiobjetivo para Mejorar el Equilibrio entre Precisión e Interpretabilidad

La mayoría de los problemas de optimización del mundo real son de naturaleza multiobjetivo, esto es, suelen tener dos o más funciones objetivo que deben satisfacerse simultáneamente y que posiblemente están en conflicto entre sí. La presencia de múltiples objetivos en un problema provoca que no exista una solución óptima única, sino que exista un conjunto de soluciones óptimas (conocidas como soluciones Pareto óptimas), mayores cuantos más objetivos haya, debido a que ninguna de ellas domina a las demás en todas las funciones objetivo a optimizar. Por ello, a falta de cualquier otra información adicional, ninguna de estas soluciones puede ser considerada mejor que otra por lo que en principio interesa encontrar tantas soluciones como sea posible.

Los AGMO permiten obtener un frente de Pareto con una variedad de soluciones para cada uno de los objetivos considerados en una simple ejecución del algoritmo, al tratar simultáneamente un conjunto de posibles soluciones, llamadas población. Esto permite elegir una solución de compromiso entre los distintos objetivos, algo que es muy útil para el caso de objetivos contradictorios, como en el caso del equilibrio entre interpretabilidad y precisión.

Aunque todas las soluciones generadas por el AGMO son igualmente válidas al no dominar ninguna al resto, cada solución del frente del Pareto tiende a satisfacer un objetivo de forma más óptima que el resto. Cuando los objetivos son la precisión y la interpretabilidad cada solución en la frontera del Pareto representa un diferente equilibrio entre ambas magnitudes. La Figura 1.4 ilustra esta idea, en la que cada elipse representa un Sistema Difuso solución.

El problema de la mejora de la precisión, manteniendo o mejorando la interpretabilidad del modelo difuso, se abordó por primera vez a mediados de 1990 [IMT97], integrando la interpretabilidad en el proceso de optimización, gracias a la aplicación de AGMOs a sistemas difusos. Desde entonces, la interpretabilidad ha adquirido una importancia creciente aumentando de manera muy significativa el interés de los investigadores en la obtención de modelos lingüísticos más interpretables a la vez que precisos [GAH11].

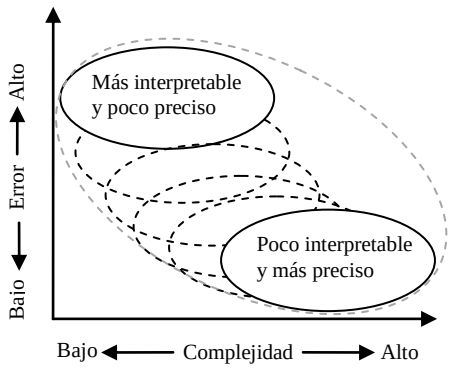


Figura 1.4: Balance entre precisión e interpretabilidad

Este hecho ha provocado que en los últimos años se haya extendido el uso de los SDEMOs como herramienta idónea para mejorar el mencionado equilibrio entre ambas magnitudes, mejorando la precisión de éstos y tratando de evitar el descenso de la interpretabilidad del sistema [ADLM09, ANHI11, BLMS09, CHP+07, Cor11, INN11]. Algunas de las propuestas obtienen el Pareto completo seleccionando [INM01, IY04] o aprendiendo [ADH+09, CDLM07, GAH10, IN07] el conjunto de reglas que mejor representan los datos de ejemplo, es decir, mejorando la precisión del sistema y disminuyendo la complejidad (número de reglas) de la BR Difusa. Otros [AGHAF07, GAH09, PK10] también proponen ajustar las funciones de pertenencia junto con la selección de reglas para obtener modelos difusos lingüísticos más simples y precisos. El trabajo pionero fue propuesto por Ishibuchi y otros en [IMT95].

Aunque hay una gran variedad de AGMOs propuestos para implementar los SDEMOs, la mayoría de los investigadores coinciden en afirmar que, de todos ellos, los AGMOs SPEA2 y NSGA-II (véase Apéndice A para una descripción detallada de ambos algoritmos) pueden ser considerados como los más representativos, ya que permite obtener un conjunto de soluciones del Pareto más amplio en una gran variedad de problemas, propiedad muy apreciada en este marco de trabajo.

La mayoría de los trabajos sobre este tema sólo consideran medidas cuantitativas de la complejidad del sistema para mejorar la interpretabilidad de tales sistemas, considerando raramente medidas cualitativas, y en casi todos ellos, los AGMOs desarrollados en dichas propuestas son ligeras modificaciones de AGMOs propuestos para uso general (SPEA2, NSGA-II, etc.) o están basados en los mismos. Por otro lado, la mayor parte de las propuestas mencionadas han sido aplicadas a problemas de clasificación, y

sólo una minoría han sido aplicadas a problemas de regresión [IY03, CDLM07, AGHAF07, GRP+07, GAH09, APLM09].

1.2.2.3. Equilibrio entre Precisión e Interpretabilidad en el aprendizaje del Mecanismo de Inferencia

En las secciones anteriores hemos visto diferentes formas de optimizar y mejorar la precisión e interpretabilidad de los SBRDs lingüísticos, haciendo uso tanto de técnicas de aprendizaje como de ajuste evolutivo multiobjetivo. Para conseguir este objetivo, la mayoría de las propuestas existentes en la literatura científica se centran en el aprendizaje o ajuste de los componentes de la BC (BR, BD o ambas de forma simultánea) al ser el elemento fundamental que define el comportamiento del sistema. No obstante, la mejora del equilibrio entre precisión e interpretabilidad también puede realizarse mediante la parametrización de los operadores del MI, empleando expresiones adaptativas en el SI, en la ID o en ambos conjuntamente.

Para una descripción más detallada de los operadores que intervienen en el SI y en la ID, y una explicación de la influencia y alcance que los parámetros tienen en el MI puede consultarse la Sección D.1.4 del Apéndice D.

La parametrización del SI y la ID han adquirido una mayor importancia [AFHMP07, CHMP04, MPH07] ya que, a diferencia de las otras alternativas, permiten encontrar una forma de defuzzificar la contribución de cada regla adecuadamente [CHMP04] y mejorar la cooperación entre las reglas [AFHMP07] conservando la BC original. Además dichas metodologías son complementarias a las basadas en las mejoras de la BC, de forma que se puede aprender la BR y los ODAs conjuntamente [MPH07], siendo muy interesantes en este sentido, ya que la sinergia obtenida por la combinación de ambas puede ayudar a lograr mayores niveles de interpretabilidad y precisión.

En este sentido, el uso y aprendizaje (o ajuste) de los parámetros del SI y de la ID es una alternativa aún poco explotada para mejorar la precisión e interpretabilidad de los SBRDs lingüísticos, por lo que resultan un campo de investigación muy interesantes. Por esta razón, en esta memoria centraremos nuestra atención en el desarrollo de nuevos operadores adaptativos en el ámbito de la inferencia y la defuzzificación, con el fin de diseñar SDEMOs que sean capaces de obtener SBRDs lingüísticos robustos, interpretables y a la vez precisos.

1.2.3. Nuevas Representaciones Difusas

La mayoría de SBRDs hacen uso de los conjuntos difusos estándar definidos por [Zad65], también llamados conjuntos difusos tipo-1, pero en la literatura especializada encontramos extensiones de este enfoque que permiten representar mejor la incertidumbre inherente a la lógica difusa. Dos de los principales exponentes de las nuevas representaciones difusas son los Conjuntos Difusos Tipo-2 [KML99] y los Conjuntos Difusos Intervalo-Valorados (CDIVs) [Sam75].

Los conjuntos difusos tipo-2, que fueron originalmente propuestos por Zadeh en 1975 como una generalización del concepto de conjunto difuso ordinario, también denominado conjunto difuso tipo-1, reducen la cantidad de incertidumbre en un sistema, porque su lógica ofrece mejores capacidades para manejar las incertidumbres lingüísticas modelando la vaguedad e incertidumbre de la información. Los conjuntos difusos tipo-2 son esencialmente conjuntos "difusos-difusos" en los que los grados de pertenencia son conjuntos difusos tipo-1 al existir incertidumbre acerca del punto donde cortan a la etiqueta, es decir, incorporan incertidumbre sobre la función de pertenencia, de forma que, para un valor específico, la función de pertenencia, adquiere diferentes valores, que no ponderan lo mismo. Por lo tanto, podemos asignar grados de pertenencia a todos esos puntos. Los conjuntos difusos tipo-2 son interesantes cuando es difícil determinar una función de pertenencia exacta para un conjunto difuso.

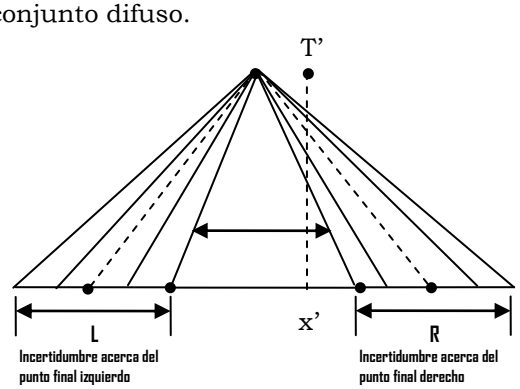


Figura 1.5: Conjunto difuso tipo-2

Para los CDIV [Sam75], el grado de pertenencia de cada elemento al conjunto viene dado por una subintervalo cerrado del intervalo [0, 1], es decir, se representa mediante un intervalo de valores en lugar de un solo valor. De este modo, esta amplitud representará la falta de conocimiento del experto para dar un valor numérico exacto para el miembro. Debemos señalar que CDIVs es un caso particular de conjuntos difusos tipo-2, en donde el grado de pertenencia de cada elemento al conjunto es así mismo un conjunto difuso en [0, 1].

Al igual que con los conjuntos borrosos estándar, los AEs se han utilizado para encontrar los valores apropiados de los parámetros y la estructura de estos sistemas difusos. En el caso de los modelos difusos tipo-2, los SDEs se puede clasificar en dos categorías [CM12]: (1) la primera categoría supone que un modelo difuso "óptimo" tipo-1 ya ha sido diseñado, y después un modelo difuso tipo-2 se construye a través de algún aumento de ruido del modelo existente [CMG+11]; (2) la segunda categoría engloba aquellos métodos de diseño que se ocupan de la construcción del modelo difuso de tipo-2 directamente a partir de datos experimentales [WH07].

En cuanto a los CDIV, los trabajos actuales inicializan los conjuntos difusos de tipo-1 de forma homogénea sobre el espacio de entrada. A continuación, los límites superior e inferior del intervalo para cada conjunto difuso se aprenden por medio de una función de débil ignorancia (ajuste de la amplitud) [SFBH10], que también puede implicar un ajuste lateral para una mejor contextualización de las variables difusas [SFBH11]. Por otro lado, en [SFBH13] los CDIV son construidos específicamente, utilizando una función de equivalencia restringida intervalo-valorada dentro de un nuevo método de razonamiento difuso intervalo-valorado. Los parámetros de estas funciones de equivalencia por variable se aprenden por medio de un AE, que también se combina con selección de reglas con el fin de disminuir la complejidad del sistema.

De entre todas las propuestas de SDEs descritas en la taxonomía anterior, en esta memoria estamos interesados en aquellas que tratan de obtener modelos difusos más interpretables y a la vez precisos mediante el uso de SDEMOs que utilizan un modelo difuso lingüístico (Mamdani) para especificar las reglas del SBRD.

1.4. Nuevas Tendencias y Retos en los Sistemas Difusos Lingüísticos Multiobjetivo

A pesar del buen comportamiento y del éxito mostrado en muchas áreas de Minería de Datos, el comportamiento de las SDEs se puede mejorar aún más si se aprovechan las nuevas características y componentes que se han propuesto recientemente para los modelos evolutivos, tanto en términos de rendimiento como de escalabilidad.

En esta sección se presentan las nuevas tendencias y desafíos en el campo de los SDEs y SDEMOs y se describen algunas recientes contribuciones relacionados con ellas. Además se plantean algunas cuestiones y líneas de trabajo futuro para dirigir la atención de los investigadores hacia los nuevos problemas y temas prolíficos para el desarrollo de sus nuevas metodologías.

Considerando estos asuntos y el estado-del-arte descrito en las secciones anteriores, hemos evidenciado los siguientes temas relacionados con los SDEMOs en los cuales todavía se puede investigar.

1.4.1. Escalabilidad de los AEs

Uno de los retos más importantes hoy en día en Aprendizaje Automático y Minería de Datos [CFT13] es el problema del aumento de la escalabilidad. Esencialmente, la escalabilidad define la capacidad de un algoritmo para mantener un nivel adecuado de rendimiento, a un coste computacional razonable, cuando el tamaño del problema aumenta. En concreto, podemos distinguir dos tipos de problemas que deben ser resueltos mediante la aplicación de diferentes tipos de técnicas: (1) problemas de alta dimensionalidad, es decir, problemas en los que un gran número de variables (características) tienen que ser consideradas y, (2) problemas de gran escalabilidad, es decir, problemas que constan de un gran volumen de datos.

Ambos tipos de problemas tienen una fuerte influencia en el modelado difuso lingüístico. Por un lado, cuando el número de variables y/o ejemplos es alto [CA98, Jin00] se produce un crecimiento exponencial del espacio de búsqueda de las reglas difusas, lo que provoca que los algoritmos de obtención de reglas usualmente utilizados en los SBRDs generen un número desmesurado de reglas, dando lugar a SBRDs poco interpretables, con muchas

reglas y/o con reglas de mucha longitud. Por otro lado, se produce un aumento considerable del tiempo de entrenamiento debido a la ingente cantidad de datos que hay tratar y un aumento de la convergencia hacia modelos compactos e interpretables durante el proceso de aprendizaje [Her08].

El hándicap en este caso es que los AEs son computacionalmente costosos por sí mismos. Por lo general, los AEs necesitan un gran número de evaluaciones para alcanzar la convergencia, y como el tamaño del conjunto de datos afecta a la función de aptitud, cuando usamos un gran volumen de datos el tiempo de cálculo de dicha función aumenta considerablemente y por consiguiente también aumenta en la misma medida el tiempo para conseguir la convergencia. Si encima además el conjunto de datos posee muchas variables el espacio de búsqueda crece de forma exponencial agravándose aún más la situación.

Para las tareas de Minería de Datos en general, y para SDEs en particular, hay varias técnicas disponibles que mejoran el rendimiento de los algoritmos tradicionales en estas situaciones [GPdHG12]. La mayoría de ellos se pueden clasificar en tres perspectivas:

1. *Algoritmos orientados* [PK99]. Incluye aquellas técnicas que modifican la definición del algoritmo adaptando sus componentes, o integrando cualquier mecanismo orientado hacia la dimensionalidad problema para mejorar su rendimiento cuando se aborda datos a gran escala [AGH11, BC13, GGAH14].

2. *Datos orientados* [GLH14]. Este tipo de técnicas se aplican directamente a los datos, reduciéndolos y distribuyéndolos con el objetivo de reducir el coste computacional de las técnicas de Aprendizaje Automático tradicionales. Esto puede llevarse a cabo por medio de procesos de selección de características [BBK09], selección de instancias o prototipos [GPdHP13], o mediante la modificación de la distribución de ejemplos [dHGGP09]. Estos enfoques de preprocesamiento pueden llevarse a cabo como parte del objetivo cuando nuestro propósito es mejorar la calidad de los datos antes de la etapa de entrenamiento.

3. *Enfoques distribuidos* [AT02]. En este caso, la metodología es desarrollada para ejecutarse en varias máquinas con el fin de ahorrar tiempo computacional [RABH09] utilizando AEs distribuidos que se ejecutan en paralelo dividiendo el conjunto de datos en subconjuntos que se aprenden cada uno en una máquina de forma distribuida.

Mientras que el primer tipo de solución requiere un diseño riguroso y completo de las características internas del AE para lograr una mejor eficiencia, los dos últimos tipos pueden ser vistos como "enfoques externos" que son de algún modo independientes de la etapa de aprendizaje del SDE.

Independientemente de la técnica utilizada, los investigadores deben desarrollar nuevas metodologías que permitan disminuir la complejidad de tiempo de cálculo en este tipo de algoritmos.

1.4.2. Selección de Instancias mediante SDEs

Como comentamos antes, uno de los principales inconvenientes de los AEs y consecuentemente de los SDEs, es el tiempo computacional consumido durante la evaluación de la función de aptitud (fitness). En este sentido, el tamaño del conjunto de entrenamiento no sólo influye en el hecho anterior, sino que también puede afectar a la complejidad del modelo resultante, ya que un mayor número de casos generalmente induce a la generación de un SBRD con un mayor número de reglas, lo que decrementa la interpretabilidad del modelo obtenido. La manera de superar ambos problemas, al menos en parte, es llevar a cabo una etapa de pre-procesamiento por medio de un método de selección de ejemplos del conjunto de entrenamiento a fin de reducir el tamaño del conjunto de datos antes de aplicar el proceso de aprendizaje. El objetivo es tratar de mantener la precisión global del modelo reduciendo el tiempo necesario para su obtención así como la complejidad del modelo obtenido usando un conjunto representativo de ejemplos del conjunto de datos inicial.

En [FGA+13], los autores presentan un análisis de la influencia y efectividad de diferentes técnicas de selección de instancias en SDE considerando 36 métodos diferentes del estado del arte y determinan que la técnica de selección de instancia a utilizar depende básicamente de la dimensión del conjunto de datos considerado, es decir, en función del tamaño del conjunto se deben usar unas técnicas u otras. Un enfoque muy diferente puede encontrarse en [ADM12], donde los autores incorporan la etapa de selección instancia dentro de todo el proceso de aprendizaje del SDE, lo que lleva a un enfoque co-evolutivo. En concreto, hacen uso de un SDEMO para aprender el SBRD, y de un AG estándar para obtener un conjunto de entrenamiento reducido, demostrando que utilizando entre un 10% y 20% de los datos originales, se puede obtener un frente de Pareto comparable al

obtenido con el conjunto de datos completo, reduciendo el tiempo de ejecución hasta un 86%.

De acuerdo con las conclusiones extraídas en estos artículos, deben llevarse a cabo nuevos trabajos siguiendo esta línea de investigación con el objetivo de desarrollar una metodología que permite una mayor integración entre la selección de instancias y el aprendizaje del SDE, de manera que las nuevas técnicas pueden ser más adecuados para los propósitos de escalabilidad, como se sugiere en el apartado anterior.

1.4.3. Datos Masivos (Big Data)

El término Big Data o Datos masivos es un concepto que hace referencia a los desafíos y ventajas derivados de la recolección y procesamiento de grandes cantidades de datos [FdRL+14]. Este término hace referencia a una cantidad de datos tal que supera la capacidad del software habitual para ser capturados, gestionados y procesados en un tiempo razonable y se identifica principalmente por la definición de las 3V que incluye un gran volumen de información, que llega a una alta tasa de velocidad y tal vez con requisitos de tiempo real, y con una variedad de formas o representaciones diferentes (estructurada, semi-estructurada o no estructurada). Definiciones adicionales incluyendo hasta 9V se puede también conocer, añadiendo términos como veracidad, valor, viabilidad, y visualización, entre otros [ZEdR+11].

Hay varias herramientas que han sido diseñados para hacer frente a los problemas de Big Data. Entre ellos, la más popular es el sistema de procesamiento distribuido MapReduce, así como su homólogo de código abierto Hadoop [Lam11]. Ambos están diseñados para ejecutar trabajos por lotes y resultan muy adecuados para realizar grandes procesamientos de datos distribuidos donde un rendimiento rápido no es un factor crítico Estos modelos permiten que el sistema automáticamente paralelice la ejecución, simplemente definiendo dos funciones, denominadas Map y Reduce. Map se utiliza para el cálculo por registro, mientras que Reduce agrega la salida de las funciones Map y aplica una función determinada para la obtención de los resultados finales. Desde sus inicios hace una década, la implementación de MapReduce se ha convertido en el sistema más popular para el almacenamiento, gestión y tratamiento de grandes volúmenes de datos de forma masiva.

Además de estos sistemas, recientemente ha aparecido Spark [KKWZ14], una plataforma emergente desarrollada para mejorar la reutilización de datos a través de múltiples cálculos, que a diferencia de Hadoop almacena los datos en memoria en lugar de en disco. Spark ha sido especialmente diseñado para soportar aplicaciones iterativas y mantiene la escalabilidad y la tolerancia a fallos de MapReduce, ejecutándose mucho más rápido gracias al uso intensivo del almacenamiento en caché y de la memoria. Esta última tecnología emergente aspira a ser la nueva referencia para el tratamiento de los datos masivos.

Al ser un marco de trabajo reciente, hay pocos trabajos que abordan este tema (usando el paradigma MapReduce) desde el punto de vista de los AEs [LHH14], y menos aún desde la perspectiva de los SBRDs [LdRBH14, LdRBH15]. Esto se debe al hecho de que MapReduce tiene una penalización de rendimiento significativo cuando se ejecutan trabajos iterativos [FdRL+14], ya que vuelve a cargar los datos de entrada desde el disco en cada iteración, independientemente de cuanto haya cambiado desde las iteraciones anteriores, cosa que no ocurre con Spark, que los mantiene siempre en memoria.

De acuerdo con lo anterior, la plataforma Spark debe ser considerada como una herramienta más productiva que puede conducir a implementaciones más eficientes para el campo de los SDEs. Por ello, la investigación en el escenario de grandes conjuntos de datos para SDEs debe centrarse en la adaptación de los algoritmos del estado del arte al paradigma de programación de Spark con objeto de aprovechar la paralelización para reducir el tiempo de aprendizaje y mantener la precisión global.

1.4.4. Problemas de Clasificación Complejos

El problema de clasificación puede ser abordado bajo diferentes perspectivas dependiendo de las propiedades del problema, y el objetivo final que se persigue. Entre las diferentes posibilidades, hay en tres problemas que actualmente tienen gran interés en la comunidad investigadora y que no han sido ampliamente abordados aún con SDEs. En concreto, nos referimos a los problemas de clasificación multi-etiquetas, multi-instancia y monótonos.

1.4.4.1. Clasificación Multi-Etiqueta

En la clasificación convencional, se supone que cada instancia pertenece exactamente a una clase de entre un conjunto de tipos de candidatos. Sin embargo, hay un marco de trabajo llamado "clasificación multi-etiqueta" cuyo objetivo es obtener un modelo que clasifica un conjunto de categorías no excluyentes o, en otras palabras, asigna más de una etiqueta al mismo ejemplo.

La clasificación multi-etiqueta ha recibido una mayor atención en los últimos años debido a su importancia práctica y su interés desde un punto de vista teórico. Este tipo de tarea de clasificación se aplicó por primera vez en el campo de la categorización de documentos, pero actualmente se aplica a múltiples áreas de interés, incluyendo visión por computador, extracción de información a través de señales acústicas, y bioinformática, entre otros.

La naturaleza del problema es muy adecuada para el uso de algoritmos basados en SBRD y SDE, pero pocos esfuerzos se han aplicado a la solución de esta tarea utilizando estas herramientas. Podemos encontrar unas pocas contribuciones con respecto a este tema, ya sea utilizando una medida difusa de similitud y k-vecinos más cercanos [JTL12] o un clasificador asociativo difuso con selección de regla genética [SKK14].

Por medio de la representación de reglas difusas, podemos extraer y representar las características de este complejo problema de una manera precisa. Además y de acuerdo a los pocas trabajos que cubren actualmente este tema con SDE, debemos ser conscientes de la necesidad de llenar el vacío en este campo de investigación.

1.4.4.2. Aprendizaje Multi-Instancia

El aprendizaje multi-instancia [DLLP97] es una variación del paradigma clásico de aprendizaje supervisado en la que tenemos información incompleta sobre las clases o instancias. A diferencia del modelo clásico, donde los ejemplos se clasifican de forma individual, en aprendizaje multi-instancia "grupos" de ejemplos se clasifican como un todo. Sin embargo, se desconoce cuál de estos ejemplos es el que clasifica el "grupo" completo. Este es un campo con muchas aplicaciones prácticas en el que el número y variedad de áreas que se han beneficiado ha aumentado significativamente, destacando

entre otros la bioinformática, la recomendación Web y la detección asistida por ordenador.

Aunque en los últimos años se han desarrollado una variedad de métodos tanto para tareas de clasificación [Amo13] como de regresión, pocos utilizan SBRD [MP12] para implementarlos. En este sentido, el uso de modelos relacionados difusos, puede ayudar a considerar aproximaciones inferiores y superiores del conjunto de grupos, utilizando medidas de similitud en este nivel, por lo que el uso de SDEs puede mejorar de manera inequívoca los resultados preliminares obtenidos por estas técnicas. Por último, debemos señalar un problema abierto adicional, que consiste en la hibridación entre aprendizaje multi-etiquetas y multi-instancia [ZZHL12], y que podríamos ser abordado en un futuro próximo.

1.4.4.3. Clasificación Ordinal y Monótona

La clasificación ordinal es una forma de clasificación multiclase en el que hay un orden inherente entre las clases, pero no una diferencia numérica significativa entre ellas. En la tarea de clasificación ordinal, el atributo de salida toma valores ordinales dentro de ciertos dominios [CS11], por lo que la precisión se calcula midiendo lo "lejos" que la clase está de la real. Además de esto, existen algunos problemas en el que tanto los atributos de entrada como la clase tienen una restricción de monotonicidad, en el sentido de que la etiqueta de clase (variable de salida) no debe disminuir cuando el valor del atributo (variable de entrada) aumenta. Esto se conoce como clasificación monotónica [KS13], donde existen relaciones parciales de ordenamiento que establecen el orden entre los ejemplos, y representa un caso particular de la primera.

El problema en este escenario de clasificación es que los modelos obtenidos mediante técnicas convencionales no garantizan el cumplimiento de la restricción principal de monotonicidad anteriormente señalada. Esto ha motivado el desarrollo de algoritmos especiales que son capaces de manejar este tipo de restricciones, en particular con similitud basado en reglas difusas [BJ11]. Además, las soluciones basadas en el procesamiento de datos están comenzando a ser estudiadas, por ejemplo, los basados en medidas ordinales difusas para la selección de características [HPZ+12].

En este sentido, los SDEs puede ayudar a abordar este problema desde un punto de vista algorítmico. Esto puede llevarse a cabo por medio del desarrollo

de técnicas específicas que se aprovechan de la representación difusa para un mejor ajuste de la clasificación de salida, así como proporcionar las métricas adecuadas para guiar todo el proceso de optimización [CRHSG14].

1.4.5. Características Intrínsecas de los Datos: Superposición, Ruido y Cambio de Datos

Las características intrínsecas de los datos de un problema pueden influir en la dificultad para resolverlo con precisión. Lo ideal es ser capaz de identificar las propiedades de estos problemas antes de la etapa de aprendizaje para poder aplicar metodologías específicas que ayuden a mejorar el comportamiento de los métodos estándar, pero precisamente el problema aquí es ser capaz de identificar dichas propiedades.

En [LFG+13] se presenta una serie de problemas significativos relacionados con las características intrínsecas de los datos que, además del conocido problema de las clases no balanceadas, contribuye a una degradación del rendimiento. Estos son el problema de la superposición entre clases [SMS07], el impacto de los datos con ruido [ZW04], y el cambio o variación de datos [MTR+12]:

- La *superposición entre clases* [SMS07] aparece cuando una región del espacio de datos contiene una cantidad similar de datos de entrenamiento de cada clase. Esta situación lleva a desarrollar una inferencia con casi las mismas probabilidades a priori en esta área de superposición, lo que hace la distinción entre las clases muy difícil o incluso imposible. De hecho, algunos investigadores han demostrado que el comportamiento de un clasificador dado en un conjunto de datos se puede caracterizar por medio de una cierta métrica de complejidad de datos que mida la superposición entre las clases [LFGH11]. Teniendo esto en cuenta, podemos encontrar un trabajo reciente con SDEs que resuelve este problema, en el que los autores proponen un simple y eficaz enfoque de función de ponderación en sinergia con un sistema de clasificación difuso para clasificación no balanceada [ABS+15].

- El *ruido en los datos* afecta a la forma en que cualquier sistema de minería de datos se comporta [FV14]. El enfoque más común es aplicar una técnica de filtrado con el fin de evitar la influencia de ejemplos con ruido en la etapa de aprendizaje [SLSH15]. Aunque la representación

de conjuntos difusos maneja la incertidumbre de una manera adecuada, y tiene una buena tolerancia frente al ruido [SLH11], que nosotros sepamos no existen importantes contribuciones que hayan hecho uso de este tipo de modelos para la gestión de datos con ruidos en problemas de clasificación. De acuerdo con lo anterior, hay que destacar este marco para llevar a cabo nuevas investigaciones sobre el tema.

- El *problema de la variación de datos* [CSSL09, MTR+12] es un problema común en los modelos de predicción que se produce cuando los datos de entrenamiento y de prueba siguen diferentes distribuciones, debido por ejemplo a la irreproducibilidad de las condiciones en tiempo de entrenamiento y tiempo de test. A menudo aparece debido a problemas de sesgo o tendencia en la selección de la muestra, siendo el desplazamiento lateral de la covariable el caso más común, es decir, la entrada de valores de atributos que tienen diferentes distribuciones entre el conjunto de entrenamiento y de prueba [MTR+12]. Para solucionar esto, recientemente se ha diseñado una técnica de validación específica más conveniente con el fin de evitar los problemas anteriores [LFH14, MTSH12], que muestra un comportamiento robusto en el caso de los SDEs [LFH13].

1.4.6. Mejoras en la Evaluación de la Calidad de los SDEMOs

La comparación de diferentes técnicas de optimización multiobjetivo es siempre una tarea difícil, debido a que el resultado es un conjunto de soluciones no dominadas en vez de una única solución. En general, los investigadores están de acuerdo en considerar dos criterios informales para asegurar la calidad de un conjunto de soluciones: La distancia de la frontera del Pareto obtenida con respecto al verdadero Pareto debe ser minimizada, y las soluciones deberían estar distribuidas uniformemente a lo largo de la frontera de Pareto. En la literatura se han propuesto diversas métricas con el fin de considerar estos criterios para evaluar la capacidad de búsqueda de estos algoritmos. Las siguientes métricas son extensamente usadas: superficie cubierta, hipervolumen, la épsilon dominancia, etc. El inconveniente de estas métricas es que la diferencia de calidad entre los SBRDs obtenidos no está clara. Además, la frontera de Pareto es generada con respecto a los datos de entrenamiento, mientras el modelo obtenido debe ser evaluado con respecto a los datos de test aplicándole un análisis estadístico. Aunque en [ADH+09, GAH10] se proponen diferentes formas de comparar las soluciones obtenidas

considerando los puntos más representativos del Pareto obtenido, las técnicas propuestas en estos trabajos presentan algunos problemas cuando la frontera de Pareto generada por diferentes algoritmos reside en distintas zonas del espacio de objetivos, por lo que no es aplicable en estos casos. Por tanto, es necesaria una representación gráfica previa de las medias de la frontera de Pareto para determinar si se puede o no aplicar estas técnicas. En este sentido sería interesante el desarrollo de nuevas propuestas y métricas que permitan comparar el conjunto de soluciones obtenidas por los SDEMOs que se pretenden comparar.

1.4.7. Medidas de Interpretabilidad Fiables

Aunque en los últimos años se han propuesto diferentes taxonomías [AMGR09, GAH11, MF08, ZG08] y métricas para medir la interpretabilidad, los investigadores destacan que no existe una única medida global para cuantificar la interpretabilidad de un modelo lingüístico.

Por todo ello, parece necesario la consideración de varias de estas medidas propuestas para cuantificar la interpretabilidad así como la manera de combinar estas métricas globalmente. Para tal fin, las diferentes medidas podrían ser optimizadas como diferentes objetivos en un entorno multiobjetivo, teniendo también en cuenta la precisión. Sin embargo, el verdadero problema reside en la elección de dichas medidas, al no haber aún hoy en día un consenso entre los investigadores.

1.4.8. Dimensionalidad de los Objetivos

Los AGMOs generalmente trabajan muy bien en problemas con dos o tres objetivos, mientras su capacidad de búsqueda empeora cuando el número de objetivos se incrementa. Esta clase de problemas se pueden manejar desde diferentes puntos de vista: 1) integrando diferentes aspectos dentro de pocos objetivos; 2) seleccionando pocos aspectos como objetivos; 3) usando todos los objetivos.

El primer enfoque consistente en combinar varios objetivos en uno solo, usando pesos u operadores de agregación apropiados representa una eficaz forma de manejar muchos objetivos cuando algunos de ellos están relacionados y se pueden combinar correctamente.

El segundo enfoque consistente en reducir la dimensionalidad de los objetivos considerando que no todos ellos pueden ser necesarios, es interesante cuando hay un cierto número de objetivos no conflictivos (que por tanto pueden ser considerados redundantes) y cuando hay algunos objetivos (conflictivos o no) que pueden ser eliminados sin perder significativamente información el problema.

El tercer enfoque es el más complejo, ya que cuando se aplica un AEMO clásico a un problema con muchos objetivos la capacidad de búsqueda del AEMO basado en el concepto dominancia de Pareto se ve muy afectado y el número de soluciones que se requieren para aproximarse a la totalidad del frente de Pareto aumenta exponencialmente con el número de objetivos. Además el proceso de toma de decisiones se hace más difícil, ya que la solución final se elige entre un número más amplio de soluciones multi-objetivo.

Una posible solución es realizar una baja presión selectiva con el fin de incluir preferencias en la búsqueda del conjunto de puntos no dominados. Las propuestas que están basadas en la preferencia en la búsqueda incluyen relajar el concepto de dominancia, controlando el área de dominancia, modificando la definición de ranking, sustituyendo la métrica de distancia, etc. Estas propuestas parecen prometedoras, pero todavía necesitan investigaciones futuras.

1.5. Nuevos Retos dentro del Aprendizaje de Operadores del Sistema de Inferencia y la Interfaz de Defuzzificación

Hoy día, el campo de los SDEs es un área madura y consolidada que necesita avanzar hacia nuevos retos y problemas, entre los cuales se encuentra, como hemos comentado antes, la mejora de la calidad de la interpretabilidad y su equilibrio con la precisión y la escalabilidad en problemas con alta dimensionalidad o con grandes conjuntos de datos, entre otros.

En el contexto del aprendizaje de los operadores del SI y la ID, que es en los que se centra nuestra tesis, estos retos representan un gran hándicap, pero no

por ello hay que dejar de intentar resolverlos y enfrentarse a ellos. A continuación vamos a finalizar este capítulo proponiendo varias líneas de trabajo que serán desarrolladas detalladamente en el resto de capítulos que componen la presente memoria:

- Hasta hace poco, los SDEs desarrollados para aprender/ajustar los SBRDs han tenido como objetivo principal la mejora de la precisión del sistema sin prestar una atención especial a la interpretabilidad, pero en los últimos años se ha producido un creciente interés por el desarrollo de SDEs que tengan en cuenta ambos aspectos, que al ser contradictorios ha derivado en tratar de encontrar un correcto equilibrio entre ambas características. Ello ha provocado que, al tener el problema planteado más de un objetivo, se haya extendido el uso de AGMOs para implementar dichos SDEs.

  Se hace por tanto necesario, en el ámbito en el que se centra nuestro trabajo de tesis, el desarrollo de SDEMOs basados en el aprendizaje de los ODAs del MI que presenten un buen equilibrio entre ambos objetivos y que mejoren los resultados obtenidos, tanto en precisión como en interpretabilidad, por los SDEs actuales. En este sentido, se puede usar como modelos de aprendizaje candidatos a ser implementados con AGMOs los propuestos por AFHMP07 y MPH07 que definen respectivamente un modelo de aprendizaje cooperativo del SI y la ID y un modelo híbrido de aprendizaje basado en la cooperación entre el MI y la BR que aprende de forma conjunta y simultánea ambos componentes: la BR y los operadores paramétricos de la inferencia y defuzzificación.

- Por otro lado el uso de parámetros en el operador de conjunción y en el operador de defuzzificación introducen una serie de valores asociados indirectamente a cada regla que producen, respectivamente, un efecto similar al producido por el uso de modificadores lingüísticos en los antecedentes de las reglas [CdJH98, LCT01], y por el uso de reglas con pesos que, aunque permiten mejorar la precisión del sistema, aumentan su complejidad y por tanto decrementan su interpretabilidad. Para solventar este problema se hace necesario desarrollar mecanismos que ayuden a mejorar la interpretabilidad (en el sentido de la complejidad) de este tipo de sistemas, que tengan en cuenta estos aspectos, para lo cual sería interesante la definición de nuevas métricas e índices de interpretabilidad que los consideren.

  Por ello una línea interesante de trabajo es la de proponer medidas de interpretabilidad relacionadas con el ámbito de la inferencia y/o defuzzificación, así como el desarrollo de modelos evolutivos multiobjetivos

que las tenga en cuenta, de forma que permitan obtener no sólo soluciones más precisas sino también más interpretables, minimizando en la medida de los posible los efectos que producen la parametrización de estos operadores.

- Otra interesante línea de investigación es la relacionada con la escalabilidad y rendimiento de los sistemas difusos con un MI adaptativo. Actualmente se ha incrementado de forma paralela tanto la cantidad de información almacenada como la necesidad de desarrollar algoritmos que permitan extraer conocimiento útil de la misma de forma automática. Cuando el problema que ha de resolverse además presenta una alta dimensionalidad, es decir, un elevado número de variables o características, el diseño de modelos difusos lingüísticos para ellos se complica bastante por el crecimiento exponencial del número de reglas que se produce con el aumento lineal del número de variables. Este aumento del número de reglas provoca un aumento en la misma proporción del número de parámetros relacionados con la inferencia y defuzzificación adaptativa, ya que éstos utilizan parámetros individuales para cada regla, por lo que la adaptación de dichos parámetros se vuelve problemática, ya que su aumento provoca un crecimiento crítico en el espacio de búsqueda que se agrava aún más si tenemos en cuenta que la mayoría de operadores adaptativos de inferencia y defuzzificación trabajan con valores continuos. Todo ello provoca que se necesiten tiempos de ejecución extremadamente grandes para la convergencia de los algoritmos de aprendizaje que lo utilizan.

  Aunque hay técnicas para abordar este tipo de problemas (alta dimensionalidad y/o gran escalabilidad) tales como la reducción de datos [ADM12, AGH11], reducción de reglas, o metodologías paralelas o distribuidas [BdVMP09, dVBMP09, RABH09], ninguna de ellas considera la posibilidad de abordar el problema a través del MI por lo que, en este sentido, sería muy interesante el desarrollo de nuevos operadores adaptativos que sean más rápidos, eficientes y con menor espacio de búsqueda que los que hay actualmente, a fin de que puedan ser usados en modelos evolutivos multiobjetivos que aborden este tipo de problemas.

  La ventaja de esta propuesta es que es complementaria a cualquiera de las indicadas anteriormente. De esta forma, estos nuevos operadores adaptativos podrían ser usados en combinación con las metodologías paralelas o de reducción de datos y/o de reglas indicadas al principio, lo que lo hace aún más interesante.

Capítulo 2

Aprendizaje Cooperativo de los Operadores Difusos Adaptativos y de la Base de Reglas de los Sistemas Difusos Lingüísticos tipo Mamdani con Algoritmos Genéticos Multiobjetivo

En este capítulo se describe una propuesta de aprendizaje evolutivo multiobjetivo para el diseño de SBRDs lingüísticos de tipo Mamdani simples, compactos y precisos, basado en el aprendizaje conjunto de los parámetros de los operadores adaptativos del SI y de la ID junto con el aprendizaje y selección de la BR mediante la metodología ad-hoc con envoltura de ejemplos conocida como metodología de Cooperación entre Reglas (COR), que permite la generación de reglas con mejor grado de cooperación para alcanzar mayor precisión.

El modelo que se propone en este capítulo pretende conseguir una sinergia positiva a través del aprendizaje y la cooperación entre los operadores difusos y la BR para mejorar la precisión al tiempo que la BR aprendida es

simplificada, es decir, al tiempo que se mejora la interpretabilidad en el sentido de la complejidad [ZG08], manteniendo la filosofía del modelo evolutivo multiobjetivo para obtener un conjunto de SBRDs soluciones con diferente equilibrio entre interpretabilidad y precisión, a fin de que el diseñador pueda seleccionar aquella solución que mejor se adapte a sus requerimientos para una determinada aplicación.

Con este objetivo, se va a emplear dos modelos diferentes de AGMO basados en los ampliamente conocidos algoritmos SPEA2 [ZLT01] y NSGA-II [DAPM02], para realizar simultáneamente el aprendizaje de las reglas (incluyendo también un proceso de selección) y el ajuste de los ODAs (SI e ID) con objeto de optimizar al mismo tiempo los dos objetivos: minimizar el error del sistema y el número de reglas.

Estas propuestas y análisis se acompañan además de un estudio experimental para comparar los modelos propuestos frente a otros modelos multiobjetivo y modelos monoobjetivo basados en la precisión [AFHMP07, MPH07]. Finalmente, y atendiendo a las recomendaciones hechas por Demšar en [Dem06], se ha considerado el uso de test estadísticos no paramétricos para la validación de los resultados obtenidos.

Este capítulo se estructura de la siguiente forma: en la siguiente sección se introduce y justifica el problema a resolver. La Sección 2.2 describe el SI adaptativo y la ID adaptativa utilizadas en nuestra propuesta de aprendizaje. La Sección 2.3 describe el aprendizaje de la BR y la selección de reglas propuestas. En la Sección 2.4 se presenta el modelo de cooperación evolutiva de aprendizaje de reglas y de los operadores adaptativos difusos del MI propuesto, los AGMOs que serán usados, sus componentes y efectos, así como algunas modificaciones propuestas sobre estos AGMOs para mejorarlos y adaptarlos al propósito de esta aplicación. La Sección 2.5 desarrolla un estudio experimental en el que se emplea el modelo propuesto sobre varios problemas reales y muestra los resultados obtenidos. Finalmente la Sección 2.6 presenta algunas conclusiones y un breve resumen del capítulo.

2.1. Introducción

Como se ha indicado en el capítulo anterior, si bien la principal ventaja asociada al uso de modelos difusos lingüísticos es su mayor

interpretabilidad, uno de los inconvenientes es su inferior precisión respecto a otros métodos al modelar sistemas complejos. Esta desventaja ha hecho que en los últimos tiempos haya aumentado el interés por mejorar la precisión de los SDEs lingüísticos, desarrollándose diferentes formas de optimizar y mejorar el funcionamiento de los SBRDs lingüísticos, haciendo uso tanto de técnicas de aprendizaje genético de la BC, que actúan sobre la BD y/o la BR, como de técnicas de post-procesamiento o ajuste que actúan sobre el MI (SI y/o ID), o sobre ambos (MI y BC) de forma simultánea, procurando obtener SBRDs con un adecuado equilibrio entre interpretabilidad y precisión.

En este contexto y como se introdujo en el Capítulo 1, al ser el problema planteado un problema multiobjetivo, los AGMOs se han convertido en una herramienta idónea para mejorar el equilibrio entre precisión e interpretabilidad en los SDEs lingüísticos, por lo que en los últimos años se ha extendido ampliamente su uso.

Por otro lado, si bien la BR es el elemento fundamental que define el comportamiento del sistema, la parametrización del SI y de la ID han adquirido una importancia creciente para contribuir a alcanzar dicho equilibrio, ya que pueden encontrar una forma de defuzzificar la contribución de cada regla adecuadamente [CHMP04], mejorar la cooperación entre ellas [AFHMP07], y adaptar o aprender la BR y los operadores difusos conjuntamente [MPH07]. Se obtiene así una sinergia positiva entre ambos elementos que permite al sistema alcanzar mayores niveles de precisión, mientras que no sólo se mantiene sino que también se mejora la interpretabilidad.

Siguiendo estas ideas, presentamos en este capítulo una propuesta de aprendizaje evolutivo multiobjetivo, basado en el ajuste conjunto de ambos operadores adaptativos del MI junto con el aprendizaje y selección de la BR basada en la metodología COR, que permite la generación de reglas con mejor cooperación y mayor precisión. Dicha propuesta hace uso de dos de los AGMO más ampliamente conocidos SPEA2 y NSGA-II, en los que se mejora su capacidad de búsqueda mediante la incorporación de un método para guiarla hacia la zona del Pareto deseada.

2.2. Operadores Difusos Adaptativos Utilizados en esta Propuesta

En esta sección se describe el SI adaptativo así como la ID adaptativa utilizadas en nuestra propuesta de aprendizaje.

2.2.1. Sistema de Inferencia Adaptativo

Como se describe en el Apéndice D, el SI lleva a cabo el proceso de inferencia difuso a nivel de reglas individuales, de forma que, tras aplicar la inferencia sobre las N reglas que componen la BR, obtiene N conjuntos difusos B'_i que representan las acciones difusas inferidas por él.

El SI emplea dos operadores: el de conjunción, $C(\cdot)$ para calcular el grado de emparejamiento, y el de implicación, $I(\cdot)$ para calcular el consecuente inferido.

Con objeto de permitir que el SI pueda adaptarse, estos dos operadores son susceptibles de utilizarse de forma parametrizada introduciendo un único parámetro α y ω, para la conjunción o la implicación respectivamente, o introduciendo parámetros individuales para cada regla de la BR, α_i o ω_i. Como cabe esperar (véase Sección D.1.4 del Apéndice D), el modelo con un sólo parámetro ofrece peores resultados. Por otro lado el operador de implicación parametrizado tiene escasa influencia frente al operador de conjunción adaptativo. Por esta razón y para dotar de mayor flexibilidad al sistema, aquí se utiliza el modelo que opta por el basado en la parametrización del operador de conjunción con parámetros individuales para cada regla α_i.

Tanto para el operador de conjunción como para el de implicación utilizamos t-normas, por las razones explicadas en el Apéndice D (véase Sección D.1.2). En concreto, como operador de implicación utilizamos la t-norma del mínimo, y como operador de conjunción adaptativo, teniendo en cuenta las diferentes t-normas parametrizadas [Miz89] (véase Tabla D.2 del Apéndice D) y los estudios realizados en [AFHMP07], la t-norma adaptativa de Dubois, por ser la más eficiente computacionalmente, la más precisa, y ofrecer en función del valor de su parámetro α, un comportamiento comprendido entre la t-norma del mínimo (cuando $\alpha = 0$) y la del producto algebraico (cuando $\alpha = 1$) (véase Tabla D.3 del Apéndice D), que son las que mejores resultados ofrecen [CHP97].

2.2.2. Método de Defuzzificación Adaptativo

Con respecto a la ID adaptativa, que es la encargada de la tarea de agregar la información aportada por cada uno de los conjuntos difusos individuales B'_i inferidos por el SI y transformarla en una salida precisa, y por las razones expuestas en las Secciones D.1.3 y D.1.4.2 del Apéndice D, en esta memoria trabajamos con el modo FITA *(First Infer Then Agregate:* defuzzificar primero, agregar después) [BH94], utilizando como término funcional para la defuzzificación el de tipo producto con un parámetro β_i para cada regla, al poder calcularse de forma más eficiente y obtener resultados similares al funcional potencia [CHMP04]. Como valor característico del conjunto difuso inferido por cada regla se ha utilizado el centro de gravedad (CG).

El método de defuzzificación que empleamos (véase Expresión D.14 del Apéndice D) es equivalente al conocido WCOA cuando los parámetros β_i se igualan a la unidad.

2.3. Aprendizaje de la Base de Reglas

Una vez descrito el SI adaptativo y la ID adaptativa utilizadas en nuestra propuesta de aprendizaje, y antes de exponer el nuevo modelo propuesto introduciremos en esta sección una parte esencial del mismo como es el modelo elegido para aprender la BR. Para este propósito, describimos a continuación las técnicas utilizadas para aprender y reducir el número de reglas: la selección de reglas difusas y el método de aprendizaje de la BR, que propicia la generación de reglas con mejor cooperación y por tanto mayor precisión.

2.3.1. Selección de Reglas Difusas

Las técnicas de reducción del conjunto de reglas difusas son aplicadas generalmente como una etapa de post-procesamiento una vez que un conjunto inicial de reglas difusas ha sido extraído. Estas técnicas tratan de minimizar el número de reglas de un SBRD manteniendo (e incluso mejorando) el rendimiento del sistema. Para ello, las reglas erróneas y contradictorias que

degradan el rendimiento se eliminan, obteniendo un conjunto de reglas difusas más cooperativo y, como resultado, potencialmente se mejora la precisión del sistema, a la vez que se aumenta su interpretabilidad. La reducción de la complejidad del modelo es una forma de mejorar la legibilidad del sistema, ya que un sistema compacto con pocas reglas generalmente requiere menos esfuerzo para su interpretación. De entre todas las técnicas existentes, la selección de reglas es una de las más utilizadas.

Estas técnicas de selección de reglas además pueden combinarse fácilmente con otras técnicas de post-procesamiento para obtener SBRDs más compactos y precisos. De esta manera, algunos trabajos han considerado la selección de reglas junto con el aprendizaje de los operadores parametrizados del SI mediante la codificación de todos ellos (reglas y parámetros) en el mismo cromosoma [AFHMP07, MPH07].

En este trabajo, utilizamos precisamente esta metodología cuando usamos modelos con un solo objetivo sin selección de reglas implícita. Sin embargo, cuando el modelo usa el aprendizaje de reglas junto con la selección, se utiliza otro mecanismo que se describe a continuación.

2.3.2. Algoritmo de Aprendizaje de la Base de Reglas

El aprendizaje de la BR lingüística utilizado en este trabajo se basa en la metodología ad-hoc con envoltura de ejemplos COR [CCH02], una técnica que permite la generación de reglas con mejor cooperación basada en la búsqueda de reglas a través de metaheurística evolutiva. Esta metodología maneja un conjunto de etiquetas consecuentes sobre los que realiza una búsqueda combinatoria para encontrar el mejor conjunto de reglas que cooperen entre sí.

En lugar de seleccionar las reglas buscando el consecuente con el mejor rendimiento en cada subespacio como hacen la mayoría de las técnicas de generación de BR mediante técnicas con envoltura de ejemplos, como por ejemplo Wang y Mendel (WM) [WM92], la metodología COR considera la posibilidad de usar otros consecuentes, diferentes del mejor, encontrando un conjunto de reglas cooperativas buscando entre los consecuentes con mejor ejecución global, lo que permite obtener SBRD más precisos gracias a tener una BR con mejor cooperación.

Esta técnica va a ser utilizada en el estudio experimental de este capítulo tanto para los algoritmos de un solo objetivo como para los algoritmos multiobjetivos cuando empleamos juntos selección y aprendizaje de regla.

Una descripción más detallada de la metodología COR puede consultarse el Apéndice B, donde además de dicha metodología se describe también el método WM [WM92], uno de los métodos de aprendizaje de BR basado en ejemplos más populares y rápidos. El método WM también va a ser utilizado en el estudio experimental de este capítulo.

2.4. Aprendizaje Cooperativo de la Base de Reglas y el Mecanismo de Inferencia con AGMOs

Como se ha indicado el objetivo principal es obtener modelos con distintos balances entre la precisión y la interpretabilidad. Proponemos un modelo difuso que presente cooperación entre el SI, la ID y la BR construido en base a dos pilares: el SI adaptativo y la ID adaptativa y la metodología COR.

El modelo propuesto para ello integra por tanto los parámetros de los operadores adaptativos como los consecuentes de las reglas, que serán aprendidos en un único proceso evolutivo para así obtener un buen nivel de cooperación entre ellos.

Antes de presentar en detalle el modelo evolutivo multiobjetivo propuesto plantearemos algunos aspectos relacionados con los AGMOs utilizados en el presente trabajo. De esta forma, en primer lugar, se describirá los AGMOs que serán usados en nuestro trabajo, así como las mejoras que proponemos realizar sobre ellos, para hacer que el proceso de búsqueda funcione mejor, de forma que se centre en la zona del frente de Pareto deseada, aquella con una alta precisión y con el menor número de reglas posibles. Finalmente se presentará el Modelo de Cooperación Evolutiva con los AGMOs propuestos.

2.4.1. AGMOs Utilizados

Utilizamos dos AGMOs específicos considerando un esquema de codificación triple: [codificación de los parámetros de los operadores adaptativos difusos (SI e ID) y BR]. Estos AGMOs que utilizamos están basados

en SPEA2 [ZLT01] y NSGA-II [DAPM02] (véase Apéndice A), que son dos de los AGMOs más conocidos y representativos de los pertenecientes a la segunda generación, caracterizados por utilizar el elitismo como mecanismo estándar.

Se han realizado algunos cambios en estos algoritmos pero antes de explicar las citadas modificaciones vamos a estudiar qué tipo de soluciones se podrían encontrar en la frontera de Pareto óptima para el problema planteado. De esta manera, una vez obtenida una aproximación del Pareto óptimo y determinar hacia qué zona de dicho Pareto queremos ir, podremos describir y justificar mejor qué modificaciones proponemos efectuar a los AGMOs SPEA2 y NSGA-II utilizados en nuestro modelo.

2.4.1.1. Frontera del Pareto en el Problema de la Interpretabilidad-Precisión

El ajuste de los parámetros de los operadores adaptativos normalmente necesita que el modelo inicial tenga un número de reglas elevado para llegar a un nivel de precisión adecuado. Para conseguir que este conjunto de reglas inicial sea suficiente se suelen utilizar métodos de aprendizaje de reglas que aseguren un nivel de cubrimiento generalmente más alto de lo que sería necesario. Todo ello da lugar a que se obtengan reglas que, aunque inicialmente son necesarias, una vez aplicado el ajuste ya no lo son e incluso dificultan un buen ajuste de las demás.

De esta forma, podemos encontrarnos con los siguientes tipos de reglas: *Reglas malas* (erróneas o conflictivas), que dificultan el buen funcionamiento del sistema; *Reglas redundantes o irrelevantes*, que no mejoran significativamente el rendimiento del sistema; *Reglas complementarias* que complementan a otras mejorando levemente el rendimiento del sistema; y *Reglas importantes* que son necesarias para que el sistema presente un rendimiento razonable.

Por otro lado, los AGMOs estándar pueden presentar algunos problemas, como por ejemplo que no todas las soluciones no dominadas obtenidas sean interesantes. Por ejemplo, no son interesantes soluciones no dominadas con un error muy alto y un número de reglas muy bajo, ya que no presentan un buen equilibrio entre interpretabilidad y precisión. Además, al existir soluciones de este tipo en el frente de Pareto, se favorece la selección de padres con un número de reglas muy distinto para el cruce, generando hijos con baja precisión pues los parámetros de ajuste son muy distintos y no tiene sentido el cruce, salvo para obtener nuevas configuraciones de las reglas seleccionadas.

Teniendo en cuenta la posible existencia de estos tipos de reglas y la distinta calidad de las soluciones no dominadas existente en el frente de Pareto, podemos distinguir las siguientes zonas en el espacio de los objetivos:

- *Zona de Reglas Malas*: En esta zona no existiría frente de Pareto ya que al quitar este tipo de reglas mejoraría la precisión y dichas soluciones serían dominadas por otras.
- *Zona de Reglas Redundantes o Irrelevantes*, formada por soluciones que ya no contienen reglas malas pero que aun mantienen reglas redundantes e irrelevantes. Quitando este tipo de reglas la precisión apenas se vería afectada.
- *Zona de Reglas Complementarias*, formada por soluciones sin reglas malas ni redundantes. A medida que se eliminen este tipo de reglas, la precisión bajaría suavemente.
- *Zona de Reglas Importantes*, formada por soluciones únicamente con las reglas imprescindibles. A medida que se supriman este tipo de reglas la precisión se verá fuertemente afectada.

En la Figura 2.1, podemos ver una aproximación del Pareto óptimo para el problema de ajuste y selección de reglas difusas con el doble objetivo de simplicidad y precisión. En dicha gráfica se muestran las distintas zonas del espacio de los objetivos junto con la zona del Pareto deseada para encontrar soluciones con un buen equilibrio ambos objetivos.

Figura 2.1: Frontera del Pareto

Esta zona deseada se corresponde con la zona de reglas complementarias, es decir, deseamos quitar todas las reglas posibles sin afectar seriamente la precisión del modelo finalmente obtenido.

Teniendo en cuenta todo lo expuesto, el AGMO considerado no debe obtener todo el frente de Pareto, puesto que esto dificultaría la obtención de soluciones precisas fomentando el cruce de soluciones con distintas configuraciones de reglas, que consiguirían el óptimo con parámetros muy distintos entre sí.

2.4.1.2. Mejoras Aplicadas a los Algoritmos NSGA-II y SPEA2

En nuestra propuesta centraremos la búsqueda en la zona del Pareto con soluciones que, dentro de un cierto rango de precisión, tengan la complejidad (número de reglas) más bajo posible. Para ello se propone guiar el proceso de búsqueda de SPEA2 y NSGA-II hacia la zona del frente del Pareto que nos interesa empleando un método denominado Método de Dominación Guiada [BKS00] (*Guided Domination Approach*), que permite dar más prioridad a un objetivo que al otro mediante la ponderación de las funciones objetivos. De esta forma, dando más peso al objetivo de precisión y menos al objetivo del número de reglas, conseguimos priorizar más la precisión que la interpretabilidad, reduciendo además el esfuerzo en la búsqueda.

La idea que se persigue básicamente es reducir el espacio de búsqueda, al concentrar ésta en una zona del frente del Pareto más reducida e interesante: aquella en la que las soluciones son más precisas, siendo la densidad de las soluciones obtenidas mayor.

La función que pondera los objetivos se define de la siguiente manera:

$$\Omega_i\left(f(x)\right) = f_i(x) + \sum_{j=1,\, j\neq i}^{M} a_{ij} f_j(x), \qquad i=1,2,\ldots M \qquad (2.1)$$

donde M es el número de objetivos y a_{ij} es la cantidad de ganancia en la función objetivo j-ésima $f_j(x)$, por cada pérdida de una unidad en la función objetivo i-ésima $f_i(x)$. Para definir el conjunto de ecuaciones generadas por la expresión anterior hay que fijar todos los valores de la matriz a, excepto los elementos de la diagonal principal, que son todos 1.

Para aplicar este método hay que redefinir el concepto de dominancia. De esta forma, si antes el concepto de dominancia, para el caso concreto de problemas de minimización de objetivos se definía, como:

$$a \succ b \leftrightarrow \forall i = 1,2,...,M \;\; f_i(a) \leq f_i(b) \wedge \exists\, i \mid f_i(a) < f_i(b) \qquad (2.2)$$

donde *a, b* representan dos soluciones y *M* es el número de objetivos.

Ahora este método redefine el concepto de dominancia de la siguiente manera:

$$a \succ b \leftrightarrow \forall i = 1,2,...,M \;\; \Omega_i\left(f(a)\right) \leq \Omega_i\left(f(b)\right) \wedge \exists\, i \mid \Omega_i\left(f(a)\right) < \Omega_i\left(f(b)\right) \qquad (2.3)$$

Es decir, decimos que una solución *a* domina a otro solución *b* si es mejor o igual en todos los objetivos ponderados y al menos estrictamente mejor en uno de ellos.

Nos gustaría hacer hincapié en que este enfoque puede ser visto como un principio de dominancia modificado, es decir, como una propuesta de optimización multiobjetivo con el principio de dominancia original actuando sobre una transformación lineal de un conjunto de funciones objetivo. Puede observarse que la definición anterior de dominancia en el vector objetivo *f* es la misma que la definición original de dominancia en el vector transformado Ω. Este enfoque es diferente a uno que hubiera tenido en cuenta sólo los algoritmos monoobjetivo utilizando estas combinaciones lineales [BKS00], porque en este caso obtenemos un conjunto de SBRDs en una sola ejecución.

Para el caso particular de nuestro trabajo, en el que hay dos objetivos (*M=2*), las dos funciones objetivos ponderadas son mostradas en la siguiente expresión:

$$\begin{aligned} \Omega_1(f_1, f_2) &= f_1 + a_{12} f_2 \\ \Omega_2(f_1, f_2) &= a_{21} f_1 + f_2 \end{aligned} \qquad (2.4)$$

donde f_1, f_2 representan el objetivo precisión y número de reglas respectivamente, a_{12} representa la ganancia en la función objetivo *número de reglas* por la pérdida de una unidad en la función objetivo *precisión*, y a_{21} la ganancia en el objetivo *precisión* por la pérdida de una unidad en el objetivo *número de reglas*.

Es importante destacar que, para poder aplicar este método, las funciones objetivos deben estar normalizadas. Por ello cada objetivo previamente debe ser dividido entre la diferencia de los valores máximo y mínimo que cada uno de ellos puede alcanzar. De esta forma, la expresión de las dos funciones objetivos ponderadas y normalizadas queda de la siguiente manera:

$$\Omega_1 (f_1, f_2) = f_1 / (f_1^{max} - f_1^{min}) + a_{12} f_2 / (f_2^{max} - f_2^{min})$$
$$\Omega_2 (f_1, f_2) = a_{21} f_1 / (f_1^{max} - f_1^{min}) + f_2 / (f_2^{max} - f_2^{min}) \quad (2.5)$$

donde f_1^{max}, f_1^{min} representan respectivamente los valores máximos y mínimos que puede alcanzar la función objetivo *precisión* y f_2^{max}, f_2^{min} indica los valores respectivos, mas grande y más pequeño, que puede alcanzar el objetivo *número de reglas*. En nuestro caso concreto, los valores máximos y mínimos que hemos tomado corresponden a los valores máximos y mínimos alcanzados en cada uno de los dos objetivos en la frontera de Pareto que hay en cada momento.

2.4.2. Propuesta Evolutiva Multiobjetivo

Una vez descritas las mejoras aplicadas a los AGMOs para hacer que el proceso de búsqueda funcione mejor y se centre en la zona del frente de Pareto con una precisión más alta y con el menor número de reglas posible, pasamos a continuación a describir en detalle los componentes del modelo evolutivo multiobjetivo propuesto.

Para ello, en primer lugar se describirá la codificación de los cromosomas de la población, a continuación se indicará cómo se genera la población inicial y por último se indicarán los operadores genéticos de cruce y mutación utilizados.

2.4.2.1. Modelo y Esquema de Codificación

Para representar los parámetros del modelo evolutivo propuesto se ha considerado un esquema de codificación en el que el cromosoma, C, consiste en un vector de genes que contiene los parámetros del MI y del aprendizaje de la BR (incluida la selección).

Se trata por tanto de un vector con un esquema de codificación triple, cada uno de ellos con un tamaño igual al número de reglas que componen la BR inicial:

$$C_C + C_D + C_R$$

donde C_C codifica los parámetros del operador de conjunción, C_D los parámetros del operador de defuzzificación y C_R el aprendizaje de la BR y la selección de las reglas.

Consideraremos dos segmentos diferenciados en el cromosoma, el destinado al MI ($C_c + C_D$) y el reservado al Aprendizaje y Selección de Reglas (C_R). A continuación se describirá detalladamente cada uno de estos elementos:

a) *Mecanismo de Inferencia ($C_C + C_D$).* La zona del cromosoma dedicada a los parámetros de los operadores del MI, tiene dos partes:

- La del operador de conjunción, C_C, que utiliza un esquema de codificación real para codificar los N parámetros reales α_i (genes) de dicho operador, uno para cada regla R_i de la BR lingüística. El universo de discurso de estos parámetros depende del operador de conjunción adaptativo que estemos evaluando. En este caso, al usar la t-norma de Dubois (véase Expresión D.9), cada gen puede tomar valores en el intervalo [0, 1], es decir, entre la t-norma del mínimo y la del producto algebraico.

- La del método de defuzzificación, C_D, que también utiliza un esquema de codificación real, formada por N parámetros β_i, uno para cada regla R_i de la BR lingüística, que codifica los parámetros de dicho operador (véase Expresión D.14). Cada gen en este caso puede tomar valores en el intervalo [0, 10], suficiente para actuar como peso de esa regla. Este intervalo ha sido seleccionado de acuerdo con el estudio desarrollado en [CHMP04], y permite la atenuación así como la mejora del grado de emparejamiento.

b) *Aprendizaje (y Selección) de Reglas (C_R).* La parte del cromosoma destinada a codificar la BR aprendida (incluida la selección de reglas) utiliza un esquema de codificación entero, formado por una cadena de N genes de tipo entero, uno para cada regla R_i de la BR lingüística inicial, cada uno representando un consecuente candidato de cada una de dichas reglas. Cada gen podrá tomar los valores en el intervalo [0, nC_{Ri}] siendo nC_{Ri} el número de

consecuentes candidatos posibles de la reglas R_i calculado por COR. En función de si una regla se selecciona o no, el valor '0' es asignado al gen correspondiente cuando la regla no es seleccionada:

$$C_R = (C_{R1}, \ldots, C_{RN}) \mid C_{Ri} \in \{ 0, \ldots, nC_{Ri} \} \qquad (2.6)$$

La Figura 2.2 ilustra el esquema de codificación propuesto en este trabajo, siendo N el número total de reglas de la BC. En la zona del MI, α_i y β_i representan la parametrización del operador de conjunción y de la ID respectivamente. En el segmento del cromosoma dedicado a la BR, los valores [0, …, nC_{Ri}] indican el consecuente de cada regla R_i indicando el 0 que dicha regla no debe tenerse en cuenta. Este esquema de cromosoma permite la búsqueda conjunta de una buena solución para elementos del MI y la BR.

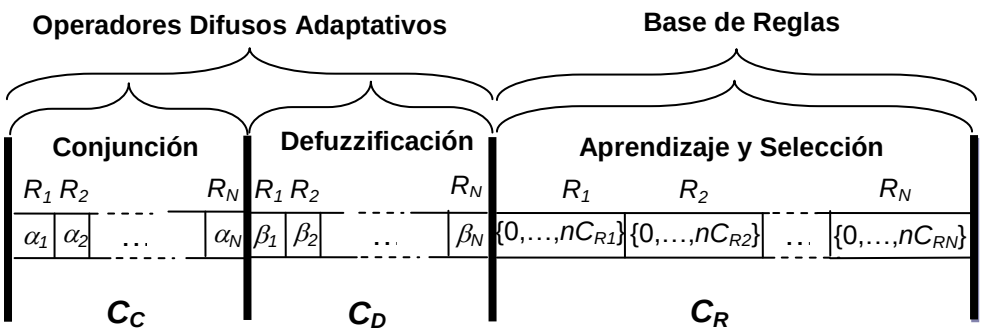


Figura 2.2: Esquema de Codificación propuesto para codificar el MI y el aprendizaje y selección de la BR

2.4.2.2. Generación de la Población Inicial

La población inicial de cromosomas en los AGMOs con los que trabajamos estará compuesta por individuos con un número de genes triple igual al número de reglas para cada problema.

La población inicial se inicializa aleatoriamente tanto en la parte correspondiente a los operadores difusos como en la parte correspondiente al aprendizaje de la BR con la excepción de un único cromosoma, de acuerdo a la siguiente configuración:

Inicialización del primer individuo de la población:

- Los N genes del operador de conjunción C_C son inicializados a 0 con el fin de hacer que la t-norma adaptativa de Dubois equivalga inicialmente a la t-norma del mínimo.

- Los N genes del método de defuzzificación C_D son inicializados a 1 para que el operador de defuzzificación comience como el WCOA estándar.

- Los N genes que codifican el aprendizaje y selección de reglas C_R son inicializados al consecuente obtenido por el método WM [WM92], seleccionando todos en su totalidad, sin descartar a priori ninguna de ellos.

Inicialización del resto de individuos de la población:

- Los N genes del operador de conjunción C_c y del método de defuzzificación C_D son inicializados de forma aleatoria, de forma que cada uno de ellos toma valores en el intervalo [0, 1] y [0, 10] respectivamente. Este último intervalo ha sido seleccionado de acuerdo con el estudio desarrollado en [CHMP04].

- Los N genes que codifican el aprendizaje y la selección de reglas C_R son inicializados aleatoriamente a uno de sus posibles consecuentes candidatos con todas las reglas activas, es decir, sin ningún 0, a fin de no descartar a priori ninguna de las reglas, al igual que el primer individuo de la población.

2.4.2.3. Importancia de la Población Inicial

Llegados a este punto, tenemos que destacar que la forma de crear las soluciones de la población inicial en la parte del aprendizaje y selección de reglas es un factor de relevancia.

Generalmente, en un AG genera la población inicial de forma totalmente aleatoria (selección aleatoria de las reglas iniciales). Sin embargo, en este caso, a fin de lograr soluciones con una alta precisión no deberíamos perder reglas que podrían presentar una positiva cooperación una vez que sus parámetros han evolucionado. La mejor forma de hacer esto es comenzar con soluciones con todas las reglas posibles activas favoreciendo una eliminación progresiva

de reglas malas (aquellas que no mejoran con el ajuste de sus parámetros) exclusivamente mediante mutación al principio y por cruce posteriormente.

En cualquier caso, se realizaron diferentes pruebas considerando la generación completamente aleatoria de la población inicial, obteniéndose de esta forma soluciones más simples, pero con peores errores en entrenamiento y en test, por lo que, como comentamos anteriormente, en la población inicial comenzamos utilizando todas las reglas posibles, sin descartar ninguna de ellas.

2.4.2.4. Operadores de Cruce y Mutación

Los operadores genéticos empleados para obtener los descendientes a partir de los padres son el cruce y la mutación. El operador de cruce es aplicado a todos los individuos que forman la población de padres, mientras que el operador de mutación es aplicado con una determinada probabilidad P_m asignada como entrada del algoritmo.

El operador de cruce utilizado depende de la parte del cromosoma donde sea aplicado:

- En la parte real, que codifica los parámetros de los ODAs del MI, se utiliza el operador BLX-0.5 [Esh91, HLS03].

- En la parte entera, que codifica la selección de reglas, se utiliza el cruce HUX [Esh91].

Todos los individuos que forman la población de padres son cruzados de dos en dos, generándose por cada cruce cuatro descendientes como resultado de combinar la parte de aprendizaje y selección de reglas C_R con la parte del MI $C_c + C_D$ de los dos hijos generados. Estos cuatro descendientes son evaluados y los dos mejores reemplazan a los padres.

Una vez obtenida la nueva población, se aplica el operador de mutación para cambiar aleatoriamente el valor de un gen en la parte C_R y el valor de otro gen (uno sólo) en la parte $C_c + C_D$.

El operador de mutación no se aplica a todos los individuos sino a unos pocos, siendo $P_m = 0.2$ la probabilidad con la que se aplica en la nueva población.

2.5. Estudio Experimental

Con el fin de analizar el modelo propuesto, se ha llevado a cabo un estudio experimental en el que dicho modelo se ha comparado con varios métodos guiados por un único objetivo y frente a varios métodos multiobjetivo. Para ello se han utilizado nueve problemas reales de regresión de complejidades diferentes (en número de variables y datos), los cuales han sido descargados de la página web del proyecto KEEL (http://sci2s.ugr.es/keel/datasets.php) [AFSG+09].

La Tabla 2.1 resume las principales características de los nueve conjuntos de datos. Para cada problema se muestra el número de variables y el número de ejemplos. Los problemas están ordenados de menor a mayor dimensionalidad, empezando por el problema PLA que es el de más baja dimensionalidad y terminando por los problemas MOR y TRE, que son los que tienen la dimensionalidad mas alta.

Tabla 2.1: Conjunto de datos considerados para el estudio experimental

Conjunto de Datos	**Nombre**	**Variables**	**Ejemplos**
Plastic Strength	PLA	3	1650
Quake	QUA	4	2178
Electrical Maintenance	ELE	5	1056
Abalone	ABA	9	4177
Stock Prices	STP	10	950
Ankara Weather	WAN	10	1609
Izmir Weather	WIZ	10	1461
Mortgage	MOR	16	1409
Treasure	TRE	16	1409

Antes de mostrar y analizar los resultados obtenidos en el estudio experimental vamos a describir como han sido configurados los conjuntos de datos y la metodología de comparación utilizada en dicho estudio.

2.5.1. Configuración del Estudio Experimental

Como comentamos en la Sección 2.3.2 del presente capítulo, el conjunto inicial de reglas difusas lingüísticos candidatas (BR inicial) de nuestro modelo se obtiene a partir de COR [CCH02]. No obstante, con el fin de poder

compararlo con otros métodos de generación de reglas, los modelos con los que nos comparamos no sólo van a obtener el conjunto inicial de reglas lingüísticos candidatas mediante COR, sino que también van a emplear otro conocido algoritmos ad-hoc de aprendizaje basado en ejemplos: el algoritmo de WM [WM92]. Ambos métodos de generación de reglas los denotaremos como BR_{COR} y BR_{WM}, respectivamente (véase el Apéndice B para los detalles de estos métodos).

La granularidad de las etiquetas ha sido elegida de forma que hubiera un razonable equilibrio entre el número de variables y la cantidad de ejemplos de datos numéricos dentro del conjunto de datos de las aplicaciones. De esta forma, las particiones lingüísticas consideradas están compuestas por cinco términos lingüísticos con forma triangular en el caso de los conjuntos de datos con menos de nueve variables y tres términos lingüísticos en las restantes (lo que ayuda a obtener un número más razonable y compacto de reglas). Los modelos considerados en el estudio experimental se describen brevemente en la Tabla 2.2.

De forma general, en la tabla se utiliza el símbolo "+" para indicar "secuencia" y el símbolo "-" para indicar "simultaneidad". De esta forma, si la selección de las reglas se realiza después del aprendizaje de la BR, lo denotamos como BR_{WM}+S cuando usamos como BR inicial la obtenida por WM y BR_{COR}+ S cuando la que usamos es COR. De la misma forma, si la selección de regla se realiza conjuntamente con el aprendizaje de la BR utilizando la metodología COR, lo denotamos como $BR_{COR\text{-}S}$ y si los ODAs se aprenden a la vez que se realiza selección de reglas lo denotamos como ODA – S. En consecuencia, BR_{WM} + ODA - S indica que a partir de la BR inicial obtenida con WM, se realiza posteriormente el aprendizaje evolutivo (usando en este caso un modelo monoobjetivo basado en AG de tipo CHC [Esh91]) de los ODAs de forma conjunta a la selección de reglas y ODA–$BR_{COR\text{-}S}$ indica que el aprendizaje evolutivo (con CHC) de los ODAs se realiza junto con el aprendizaje de la BR (y selección de reglas) utilizando la metodología COR.

Con respecto a los modelos multiobjetivo utilizados, BR_{WM} + (ODA - S)$_{SP2}$ y BR_{WM} + (ODA - S)$_{NS\text{-}II}$ son los AGMOs que, a partir de la BR inicial generada por WM, aprenden los ODAs a la vez que realizan la selección de reglas (con los modelos basados en SPEA2 y NSGAII, respectivamente). Los nuevos modelos propuestos en este trabajo son (ODA - $BR_{COR\text{-}S}$)$_{SP2}$ y (ODA - $BR_{COR\text{-}S}$)$_{NS\text{-}II}$, es decir, el uso de los AGMOs basados en SPEA2 y NSGA-II respectivamente, para aprender conjuntamente en el mismo proceso evolutivo multiobjetivo los ODAs y la BR utilizando la metodología COR a la vez que se realiza la selección de reglas.

Tabla 2.2: Modelos considerados para la comparación

Ref.	Modelo	Descripción
		SDEs
[WM92]	BR_{WM}	BR obtenida con WM
[CCdJH05]	BR_{WM}+S	BR obtenida con WM y Selección de Reglas posterior
[CCH02]	BR_{COR}	BR obtenida con COR
[CCH02]	$BR_{COR\text{-}S}$	BR obtenida con COR usando Selección de Reglas
-	BR_{COR}+S	BR obtenida con COR y Selección de Reglas posterior
[MPH07]	BR_{WM}+ODA–S	BR obtenida con WM y aprendizaje posterior de los ODAs con Selección de Reglas usando un modelo monoobjetivo basado en CHC
[MPH07]	ODA–$BR_{COR\text{-}S}$	Aprendizaje simultáneo de los ODAs y de la BR mediante COR con selección de reglas usando un modelo monoobjetivo basado en CHC
		SDEMOs
-	BR_{WM}+$(ODA–S)_{SP2}$	BR obtenida con WM y aprendizaje posterior de los ODAs con Selección de Reglas usando un modelo multiobjetivo basado en SPEA2
-	BR_{WM}+$(ODA–S)_{NS\text{-}II}$	BR obtenida con WM y aprendizaje posterior de los ODAs con Selección de Reglas usando un modelo multiobjetivo basado en NSGA-II
-	$(ODA–BR_{COR\text{-}S})_{SP2}$	Aprendizaje simultáneo de los ODAs y de la BR, basado en COR-S, usando un modelo multiobjetivo basado en SPEA2
-	$(ODA–BR_{COR\text{-}S})_{NS\text{-}II}$	Aprendizaje simultáneo de los ODAs y de la BR, basado en COR-S, usando un modelo multiobjetivo basado en NSGA-II

Para comprobar la fiabilidad de los resultados obtenidos en todos los experimentos se ha utilizado un método de validación cruzada. Dicho método consiste en dividir el conjunto de datos en dos partes, realizando el análisis del sistema sobre una de ellas (conjunto de entrenamiento) y dejando la otra para la posterior validación del sistema (conjunto de test).

El tipo de validación cruzada usado es el *k-fold*, que consiste en dividir el conjunto de datos en *k* subconjuntos, dejando uno de ellos como conjunto de test y el resto como conjunto de entrenamiento. Este proceso se repite para los *k* subconjuntos.

Concretamente, para llevar a cabo el estudio experimental hemos adoptado un *modelo de validación cruzada de 5 particiones* (*5-fold cross-validation*), es decir, se han generado 5 particiones aleatorias de los datos, cada una con el 20% del conjunto total de datos, utilizando para entrenamiento la combinación de cuatro de ellas (80%) y para test la partición restante (20%), obteniéndose así 5 particiones distintas al 80% y 20% en entrenamiento y test respectivamente. Para cada una de estas 5 particiones de datos (formadas por las combinaciones de 4 de las 5 particiones aleatorias generadas inicialmente), los métodos de aprendizaje se han ejecutado 6 veces, utilizando 6 semillas distintas para el generador de números aleatorios, lo que supone 30 ejecuciones para cada uno de los modelos. Por tanto, para cada conjunto de datos, se ha considerado los resultados medios de las 30 ejecuciones de cada modelo.

En el caso de los modelos con enfoque multiobjetivo, para cada conjunto de datos y por cada prueba, se ha generado el frente de Pareto aproximado y, de todos los puntos que componen el Pareto, se han considerado sólo los tres puntos más representativos: el punto de máxima interpretabilidad (MAX_{INT}), el mediano ($MEDIA_{(INT\ /\ PRE)}$) y el más preciso (MAX_{PRE}). Para cada uno de dichos puntos y para conjunto de datos, se han calculado sobre las 30 pruebas realizadas, los valores medios y las desviaciones estándar del ECM obtenido en los SBRDs en los conjuntos de entrenamiento y test así como los valores medios y las desviaciones estándar del número de reglas (#*R*) en test. Para los enfoques basados en un solo objetivo, calculamos los mismos valores medios y desviaciones estándar correspondientes a las 30 soluciones obtenidas para cada conjunto de datos (en este caso en el único punto solución que generan).

Esta forma de trabajar se empleó en problemas de dos objetivos en [ADH+09], a fin de comparar los modelos con un solo objetivo con los multiobjetivo considerando solamente el objetivo de precisión, y posteriormente, en [GAH10], se extendió a problemas con más de dos objetivos mediante la proyección de los frentes de Pareto obtenidos en los planos generados al considerar pares de objetivos, permitiendo de esta forma que las soluciones no dominadas pudieran ser analizadas considerando los 3 puntos más representativos del frente del Pareto para cada par de objetivos (cada plano proyectado).

Para evaluar un cromosoma se utiliza el ampliamente conocido ECM:

$$ECM(S) = \frac{\frac{1}{2}\sum_{k=1}^{N}(y_k - S(x_k))^2}{N} \qquad (2.7)$$

donde S denota el modelo difuso cuyo SI utiliza como operador de conjunción la t-norma adaptativa de Dubois mostrada en la Expresión D.9, la t-norma del mínimo como operador de inferencia, y el método de defuzzificación adaptativo mostrado en la Expresión D.14 como operador de defuzzificación. Para calcular dicho ECM se usan un conjunto de evaluaciones sobre los datos formados por N pares de datos numéricos $Z_k = (x_k, y_k)$, $k = 1,...,N$, siendo x_k los valores de las variables de entrada, y siendo y_k los valores correspondientes a las variables de salida asociadas. Este ECM se utilizará por parte de los AEs para evaluar cada uno de los cromosomas que tenga la población, siendo el objetivo de los algoritmos monoobjetivos minimizar dicho error, igual que lo es para los AGMOs, junto con la minimización del número de reglas $\#R$.

Con el fin de evaluar y determinar si existen diferencias significativas entre los resultados obtenidos, se ha realizado un análisis estadístico [Dem06, GFLH09, GFLH10, GH08] utilizando tests no paramétricos, de acuerdo con las recomendaciones formuladas en [Dem06, GFLH09], donde presentan un conjunto de sencillos, seguros y sólidos tests no paramétricos para la comparación estadística de clasificadores. En concreto, en este estudio se ha utilizado el test de ranking de signos de Wilcoxon [She03, Wil45] para realizar comparación por pares. Una descripción detallada de este test se presenta en el Apéndice C.

Para realizar los tests, se utilizó un nivel de confianza $\alpha = 0.10$. En el caso concreto del test de Wilcoxon, como los resultados que obtiene se basa en el cálculo de las diferencias entre dos medias muestrales (normalmente, la media de los errores de las pruebas obtenidas por un par de algoritmos diferentes en distintos conjuntos de datos), ha sido necesario normalizar los datos para hacer las diferencias comparables (en el marco de la clasificación, estas diferencias están bien definidos, ya que estos errores se encuentran en el mismo dominio, pero en el marco de la regresión no). Por ello, para hacer las diferencias comparables, se propone adoptar la diferencia normalizada DIFF propuesta en [ADH+09, GAH10], que se define como:

$$DIFF = \frac{Media\ (otro) - Media\ (referencia)}{Media\ (otro)} \quad (2.8)$$

donde *Media (x)* representa el ECM o el número final de reglas ($\#R_F$) obtenido por el algoritmo *x*. Esta diferencia expresa el porcentaje de mejora del algoritmo de referencia en el otro algoritmo.

Los resultados medios de los SBRDs iniciales obtenidos por WM y COR se presentan en la Tabla 2.3 como datos de referencia, donde ECM_{ENT} y ECM_{TST} representan el ECM obtenido en entrenamiento y test respectivamente y $\#R$ representa el número medio de reglas. El número de reglas $\#R$ son las mismas para ambos métodos debido al hecho de que los subespacios de entrada difusos se obtuvieron del mismo modo.

Tabla 2.3: Resultados iniciales obtenidos por WM y COR

		BR_{WM}		BR_{COR}	
Conjunto de Datos	**#R**	**ECM_{ENT}**	**ECM_{TST}**	**ECM_{ENT}**	**ECM_{TST}**
PLA	14.8	3.434	3.557	1.477	1.480
QUA	53.6	0.0258	0.0267	0.0178	0.0189
ELE	65.0	56135	56359	50711	54585
ABA	68.2	8.407	8.424	3.047	3.058
STP	122.8	9.074	9.042	5.254	5.279
WAN	156.0	16.063	16.403	7.150	7.357
WIZ	104.8	6.945	7.139	4.931	5.083
MOR	77.6	0.985	0.973	0.646	0.653
TRE	75.0	1.636	1.632	1.519	1.524

Para facilitar las comparativas, se han seleccionado parámetros estándar en vez de utilizar parámetros muy específicos para cada uno de los métodos. Además, para permitir que los algoritmos estudiados presenten una buena convergencia, se ha considerado un número suficientemente alto de evaluaciones (no se han observado cambios significativos incrementando dicho número de evaluaciones).

De esta forma, los valores de los parámetros de entrada considerados por los modelos de un solo objetivo son: tamaño de la población 50, 240000 evaluaciones, probabilidad de cruce 0.6 la y probabilidad de mutación por cromosoma 0.2.

Los valores de los parámetros de entrada considerados por los AGMOs son: tamaño de la población 200 individuos, tamaño de la población externa 61 individuos, 240000 evaluaciones y 0.2 como probabilidad de mutación.

Los parámetros a_{12}, a_{21} utilizados en el Método de Dominación Guiada de los AGMOs se determinaron después de varias pruebas, y fueron fijados a 0 y 16 respectivamente, es decir, fuertemente enfocada a la precisión.

2.5.2. Resultados y Análisis

En esta sección se presentan y analizan los resultados obtenidos por los dos modelos multiobjetivo propuestos (ODA - $BR_{COR\text{-}S})_{SP2}$ y (ODA - $BR_{COR\text{-}S})_{NS\text{-}II}$ comparándolos con los dos modelos multiobjetivo secuenciales BR_{WM} + (ODA - $S)_{SP2}$, y BR_{WM} + (ODA - $S)_{NS\text{-}II}$ y sus homólogos monoobjetivos BR_{WM} + (ODA – S) y (ODA - $BR_{COR\text{-}S}$).

Las Tablas 2.4, 2.5, 2.6 y 2.7 muestran los resultados obtenidos por estos modelos. Para los enfoques basados en un solo objetivo sólo se muestra los puntos más precisos mientras que para las propuestas multiobjetivo se muestran los resultados obtenidos en los tres puntos más representativos del Pareto en el plano interpretabilidad/precisión: el punto de máxima interpretabilidad (MAX $_{INT}$), el punto mediano (MEDIA $_{(INT\ /\ PRE)}$) y el punto de máxima precisión (MAX $_{PRE}$). Los resultados con mayor precisión se resaltan en negrita.

En las Tablas 2.4 y 2.5 se muestran los resultados obtenidos por los modelos secuenciales multiobjetivo BR_{WM} + (ODA - $S)_{SP2}$ y BR_{WM} + (ODA - $S)_{NS\text{-}II}$ comparados con su modelo equivalente monoobjetivo BR_{WM} + (ODA – S) y con su modelo de referencia BR_{WM} + S, en los conjuntos de datos de menor y mayor dimensionalidad, respectivamente.

En las Tablas 2.6 y 2.7 se muestran los resultados obtenidos por los modelos cooperativos multiobjetivo que se han propuesto (ODA - $BR_{COR\text{-}S})_{SP2}$ y (ODA - $BR_{COR\text{-}S})_{NS\text{-}II}$ comparados con su modelo monoobjetivo homólogo (ODA - $BR_{COR\text{-}S}$) y con sus modelos de referencia secuencial (BR_{COR} + S) y cooperativo $BR_{COR\text{-}S}$, en los conjuntos de datos de menor y mayor dimensionalidad, respectivamente.

Tabla 2.4: Resultados obtenidos por los modelos secuenciales multiobjetivo comparados con su modelo equivalente monoobjetivo y con el de referencia

Datos	Modelo	MAX INT			MEDIA INT / PRE			MAX PRE		
		#R	ECM_{ENT}	ECM_{TST}	*#R*	ECM_{ENT}	ECM_{TST}	*#R*	ECM_{ENT}	ECM_{TST}
PLA	BR_{WM}+*S*							11.6	2.403	2.477
	BR_{WM}+*ODA–S*	-	-			-		12.9	1.783	1.886
	BR_{WM}+$(ODA–S)_{SP2}$	11.3	1.651	1.722	12.1	1.641	1.712	13.3	1.639	**1.711**
	BR_{WM}+$(ODA–S)_{NS-II}$	11.4	1.654	1.722	12.1	1.646	1.715	13.3	1.642	1.713
QUA	BR_{WM}+*S*	-						29.9	0.022	0.023
	BR_{WM}+*ODA–S*	-						34.6	0.020	0.022
	BR_{WM}+$(ODA–S)_{SP2}$	25.3	0.0205	0.0221	27.7	0.0205	0.0220	30.5	0.020	0.022
	BR_{WM}+$(ODA–S)_{NS-II}$	25.7	0.0206	0.0222	27.9	0.0205	0.0220	30.8	0.020	**0.021**
ELE	BR_{WM}+*S*							41.8	41642	44037
	BR_{WM}+*ODA–S*							50.8	21804	26054
	BR_{WM}+$(ODA–S)_{SP2}$	44.1	22812	26448	45.2	22723	26201	46.9	22663	**25840**
	BR_{WM}+$(ODA–S)_{NS-II}$	44.2	23595	27639	45.0	23595	27639	45.5	23542	26593
ABA	BR_{WM}+*S*							25.6	5.083	5.049
	BR_{WM}+*ODA–S*							36.5	4.578	**4.600**
	BR_{WM}+$(ODA–S)_{SP2}$	20.8	4.627	4.654	22.8	4.617	4.648	25.2	4.614	4.647
	BR_{WM}+$(ODA–S)_{NS-II}$	21.5	4.654	4.685	23.5	4.645	4.680	25.2	4.635	4.670
STP	BR_{WM}+*S*							36.3	2.624	2.786
	BR_{WM}+*ODA–S*							61.4	1.364	**1.434**
	BR_{WM}+$(ODA–S)_{SP2}$	36.8	1.391	1.472	41.3	1.397	1.456	46.3	1.376	1.450
	BR_{WM}+$(ODA–S)_{NS-II}$	38.1	1.419	1.492	42.9	1.406	1.475	45.1	1.403	1.466
WAN	BR_{WM}+*S*							55.8	6.567	7.181
	BR_{WM}+*ODA–S*							90.0	3.566	4.340
	BR_{WM}+$(ODA–S)_{SP2}$	67.1	3.680	4.267	70.7	3.666	4.270	74.8	3.662	**4.265**
	BR_{WM}+$(ODA–S)_{NS-II}$	69.5	3.763	4.374	73.5	3.748	4.337	77.7	3.743	4.335
WIZ	BR_{WM}+*S*							46.0	3.036	3.506
	BR_{WM}+*ODA–S*							71.4	0.782	1.257
	BR_{WM}+$(ODA–S)_{SP2}$	55.5	0.858	1.282	59.7	0.840	1.254	64.5	0.834	**1.184**
	BR_{WM}+$(ODA–S)_{NS-II}$	57.7	0.903	1.456	61.6	0.887	1.414	61.6	0.888	1.388

Tabla 2.5: Resultados obtenidos por los modelos secuenciales multiobjetivo comparados con su modelo equivalente monoobjetivo y con el de referencia (conjuntos de datos de mayor dimensionalidad)

Datos	Modelo	MAX INT			MEDIA INT / PRE			MAX PRE		
		#R	ECM_{ENT}	ECM_{TST}	*#R*	ECM_{ENT}	ECM_{TST}	*#R*	ECM_{ENT}	ECM_{TST}
MOR	BR_{WM}+*S*							23.2	0.204	0.213
	BR_{WM}+*ODA–S*	-	-			-		41.7	0.082	**0.097**
	BR_{WM}+$(ODA–S)_{SP2}$	29.8	0.095	0.105	32.4	0.094	0.104	35.6	0.094	0.104
	BR_{WM}+$(ODA–S)_{NS-II}$	30.7	0.101	0.113	33.2	0.100	0.111	36.1	0.100	0.111
TRE	BR_{WM}+*S*							23.3	0.342	0.374
	BR_{WM}+*ODA–S*							41.4	0.078	**0.091**
	BR_{WM}+$(ODA–S)_{SP2}$	30.0	0.103	0.116	33.1	0.100	0.105	36.6	0.100	0.111
	BR_{WM}+$(ODA–S)_{NS-II}$	33.6	0.113	0.123	33.6	0.109	0.120	37.1	0.109	0.119

Tabla 2.6: Resultados obtenidos por los modelos cooperativos multiobjetivo comparados con su modelo homólogo monoobjetivo y modelos de referencia

Datos	Modelo	MAX $_{INT}$			MEDIA $_{INT\,/\,PRE}$			MAX $_{PRE}$		
		#R	ECM_{ENT}	ECM_{TST}	***#R***	ECM_{ENT}	ECM_{TST}	***#R***	ECM_{ENT}	ECM_{TST}
PLA	BR_{COR}+*S*		-			-		14.8	1.477	1.480
	$BR_{COR\text{-}S}$		-			-		14.8	1.477	1.480
	ODA–$BR_{COR\text{-}S}$		-			-		14.5	1.110	1.149
	(ODA–$BR_{COR\text{-}S})_{SP2}$	12.9	1.109	1.146	13.5	1.106	1.143	14.5	1.105	**1.145**
	(ODA–$BR_{COR\text{-}S})_{NS\text{-}II}$	12.9	1.119	1.156	13.5	1.113	1.149	14.5	1.111	1.149
QUA	BR_{COR}+*S*		-			-		45.8	0.0178	0.0189
	$BR_{COR\text{-}S}$		-			-		42.5	0.0178	0.0186
	ODA–$BR_{COR\text{-}S}$		-			-		48.0	0.0170	0.0186
	(ODA–$BR_{COR\text{-}S})_{SP2}$	25.1	0.0177	0.0190	31.4	0.0172	0.0186	38.3	0.0171	**0.0185**
	(ODA–$BR_{COR\text{-}S})_{NS\text{-}II}$	25.9	0.0177	0.0190	31.8	0.0173	0.0187	38.1	0.0172	**0.0185**
ELE	BR_{COR}+*S*		-			-		44.7	40763	43228
	$BR_{COR\text{-}S}$		-			-		40.8	38153	38926
	ODA–$BR_{COR\text{-}S}$		-			-		53.9	18981	**21122**
	(ODA–$BR_{COR\text{-}S})_{SP2}$	37.6	21648	24975	41.1	20366	23141	45.1	19959	22585
	(ODA–$BR_{COR\text{-}S})_{NS\text{-}II}$	38.0	22240	25099	41.7	20835	23445	46.1	20331	22763
ABA	BR_{COR}+*S*		-			-		44.4	2.829	2.908
	$BR_{COR\text{-}S}$		-			-		39.0	2.737	2.760
	ODA–$BR_{COR\text{-}S}$		-			-		50.3	2.345	**2.465**
	(ODA–$BR_{COR\text{-}S})_{SP2}$	29.9	2.443	2.557	32.7	2.425	2.548	36.1	2.417	2.541
	(ODA–$BR_{COR\text{-}S})_{NS\text{-}II}$	31.2	2.466	2.586	34.1	2.447	2.568	37.2	2.442	2.553
STP	BR_{COR}+*S*		-			-		52.9	2.222	2.433
	$BR_{COR\text{-}S}$		-			-		51.2	2.185	2.450
	ODA–$BR_{COR\text{-}S}$		-			-		72.5	1.101	1.188
	(ODA–$BR_{COR\text{-}S})_{SP2}$	41.9	1.116	1.207	46.2	1.106	1.192	50.9	1.103	**1.187**
	(ODA–$BR_{COR\text{-}S})_{NS\text{-}II}$	45.1	1.143	1.225	50.0	1.131	1.210	55.4	1.128	1.204
WAN	BR_{COR}+*S*		-			-		73.4	4.113	4.534
	$BR_{COR\text{-}S}$		-			-		63.0	3.926	4.562
	ODA–$BR_{COR\text{-}S}$		-			-		101.9	1.279	1.755
	(ODA–$BR_{COR\text{-}S})_{SP2}$	68.5	1.208	1.642	75.6	1.161	1.601	83.0	1.148	**1.590**
	(ODA–$BR_{COR\text{-}S})_{NS\text{-}II}$	68.5	1.287	1.769	75.5	1.234	1.674	83.1	1.219	1.654
WIZ	BR_{COR}+*S*		-			-		52.1	3.136	3.637
	$BR_{COR\text{-}S}$		-			-		46.8	2.948	3.369
	ODA–$BR_{COR\text{-}S}$		-			-		75.2	0.691	**0.952**
	(ODA–$BR_{COR\text{-}S})_{SP2}$	51.3	0.769	1.388	58.2	0.717	1.189	65.7	0.704	1.101
	(ODA–$BR_{COR\text{-}S})_{NS\text{-}II}$	53.0	0.781	1.206	59.4	0.738	1.145	66.3	0.725	1.076

Tabla 2.7: Resultados obtenidos por los modelos cooperativos multiobjetivo comparados con su modelo homólogo monoobjetivo y modelos de referencia (conjuntos de datos de mayor dimensionalidad)

Datos	Modelo	MAX $_{INT}$			MEDIA $_{INT / PRE}$			MAX $_{PRE}$		
		#R	ECM_{ENT}	ECM_{TST}	*#R*	ECM_{ENT}	ECM_{TST}	*#R*	ECM_{ENT}	ECM_{TST}
MOR	$BR_{COR}+S$		-			-		26.6	0.1413	0.1466
	BR_{COR-S}		-			-		25.0	0.135	0.140
	$ODA–BR_{COR-S}$		-			-		50.2	0.0412	**0.049**
	$(ODA–BR_{COR-S})_{SP2}$	36.6	0.048	0.053	39.8	0.046	0.052	43.5	0.046	0.052
	$(ODA–BR_{COR-S})_{NS-II}$	36.5	0.052	0.058	40.1	0.050	0.056	43.9	0.050	0.056
TRE	$BR_{COR}+S$		-			-		25.8	0.337	0.364
	BR_{COR-S}		-			-		25.0	0.311	0.342
	$ODA–BR_{COR-S}$		-			-		43.1	0.082	**0.097**
	$(ODA–BR_{COR-S})_{SP2}$	30.6	0.102	0.121	33.8	0.099	0.117	37.6	0.098	0.116
	$(ODA–BR_{COR-S})_{NS-II}$	30.7	0.113	0.130	33.8	0.110	0.125	37.3	0.108	0.124

Aunque el comportamiento del modelo puede ser observado en las tablas anteriores, es más fácil llegar a conclusiones una vez realizado el análisis estadístico para evaluar y determinar si existen diferencias significativas entre los resultados obtenidos. Las Tablas 2.8, 2.9 y 2.10 muestran los resultados del test de Wilcoxon sobre el ECM_{TST} y el *#R* para el punto de máxima precisión (MAX $_{PRE}$).

Tabla 2.8 Test de Wilcoxon para comparar modelos secuenciales: algoritmo monoobjetivo BR_{WM}+ODA–S (R^+) comparado con algoritmo multiobjetivo $BR_{WM}+(ODA–S)_{SP2\ y\ NS-II}$ (R^-) en ECM_{TST} y *#R*

Comparación	Medida	R^+	R^-	*Hipótesis* ($\alpha = 0.10$)	*p-valor*
$BR_{WM}+ODA–S$ *vs* $BR_{WM}+(ODA–S)_{SP2}$	ECM_{TST}	19	17	Aceptada	1.000
	#R	1	44	Rechazada	0.011
$BR_{WM}+ODA–S$ *vs* $BR_{WM}+(ODA–S)_{NS-II}$	ECM_{TST}	32	13	Aceptada	0.260
	#R	1	44	Rechazada	0.011

Tabla 2.9 Test de Wilcoxon para comparar modelos cooperativos: algoritmo monoobjetivo ODA-BR_{COR-S} (R^+) comparado con algoritmo multiobjetivo propuesto $(ODA–BR_{COR-S})_{SP2\ y\ NS-II}$ (R^-) en ECM_{TST} y *#R*

Comparación	Medida	R^+	R^-	*Hipótesis* ($\alpha = 0.10$)	*p-valor*
$ODA–BR_{COR-S}$ *vs* $(ODA –BR_{COR-S})_{SP2}$	ECM_{TST}	32	13	Aceptada	0.260
	#R	0	36	Rechazada	0.008
$ODA –BR_{COR-S}$ *vs* $(ODA –BR_{COR-S})_{NS-II}$	ECM_{TST}	25	11	Aceptada	0.327
	#R	0	36	Rechazada	0.008

Tabla 2.10 Test de Wilcoxon para comparar los modelos secuenciales (R^+) con los modelos cooperativos (R^-), en un solo objetivo (BR_{WM}+ODA–S vs ODA-BR_{COR-S}) y en multiobjetivo (BR_{WM}+(ODA–S)$_{SP2\ y\ NS-II}$ vs ODA–BR_{COR-S})$_{SP2\ y\ NS-II}$) en ECM_{TST} y #*R*

Comparación	Medida	R^+	R^-	*Hipótesis* ($\alpha = 0.10$)	*p-valor*
BR_{WM}+ ODA –S vs ODA –BR_{COR-S}	ECM_{TST}	1	44	Rechazada	0.011
	#R	45	0	Rechazada	0.008
BR_{WM}+(ODA –S)$_{SP2}$ vs (ODA –BR_{COR-S})$_{SP2}$	ECM_{TST}	1	44	Rechazada	0.011
	#R	44	1	Rechazada	0.011
BR_{WM}+(ODA –S)$_{NS-II}$ vs (ODA –BR_{COR-S})$_{NS-II}$	ECM_{TST}	1	44	Rechazada	0.011
	#R	44	1	Rechazada	0.011

La Tabla 2.8 muestra los resultados cuando se comparan los modelos secuenciales entre sí, concretamente el modelo de un solo objetivo BR_{WM}+ODA–S con los modelos multiobjetivo BR_{WM}+(ODA–S)$_{SP2}$ y BR_{WM}+(ODA–S)$_{NS-II}$.

La precisión es similar entre ellos, pero los multiobjetivo son más interpretables. La hipótesis nula para el test de Wilcoxon fue rechazada con un muy pequeño *p-valor* para el #*R*, que apoya nuestra conclusión con un alto grado de confianza. Sin embargo, son similares en precisión. En este caso, la hipótesis nula del test de Wilcoxon es aceptada, por lo que no se puede concluir que los resultados obtenidos por los modelos multiobjetivo son diferentes en ECM_{TST}, aunque éstos sean mejores más veces (R^+), por ser las diferencias pequeñas.

La Tabla 2.9 es similar a la Tabla 2.8, pero cuando se comparan los modelos cooperativos entre sí, concretamente el modelo de un solo objetivo ODA-BR_{COR-S} contra los modelos multiobjetivo propuestos en este trabajo.

Al igual que antes, los enfoques multiobjetivo y monoobjetivo obtienen una precisión similar, pero aunque los modelos de un solo objetivo, orientados a precisión, obtienen una precisión muy buena utilizando el mismo número de evaluaciones que los multiobjetivo, su interpretabilidad está muy lejos de la solución más precisa del modelo multiobjetivo. Por lo tanto, el uso de los enfoques multiobjetivo proporciona una importante ventaja ya que permite obtener soluciones con la misma precisión pero mucho más interpretables.

La Tabla 2.10 muestra una comparación interesante entre los modelos secuenciales y los modelos cooperativos entre sí, tanto en un solo objetivo como en multiobjetivo, concretamente en monoobjetivo, el modelo secuencial BR_{WM}+ODA–S contra el modelo cooperativo ODA-BR_{COR-S} y los modelos secuenciales monoobjetivo BR_{WM}+(ODA–S)$_{SP2}$ y BR_{WM}+(ODA–S)$_{NS-II}$ contra los modelos cooperativos multiobjetivo (ODA–BR_{COR-S})$_{SP2}$ y (ODA–BR_{COR-S})$_{NS-II}$.

Ahora, en ambos casos, la hipótesis nula asociada con el test de Wilcoxon es rechazada ($p < \alpha$), pero, el resultado es justo al contrario entre $\#R$ y ECM_{TST}. Por lo tanto, se puede concluir que los resultados obtenidos por estos métodos son estadísticamente diferentes en el ECM_{TST} y en el $\#R$ y que los modelos cooperativos son más precisos que los secuenciales, mientras que con respecto a la interpretabilidad, los modelos secuenciales son más interpretables. La explicación a esto, es que BR_{COR-S} produce BRs más precisas que BR_{WM}+S, gracias a la cooperación entre las reglas, mientras que el modelo secuencial usa un mecanismo de selección de regla más eficaz que le permite obtener BRs más compactas.

Por lo tanto, podemos concluir que el modelo cooperativo multiobjetivo propuesto obtiene:

- Una mayor interpretabilidad que las propuestas de un solo objetivo con similar precisión.

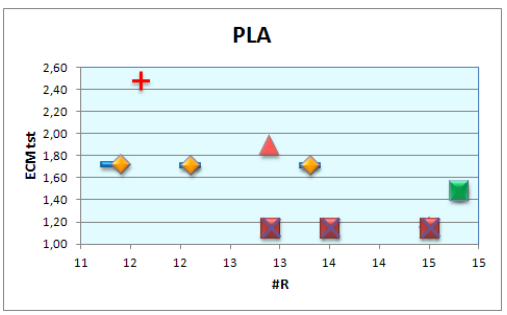
- Una mayor precisión que las propuestas secuenciales multiobjetivo.

2.5.3. Análisis Gráfico de los Frentes de Pareto

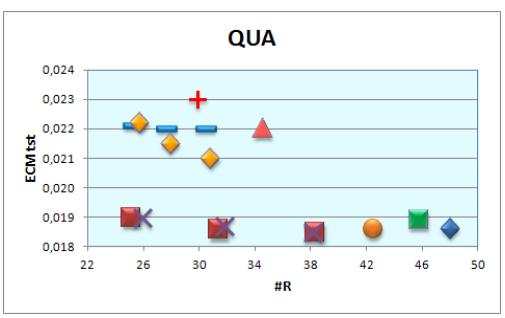
La Figura 2.3 muestra las soluciones medias obtenidas en los nueve conjuntos de datos por los diferentes algoritmos presentes en el estudio (tanto monoobjetivos como multiobjetivo).

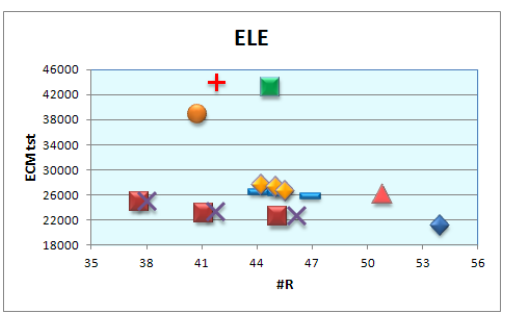
En el caso de los modelos basados en un solo objetivo, para cada conjunto de datos se muestra la solución obtenida por cada algoritmo monoobjetivo, calculada como el valor medio de las 30 pruebas del ECM y del número de reglas en el conjunto de test.

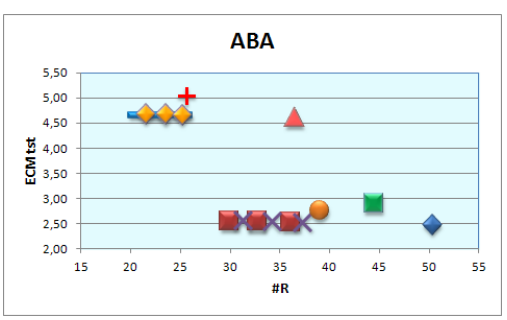
En el caso de los modelos con enfoque multiobjetivo y con idea de hacer las gráficas más legibles, en lugar de mostrar todo el frente de Pareto medio obtenido por cada uno de los AGMOs, se ha optado por mostrar de cada Pareto sólo las soluciones medias obtenidas en los tres puntos más representativos de los objetivos interpretabilidad-precisión: el punto de máxima interpretabilidad (MAX_{INT}), el mediano ($MEDIA_{(INT\ /\ PRE)}$) y el más preciso (MAX_{PRE}) en test, calculados al igual que antes, como la media del ECM y del número de reglas en el conjunto de test correspondientes a las 30 pruebas realizadas para cada conjunto de datos.

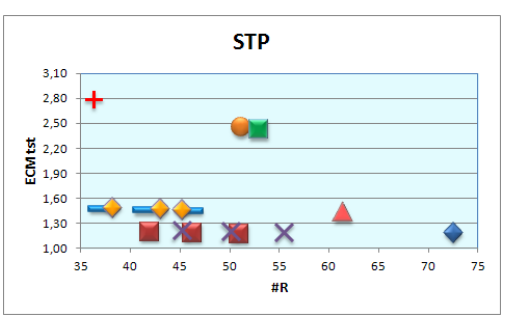


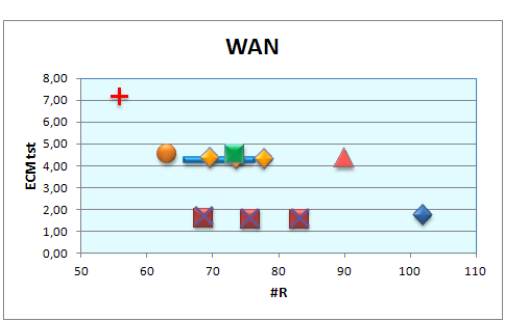
MODELO	SIMBOLO
$ODA\text{-}BR_{COR\text{-}S}$	
$(ODA\text{-}\ BR_{COR\text{-}S})_{SP2}$	
$(ODA\text{-}\ BR_{COR\text{-}S})_{NSII}$	
$BR_{COR\text{-}S}$	
$BR_{WM}+S$	
$BR_{WM}+\ ODA\text{-}S$	
$BR_{WM}+(\ ODA\text{-}S)_{SP2}$	
$BR_{WM}+(\ ODA\text{-}S)_{NSII}$	
$BR_{COR}+S$	

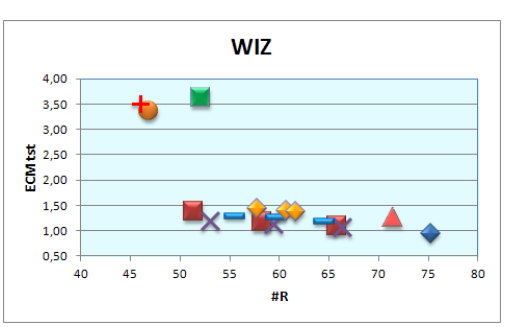












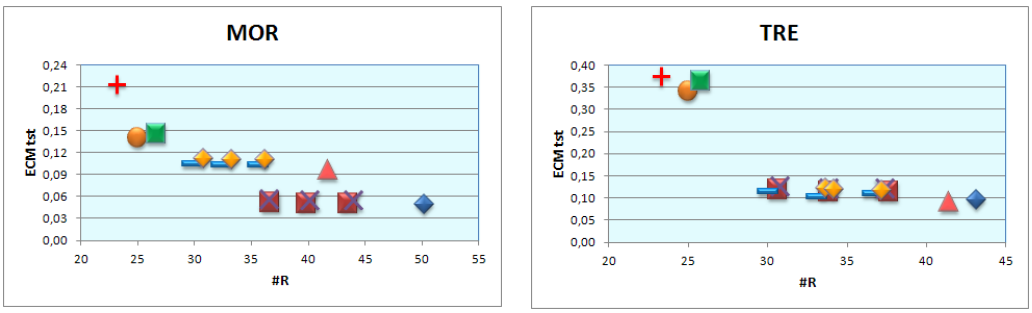


Figura 2.3: Frentes de Paretos medios obtenidos en todos los problemas

Observando la Figura 2.3, podemos indicar lo siguiente:

- Los resultados presentados por los dos modelos cooperativos propuestos, ambos basados en los dos AEs, están en general, por debajo de los otros modelos de estudio, es decir, son más precisos.

- Sin embargo, las tres soluciones de los AGMOs, la más interpretable (MAX $_{INT}$), la mediana (MEDIA $_{(INT\ /\ PRE)}$) y la más precisa (MAX $_{PRE}$) parecen estar próximas, y se colocan en la zona de más precisión. La razón es que no estamos usando el SPEA2 y NSGA-II estándar, sino una versión mejorada más precisa de ellas basados en el método de Dominación Guiada explicada en la Sección 2.4.1.2 El proceso de búsqueda está centrado en la zona reducida de frente de Pareto con mayor precisión, y los resultados obtenidos son los extremos y el punto central de dicha zona deseada.

- La cooperación entre los ODAs y la BR hace que los modelos cooperativos propuestos muestren soluciones diferentes, más precisas que las obtenidas con los ODAs con modelos secuenciales, como se concluyó en la Sección 2.5.2.

2.6. Sumario

En este capítulo, hemos propuesto un modelo cooperativo evolutivo multiobjetivo de aprendizaje con el doble objetivo de minimizar el error del sistema y el número de reglas, donde los ODAs, incluyendo la inferencia y defuzzificación, se aprenden de forma conjunta y simultánea con la BR.

El modelo evolutivo multiobjetivo de aprendizaje propuesto se ha implementado empleando dos modelos diferentes de algoritmos multiobjetivo basados en dos conocidos AGMO de segunda generación (SPEA2 y NSGA-II), a los que se les ha aplicado unas mejoras para centrar el proceso de búsqueda en la región más precisa del frente de Pareto.

Para analizar el modelo propuesto se ha llevado a cabo un estudio experimental con nueve conjuntos de datos reales de regresión de diferentes características, en el que se ha comparado la metodología cooperativa multiobjetivo propuesta con diferentes modelos secuenciales multiobjetivo y diferentes modelos de un solo objetivo basados en la precisión.

Para validar los resultados obtenidos se ha realizado un análisis estadístico en el que se han utilizado test estadísticos no paramétricos por pares, en los que se ha tenido en cuenta los tres puntos más representativos del Pareto obtenido por cada uno de los modelos multiobjetivos comparados: el punto más interpretable, el mediano y el más preciso. Del estudio estadístico se han derivado las siguientes conclusiones:

- El modelo cooperativo multiobjetivo propuesto mejora de forma significativa la interpretabilidad respecto a los modelos de un solo objetivo, basados en la precisión, a la vez que consigue obtener soluciones con un nivel similar en precisión.

- La precisión alcanzada por el modelo cooperativo multiobjetivo propuesto es mayor que la obtenida por los modelos secuenciales multiobjetivos.

Se ha realizado un análisis gráfico de los frentes de Pareto obtenidos por los diferentes algoritmos multiobjetivos presentes en el estudio en el que se ha representado también la solución alcanzada por los modelos monoobjetivos. Los gráficos ponen de manifiesto que las soluciones alcanzadas por el modelo propuesto son más precisas que las otras, aunque no siempre más interpretables, debido a que los AGMOs utilizados en nuestra propuesta son una versión mejorada de los utilizados por las otras propuestas multiobjetivos, que hace que el proceso de búsqueda se centre en una zona reducida del frente de Pareto, aquella donde la precisión es más alta, en lugar del frente de Pareto completo como hace los AGMO originales usados por las otras propuestas.

Capítulo 3

Un Mecanismo para Mejorar la Interpretabilidad de los Sistemas Difusos Lingüísticos con Defuzzificación Adaptativa Basado en el Uso de Algoritmos Genéticos Multiobjetivo

En el marco del equilibrio entre precisión e interpretabilidad, en el capítulo anterior se propuso un modelo evolutivo multiobjetivo basado en el aprendizaje conjunto de los parámetros de los operadores adaptativos del SI y de la ID junto con el aprendizaje y selección de la BR.

La utilización de parámetros en el MI, como la que se ha propuesto en el capítulo anterior para obtener mejoras en el equilibrio entre precisión e interpretabilidad, beneficia la precisión pues permite adaptar, bien la agregación, bien la defuzzificación e incluso ambas, para cada regla. Sin embargo, por otro lado afecta a la interpretabilidad en tanto introduce esos parámetros.

Así como medir la precisión de un sistema difuso es tradicionalmente una labor generalmente sencilla, ocurre lo contrario con la propiedad de la

intepretabilidad. En los últimos años, diversos autores han propuesto diferentes enfoques y expresiones que se describieron en el Capítulo 1 para contribuir a avanzar en este terreno. En este capítulo abordamos también esta cuestión desde el punto de vista de los operadores adaptativos. Proponemos una serie de medidas relevantes que se combinan en un índice para los SBRDs adaptativos más empleados que son los que utilizan un parámetro o peso asociado a cada regla que se emplea para adaptar la defuzzficación, y posteriormente mostramos su utilidad en el proceso de obtener sistemas con mejor equilibrio entre precisión e interpretabilidad usando dicho índice como objetivo en el AGMO que los optimiza.

Como en cada capítulo, desarrollamos un estudio experimental que permite verificar los modelos propuestos con distintos conjuntos de datos validado mediante test estadísticos no paramétricos.

El capítulo se organiza en las siguientes secciones: En primer lugar, en la Sección 3.1 se introduce y justifica el problema a resolver. En la Sección 3.2 se presenta las bases del mecanismo de mejora de la interpretabilidad propuesto y se describen las métricas utilizas para implementar dicho mecanismo. En la Sección 3.3 se muestran los dos modelos multiobjetivo utilizados atendiendo particularmente a sus principales características y operadores genéticos. En la Sección 3.4 se llevan a cabo dos estudios experimentales para mostrar la utilidad de los dos modelos anteriormente descritos, y por último en la Sección 3.5 se presentan las conclusiones junto con un breve resumen del capítulo.

3.1. Introducción

Como ya sabemos, el uso de parámetros en el operador de defuzzificación introduce una serie de valores o pesos asociados indirectamente a cada regla de la BR, que permiten mejorar la precisión, a costa de aumentar la complejidad y por tanto decrementar la interpretabilidad del sistema [AMGR09, CHMP04, GAH11, MF08, ZG08], por lo que sería deseable minimizar los elementos que aumenten dicha complejidad.

En este sentido, debe tenerse en cuenta que si bien en la literatura científica hay diversos estudios e índices propuestos para medir la interpretabilidad (véase Sección 1.2.2.1 del Capítulo 1), la mayoría de los trabajos son propuestas basadas en las funciones de pertenencia (semántica) o basadas en alguna propiedad de la BR (complejidad), pero ninguna de ellas se

basa directamente en el SI o en la ID adaptativa. Por ello se hace necesario definir una o varias medidas para cuantificar la interpretabilidad de los SBRDs con defuzzificación adaptativa, las cuales pueden ser usadas en un modelo multiobjetivo para optimizar la interpretabilidad de los parámetros adaptivos cuando el proceso de aprendizaje es llevado a cabo.

Consecuentemente, debido al interés de los métodos de defuzzificación adaptativos desde el punto de vista de la precisión, y teniendo en cuenta su antes mencionada desventaja con respecto a la interpretabilidad, se ha considerado interesante trabajar en el equilibrio entre ambos aspectos, proponiendo un mecanismo para mejorar la interpretabilidad (en el sentido de la complejidad) de los sistemas difusos lingüísticos con defuzzificación adaptativa. El mecanismo propuesto se basa en tres objetivos relacionados con la interpretabilidad:

1) reducir el número total de reglas, eliminando aquellas con peso cercano a cero,

2) reducir el número de reglas con peso asociado, eliminando el peso a aquellas reglas con peso cercano a uno y

3) reducir las reglas disparadas de forma conjunta con cada ejemplo,

todo ello mediante el uso de varias métricas y un índice de interpretabilidad propuesto.

Un aspecto interesante a destacar es que la defuzzificación adaptativa y por lo tanto la propuesta desarrollada en este trabajo puede ser usada junto con otras metodologías para mejorar la interpretabilidad y la precisión del sistema.

3.2. Una Nueva Propuesta para Mejorar la Interpretabilidad de la Defuzzificación Adaptativa

Una vez planteado el problema de la interpretabilidad en los sistemas con defuzzificación adaptativa y los objetivos que se pretenden optimizar en el nuevo modelo, introduciremos en esta sección una parte esencial del mismo como es el mecanismo y métricas usadas, antes de exponer el nuevo modelo evolutivo propuesto en la siguiente sección.

Comenzaremos por describir un modelo básico y más adelante presentaremos un modelo adaptativo mejorado también basado en los mismos principios. Tanto el modelo básico como el modelo adaptativo serán explicados en detalle en la Sección 3.3.

3.2.1. Descripción del Mecanismo Propuesto

En este apartado, vamos a describir los elementos del mecanismo propuesto para mejorar la interpretabilidad, que permiten reducir las reglas con peso y las reglas activadas simultáneamente en cada ejemplo, todo con el fin de reducir la complejidad del sistema y a favor de la compacidad de la BR. En las siguientes subsecciones se describirán con más detalle estos mecanismos usados.

3.2.1.1. Mecanismo de Selección Global de Reglas Basado en un Umbral Bajo

Este mecanismo se basa en la idea de que las reglas con pesos cercanos a cero son las que tienen una menor influencia en el resultado final, pues interpretamos un peso bajo con una regla casi prescindible.

Por lo tanto, proponemos evitar el uso de reglas con valores de peso más bajos, eliminando la contribución de dichas reglas durante el proceso de aprendizaje de los pesos para que el modelo evolutivo pueda ajustar los pesos de las restantes reglas para adaptarse a dicha eliminación, de forma que la precisión no se vea perjudicada. Esta idea es aplicada estableciendo un umbral, U_B (umbral bajo) que define el nivel de corte para eliminar reglas.

Con este mecanismo de selección de reglas pretendemos incrementar la interpretabilidad del sistema debido a que la *interpretabilidad de alto nivel* [ZG08] y *la complejidad del sistema a nivel de BR* [GAH11] mejoran, haciendo más legible la estructura del sistema [AMGR09]. Además, dicho mecanismo también permite mejorar la precisión del sistema debido a que puede mejorar la cooperación entre las reglas de la BR.

3.2.1.2. Mecanismo para Reducir el Número de Reglas con Pesos Basado en un Umbral Alto

Este mecanismo se basa en la idea de que los valores de peso cercanos a uno pertenecen a las reglas más importantes de la BR. Por esta razón, consideramos que estas reglas podrían utilizarse sin necesidad de ninguna ponderación o peso, reduciendo así la complejidad del sistema (mejor cuanto mayor sea el número de reglas sin ponderar).

Por lo tanto proponemos la eliminación de estos valores durante el proceso de aprendizaje de los parámetros de la defuzzificación utilizando un nuevo umbral en segundo lugar, U_A (umbral alto). Eliminar los pesos en las reglas más importantes en la fase de aprendizaje permite al sistema readaptarse a dichos cambios, reajustando los pesos de las restantes reglas, a fin de que la precisión no se vea afectada.

Esta técnica reduce la complejidad, disminuye el impacto sobre la interpretabilidad semántica de la defuzzificación adaptativa [GAH11, NK98] y mejora la *interpretabilidad descriptiva* [AMGR09] que se relaciona con la comprensión del modelo.

Para aplicar estos mecanismos los parámetros de la defuzzificación adaptativa con funcional tipo producto (véase Sección D.1.4.2 Apéndice D) actuarán en el rango [0, 1]. Además debemos establecer los valores de los 2 umbrales antes mencionados que definen respectivamente qué reglas son eliminadas y cuáles actúan sin peso asociado.

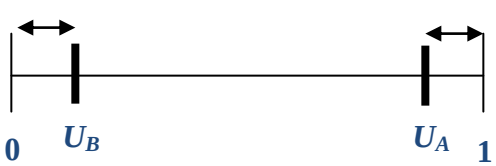


Figura 3.1: Rango de los parámetros de la defuzzificación adaptiva y umbrales para eliminar reglas y pesos asociados a las reglas

3.2.2. Métricas Propuestas

Una vez descritos los mecanismos de mejora de la interpretabilidad que va a utilizar nuestra propuesta, vamos a exponer a continuación las métricas que vamos a utilizar para medirla. Por ello, con el fin implementar dichos mecanismos de mejora, proponemos las métricas descritas a continuación.

3.2.2.1. Número de Reglas Final, ($\#R_F$)

Esta métrica mide el número de reglas de la BR inicial que no han sido eliminadas durante el proceso evolutivo de aprendizaje y selección de reglas. Su expresión es:

$$Minimizar\ \#R_F = \#R - \#(reglas\ con\ pesos\ menores\ de\ U_B) \quad (3.1)$$

donde $\#R$ es el número de reglas iniciales de la BR, antes de comenzar el proceso de aprendizaje de los parámetros de la defuzzificación adaptativa, y $\#(reglas\ con\ pesos\ menores\ de\ U_B)$ el número de reglas cuyo peso es cercano a cero.

Esta métrica está basada en la primera idea: aquellas reglas con pesos cercanos a 0 (entre 0 y U_B) representan reglas con baja influencia que puede indicar una regla prescindible, continuando el sistema evolutivo con el aprendizaje del resto del sistema sin ella.

3.2.2.2. Número de Reglas con Peso, ($\#R_P$)

Esta métrica mide el número de reglas de la BR con un peso asociado. Su expresión es la siguiente:

$$Minimizar\ \#R_P = \#R - \#(reglas\ con\ pesos\ mayores\ de\ U_A) \quad (3.2)$$

donde $\#R$ es el número de reglas iniciales de la BR, antes de comenzar el proceso de aprendizaje de los parámetros de la defuzzificación adaptativa, y $\#(reglas\ con\ pesos\ mayores\ de\ U_A)$ el número de reglas con peso cercano a uno.

Esta métrica está basada en la segunda idea: aquellas reglas con pesos cercanos a 1 (entre U_A y 1) representan reglas importantes que podrían ser usadas sin peso, eliminando dicho valor, reduciendo así la complejidad del sistema.

3.2.2.3. Número Medio de Reglas Activadas por cada Ejemplo, (MR_{AC})

La reducción del número final de reglas y del número de reglas con peso mejora la interpretabilidad del sistema. Sin embargo, la interpretabilidad del sistema también depende del número de reglas disparadas simultáneamente por cada ejemplo, de forma que el sistema es más interpretable cuanto menor es el número de reglas activadas al mismo tiempo [CL00]. Para tener este hecho en cuenta, utilizaremos un nuevo índice que medirá el número medio de reglas activadas por cada ejemplo, (MR_{AC}).

Su expresión es la siguiente:

$$Minimizar \quad MR_{AC} = \frac{\sum_{j=1}^{N} \# R^j{}_{AC}}{N} \tag{3.3}$$

donde N es el número de ejemplos y $\#R^j{}_{AC}$ es el número de reglas activadas por el ejemplo j.

De este modo, reduciendo MR_{AC}, la interpretabilidad *de alto nivel* [ZG08] y la interpretabilidad *semántica* [GAH11] mejoran, así como la interpretabilidad *descriptiva* [AMGR09].

3.2.3. Un Índice de Interpretabilidad Global Basado en la Agregación de Métricas

A fin de evitar tener muchos objetivos a optimizar (en principio, uno por cada métrica utilizada) en el modelo multiobjetivo, se propone la agregación de dos métricas, $\#R_P$ y MR_{AC}, en un índice global basado en la media aritmética de ambos, el cual se denota como $R_P\_MR_{AC}$, y cuya expresión es la siguiente:

$$Minimizar \quad R_P\_MR_{AC} = \frac{\# R_P + MR_{AC}}{2} \tag{3.4}$$

Para calcular la expresión final de este nuevo índice, primero normalizamos ambas métricas en el intervalo [0, 1], de forma que el valor de $R_P\_MR_{AC}$ varía entre 0 (el más alto nivel de interpretabilidad) y 1 (el más bajo).

La minimización de este índice recupera la interpretabilidad perdida debido al efecto de la defuzzificación adaptativa, mientras que mejora la interpretabilidad del sistema global, el cual podría tener un margen de mejora debido al efecto del aprendizaje y optimización de la BR.

3.3. Modelos Evolutivos Multiobjetivo Propuestos

Esta sección describe los dos modelos evolutivos propuestos que utilizan un AGMO basado en el conocido NSGA-II [DAPM02] con el mecanismo para mejorar la interpretabilidad de los sistemas con defuzzificación adaptativa descritas en el apartado anterior.

Como comentamos anteriormente, la introducción del índice de interpretabilidad global (compuesto por la agregación de las métricas antes explicadas) en el proceso de aprendizaje es un elemento importante, ya que el sistema debe readaptarse tanto a la eliminación de reglas con baja aportación, como a la de algunos de los pesos en las reglas más importantes.

Las métricas en las que se basa dicho índice requieren el establecimiento de dos umbrales U_B y U_A (umbral bajo y alto) que definen respectivamente los niveles de corte para eliminar reglas y el peso asociado a una regla. De esta forma, los pesos que desciendan por debajo de un umbral cercano a 0, serán reglas a eliminar, mientras que los valores del peso que superen un determinado valor próximo a 1, serán evaluados como 1, eliminándose el peso asociado a dicha regla. Dependiendo de si los umbrales con los que trabajamos son fijados a priori o son aprendidos por el sistema damos lugar a dos modelos evolutivos diferentes: uno que utiliza umbrales fijos y otro que utiliza umbrales adaptativos.

3.3.1. Un Primer Modelo: Umbrales Fijos

El primer modelo presentado aprende los parámetros de la defuzzificación adaptativa, usando el mecanismo de mejora de la interpretabilidad descrito anteriormente con unos valores de umbrales fijos.

A continuación, exponemos sus principales componentes y parámetros.

3.3.1.1. Objetivos y Umbrales

A fin de minimizar el número de objetivos a optimizar por el AGMO sólo vamos a utilizar como objetivos en el presente modelo el ECM y el índice de interpretabilidad global $R_P\_MR_{AC}$. Aunque la minimización del número de reglas finales $\#R_F$ parece también un objetivo candidato claro a añadir a los otros dos, éste no va a ser añadido debido al hecho de que el $\#R_F$ se minimiza también al minimizar el índice $R_P\_MR_{AC}$, al incluir éste el número de reglas con peso $\#R_P$ y utilizar el mecanismo de umbral bajo U_B para reducir el número de reglas.

Por lo tanto, los objetivos que utiliza el modelo multiobjetivo son dos, en lugar de tres:

1. maximización de la interpretabilidad: minimizando el índice $R_P\_MR_{AC}$.

2. maximización de la precisión: minimizando el habitual ECM (véase Expresión 2.7 del Capítulo 2), que mide la precisión del sistema

A diferencia del modelo del Capítulo 2, el modelo difuso que proponemos ahora usa la t-norma del mínimo como operador de conjunción y de inferencia. Como método de defuzzificación adaptativo utiliza el mismo del capítulo anterior: defuzzificación adaptativa B-FITA tipo producto (véase Expresión D.14 del Apéndice D).

En cuanto a los umbrales U_A y U_B (umbral alto y bajo) considerados para los índices $\#R_P$ y $\#R_F$, se han seleccionado los siguientes:

- Cuando el valor del parámetro es ≥ 0.9 se considera que dicho parámetro es 1 y por tanto dicha regla no tiene peso.

- Cuando el valor del parámetro sea ≤ 0.1, evaluaremos dicho parámetro como 0 y dicha regla será eliminada.

Ambos umbrales (U_A=0.9 y U_B=0.1) han sido escogidos empíricamente tras la realización de distintas pruebas.

3.3.1.2. Esquema de Codificación y Población Inicial

Para representar los parámetros del modelo evolutivo propuesto se ha utilizado un esquema de codificación real, donde N es el número de parámetros β_i (de la defuzzificación adaptativa, una por cada regla R_i de la BR inicial). Cada uno toma valores en el intervalo [0, 1].

$$C = (\beta_1, \ldots, \beta_N) \mid \beta_i \in \{0, 1\} \tag{3.5}$$

La población inicial se obtiene de la siguiente manera: Un individuo de la población inicial tiene todos los genes inicialmente fijados a 1 a fin de comenzar el proceso evolutivo con todas las reglas sin peso. Los restantes individuos de la población son inicializados aleatoriamente.

3.3.1.3. AGMO Basado en NSGA-II

En este modelo, hemos empleado un AGMO basado en el NSGA-II original [DAPM02], con algunas adaptaciones, ya que no nos interesa todo el conjunto de soluciones no dominadas, como por ejemplo aquellas con un error muy alto y un número de reglas muy bajo, sino aquellas que tienen un razonable equilibrio entre precisión e interpretabilidad, mejor cuanto mayor sea ambas características simultáneamente. La razón por la que nos hemos decantado por el NSGA-II es ya conocida: es uno de los AGMO de segunda generación más conocidos y utilizados en la literatura para la resolución de problemas multiobjetivo.

El algoritmo propuesto hace uso del mecanismo de selección de NSGA-II, pero con el fin de mejorar la capacidad de búsqueda del algoritmo, el esfuerzo de búsqueda se concentra en la zona del Pareto con las soluciones de mayor precisión.

Para ello, en cada etapa del algoritmo, forzamos a que en la población utilizada como base para la selección de la próxima generación haya un porcentaje de individuos dominados (aquellos con menor error), que inicialmente será del 80% y que irá disminuyendo hasta su desaparición a medida que transcurren las generaciones. En otras palabras, forzamos a que la población no esté formada exclusivamente por soluciones no dominadas (si esto fuera posible), si no que mantenemos en dicha población las mejores soluciones dominados (aquellas con menor ECM), aumentando de forma gradual el número de soluciones no dominadas permitidas del 20% al principio al 100% al final, de forma que, a medida que el AE avanza el número de

soluciones dominadas presentes en la población disminuye y el número de soluciones no dominadas aumenta.

La idea que se persigue con esto es que al mantener en el proceso evolutivo soluciones dominadas con los ECM más pequeños (algunas soluciones dominadas pueden tener un ECM menor que algunas de las soluciones no dominadas, que lo son por ser más interpretables) el proceso de búsqueda converja hacia la zona del Pareto con mayor precisión. Además al ir disminuyendo el número de soluciones dominadas y aumentando el número de soluciones no dominadas a medida que el algoritmo evoluciona, conseguimos que dentro de la zona del Pareto con mayor precisión la búsqueda se vaya diversificando explorando por tanto también la parte más interpretable de dicha zona.

3.3.1.4. Cuestiones Importantes relacionados con la Implementación

Este apartado plantea dos cuestiones relacionadas con la implementación.

La primera está relacionada con la integridad o completitud de las BRs [CH97] obtenidas con esta metodología. Para asegurar esta propiedad, es decir, para asegurar que la BR generada cubra todos los ejemplos del conjunto de datos, a la hora de eliminar una regla de la BR, se comprueba si dicha regla es la única que cubre un ejemplo específico, de forma que si es así, la regla no es eliminada, ya que el efecto sobre el SBRD sería negativo [DHR93, Lee90].

La segunda cuestión está relacionado con la implementación de los umbrales y la habilidad del mecanismo de búsqueda: durante el proceso evolutivo, el valor de cada uno de los parámetros del cromosoma podría implicar la eliminación de una regla, o eliminar el peso de una regla, de forma que cuando el valor de un parámetro es ≤ 0.1 evaluamos dicho parámetro como 0 y cuando es ≥ 0.9 lo evaluamos como 1, pero el valor original se mantiene dentro del cromosoma con el fin de evitar afectar al operador de cruce y al proceso evolutivo.

3.3.2. Un Segundo Modelo: Umbrales Adaptativos

En este apartado se propone una nueva propuesta basada en el uso de umbrales que se autoadaptan en el mismo proceso de aprendizaje de los parámetros de la defuzzificación, en lugar de usar umbrales con valores fijos

predefinidos, como en el modelo anterior. De esta forma, el proceso de aprendizaje multiobjetivo obtiene un conjunto de SBRDs con diferente equilibrio entre precisión e interpretabilidad utilizando no sólo los parámetros de la defuzzificación adaptativa sino también diferentes umbrales.

3.3.2.1. Mecanismo con Umbrales Adaptativos

A fin de reducir el espacio de búsqueda del AE, en vez de aprender el valor de los dos umbrales U_B y U_A, proponemos adaptar el rango disponible de los parámetros de la defuzzificación adaptativa utilizando un solo parámetro λ_U para el sistema en su conjunto, cuyo valor indica la anchura centrada en la mitad del intervalo, de forma que los extremos de ésta indica los valores de los dos umbrales, como se muestra en la Figura 3.2 siguiente.

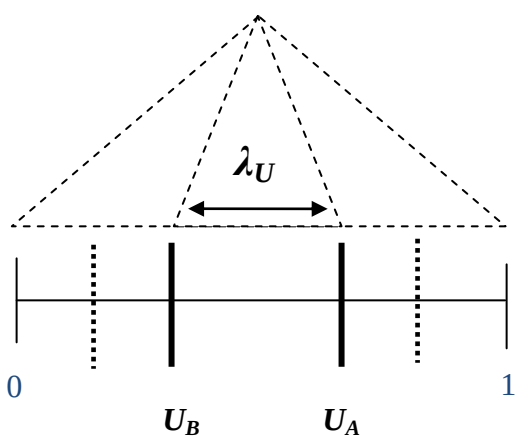


Figura 3.2: Descripción del parámetro para adaptar los umbrales

De esta forma, el valor de los umbrales vendrá determinado por la siguiente fórmula:

$$U_B = 0.50 - \lambda_U/2$$
$$U_A = 0.50 + \lambda_U/2 \qquad (3.6)$$

El uso de un único parámetro para llevar a cabo la adaptación de ambos umbrales U_B y U_A conjuntamente que proponemos es menos flexible en comparación con el uso de dos parámetros que representan los umbrales, ya

que fuerza a que ambos umbrales tengan el mismo rango de anchura y sean simétricos, pero reduce considerablemente el espacio de búsqueda del modelo.

3.3.2.2. Esquema de Codificación y Población Inicial

El nuevo AGMO tiene las mismas características que la primera propuesta descrita. La única diferencia estriba en que para representar los parámetros del modelo evolutivo propuesto este modelo utiliza un doble esquema de codificación real en el que el cromosoma C consiste en un vector de genes que contiene los parámetros de la defuzzificación adaptativa C_D (al igual que el primero modelo) y de los umbrales adaptativos C_U (ésta es la novedad). De esta forma, el cromosoma resultante es:

$$C = C_D + C_U \tag{3.7}$$

donde C_D codifica los parámetros del operador de defuzzificación adaptativo y C_U el aprendizaje de los umbrales adaptativos.

La Figura 3.3 ilustra el esquema de codificación propuesto en este trabajo, siendo N el número total de reglas de la BC. Los N valores β_i representan la parametrización del operador de defuzzificación, mientras que en el segmento del cromosoma dedicado a los umbrales, el valor λ_U representa el valor de los umbrales U_B y U_A.

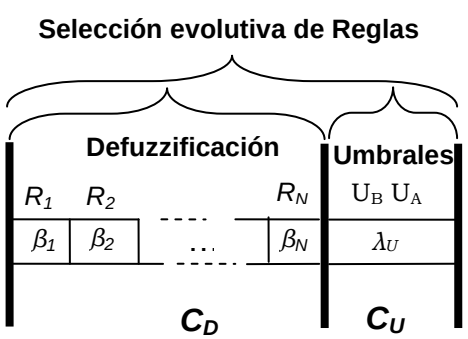


Figura 3.3: Esquema de Codificación propuesto para codificar los parámetros del operador de defuzzificación y los umbrales adaptativos

La primera parte del cromosoma, que pertenece a los parámetros de la defuzzificación adaptativa (C_D) está compuesto por N parámetros β_i, uno por cada regla R_i de la BR inicial. Cada uno toma valores en el intervalo [0, 1]:

$$C_D = (\beta_1, \ldots, \beta_N) \mid \beta_i \in [0, 1] \quad (3.8)$$

La segunda parte, usada para el aprendizaje de los umbrales adaptativos (C_U), se compone de un único gen que toma valores en el intervalo [0.30, 1]:

$$C_U = \lambda_U \mid \lambda_U \in [0.30, 1] \quad (3.9)$$

Al ser $U_B = 0.50 - \lambda_U/2$ y $U_A = 0.50 + \lambda_U/2$, este intervalo para λ_U permite que los umbrales estén comprendidos entre:

$U_B = 0.00$ y $U_A = 1{,}00$, cuando $\lambda_U = 1$

$U_B = 0.35$ y $U_A = 0.65$, cuando $\lambda_U = 0.30$

Estos valores se han seleccionado empíricamente tras la realización de diferentes pruebas.

Para obtener la población inicial, todos los individuos excepto el primero son inicializados de forma aleatoria asignando:

- un valor entre 0 y 1 a los N genes de la parte C_D, que contiene los parámetros de la defuzzificación adaptativa, y
- un valor entre 0.30 y 1 al gen λ_U de la parte C_U que determina el valor de los umbrales adaptativos U_B y U_A.

El primer individuo de la población es inicializado:

- con todos los valores a 1 en la parte C_D, a fin de comenzar el proceso evolutivo con todas las reglas activadas y sin peso asociado, y
- con $\lambda_U = 1$ en la parte correspondiente a los umbrales adaptativos C_U, a fin de tener como valores de umbrales iniciales $U_B = 0$ y $U_A = 1$.

3.4. Estudio Experimental

Para analizar el comportamiento práctico de la metodología propuesta, se ha llevado a cabo un estudio experimental, dividido en dos partes. La primera centrada en el primer modelo y la segunda en el modelo de umbrales adaptativos. Los modelos propuestos se han comparado con varios métodos guiados por un único objetivo. Para ello se han utilizado trece problemas reales de regresión de complejidades diferentes (diferente número de variables y datos). Dichos problemas están disponibles en la página web del proyecto KEEL [AFSG+09] (http://sci2s.ugr.es/keel/datasets.php) de donde pueden ser descargados.

La Tabla 3.1 resume las principales características de los trece conjuntos de datos. Para cada problema se muestra el número de variables y el número de ejemplos. Los problemas están ordenados de menor a mayor dimensionalidad, de forma que el problema PLA es el de más baja dimensionalidad y MOR y TRE los dimensionalidad más alta.

Tabla 3.1: Conjunto de datos considerados para el estudio experimental

Conjunto de Datos	**Nombre**	**Variables**	**Ejemplos**
Plastic Strength	PLA	3	1650
Quake	QUA	4	2178
Electrical Maintenance	ELE	5	1056
AutoMPG6	AU6	6	392
AutoMPG8	AU8	8	392
Anacalt	ANA	8	4052
Abalone	ABA	9	4177
Concrete	CON	9	1030
Stock Prices	STP	10	950
Ankara Weather	WAN	10	1609
Izmir Weather	WIZ	10	1461
Mortgage	MOR	16	1409
Treasure	TRE	16	1409

Antes de mostrar y analizar los resultados obtenidos en el estudio experimental vamos a describir como han sido configurados los conjuntos de datos y la metodología de comparación utilizada en dicho estudio.

3.4.1. Configuración Común del Estudio Experimental

Para obtener el conjunto inicial de reglas lingüísticas candidatas (BR inicial) sobre las que se aplica los modelos propuestos, se han empleado dos algoritmos: el conocido algoritmo ad-hoc basado en datos de WM [WM92] y la primera fase de MOGUL-IRL [CdJHL99, CH97], que denominamos FS-MOGUL. No se han utilizado otros métodos, debido a los problemas que presentan cuando se utiliza conjuntos de datos con un gran número de variables como los utilizados en este trabajo (véase la parte inferior de la Tabla 3.1).

Las Tablas 3.2 y 3.3 muestran los resultados medios de los SBRDs de referencia iniciales obtenidos con WM y FS-MOGUL, donde ECM_{ENT} y ECM_{TST} representan el ECM obtenido en entrenamiento y test respectivamente y $\#R$ representa el número medio de reglas. En este caso, se ha optado por ordenar los problemas de forma ascendente por el número de variables que posee (dimensionalidad), al igual que se hizo en la Tabla 3.1. Estos métodos obtienen la BR inicial sobre la que trabajan nuestros modelos propuestos.

Tabla 3.2: Resultado inicial usando la BR generada por WM

Nombre	**#R**	**ECM_{ENT}**	**ECM_{TST}**
PLA	14.80	3.434	3.557
QUA	53.60	0.0258	0.0267
ELE	65.00	56135	56359
AU6	116.00	4.338	6.819
AU8	70.60	12.71	13.73
ANA	72.40	0.187	0.189
ABA	68.20	8.407	8.424
CON	135.40	91.176	94.190
STP	122.80	9.074	9.042
WAN	156.00	16.063	16.403
WIZ	104.80	6.945	7.139
MOR	77.60	0.985	0.973
TRE	75.00	1.636	1.632

Tabla 3.3: Resultado inicial usando la BR generada por FS-MOGUL

Nombre	**#R**	**ECM_{ENT}**	**ECM_{TST}**
PLA	75.40	5.246	5.262
QUA	227.60	0.0633	0.0648
ELE	88.80	129400	133564
AU6	170.60	8.565	13.154
AU8	78.80	21.47	22.07
ANA	211.20	0.195	0.200
ABA	50.20	24.790	24.720
CON	124.40	163.157	164.638
STP	45.60	16.745	16.906
WAN	33.40	56.898	58.880
WIZ	52.40	38.414	41.442
MOR	31.40	2.002	1.995
TRE	33.00	2.667	2.685

Como operador de conjunción e implicación se ha utilizado la t-norma del mínimo.

Para todos nuestros experimentos, hemos seguido la misma metodología utilizada en el capítulo anterior. Esto es:

- las particiones lingüísticas están compuestas por cinco términos lingüísticos con forma triangular en el caso de los conjuntos de datos con menos de nueve variables y tres términos lingüísticos en las restantes,

- se ha considerado un *modelo de validación cruzada de 5 particiones*, con 6 semillas diferentes (véase Sección 2.5.1 del Capítulo 2),

- Para comparar los diferentes modelos multiobjetivos se ha considerado sólo los tres puntos del frente de Pareto más representativos: el punto más preciso (MAX_{PRE}), el punto mediano ($MEDIO_{INT\ /\ PRE}$) y el punto más interpretable (MAX_{INT}), como en [ADH+09, GAH10]. Estos tres puntos (solución más precisa, solución media y solución más interpretable) son lugares representativos en el plano ECM - $R_P\_MR_{AC}$, y se han tenido en cuenta para llevar a cabo un análisis estadístico para comparar entre si las diferentes propuestas multiobjetivo.

Para los modelos basados en un solo objetivo, para cada conjunto de datos se ha calculado los valores medios de las 30 pruebas del ECM en el conjunto de entrenamiento y test (ECM_{ENT} y ECM_{TST}), así como los valores medios del número de reglas finales $\#R_F$ y del índice de interpretabilidad global $R_P\_MR_{AC}$,

aunque no forman parte del objetivo, a fin de poder comparar sus valores con los obtenidos en las propuestas multiobjetivo. En el caso de los modelos con enfoque multiobjetivo, se han calculado los mismos valores medios (aunque $\#R_F$ no es un objetivo del AGMO).

Para comparar los modelos multiobjetivo entre sí, ha sido necesario utilizar una medida adicional para calcular el rendimiento total del AGMO [CVL02, ZDT00]. De entre las diferentes medidas propuestas en la literatura científica, hemos seleccionado el ratio CS [ZDT00] (ratio de cobertura de dos conjuntos) como herramienta para comparar los frentes de Pareto obtenidos por las diferentes propuestas multiobjetivo. El ratio CS, que permite comparar dos frentes de Pareto, se define como el porcentaje de puntos de un frente de Pareto que son iguales o dominados por los puntos del otro frente de Pareto con el que se compara:

$$CS(X', X'') = \frac{\left|\{a'' \in X''; \exists a' \in X' : a' \geq a''\}\right|}{\left|X''\right|}, \tag{3.10}$$

donde $|\cdot|$ representa la cardinalidad del conjunto y el símbolo ≥ corresponde a la relación de dominancia o igualdad del Pareto, siendo X', X'' los conjuntos de puntos de los dos frentes de Pareto a comparar y a', a'' dos puntos pertenecientes a los conjuntos de X' y X'', respectivamente.

Observando la ecuación vemos que CS se define en el intervalo [0, 1]. Si todos los puntos de X'' son iguales o dominados por puntos de X', entonces por definición CS = 1. En caso contrario, es decir, si no existe en X' ningún punto que domine o sea igual a algún punto de X'', entonces CS = 0.

En general hay que considerar tanto $CS(X', X'')$ como $CS(X'', X')$, debido a que el conjunto intersección de ambos no es vacío. La ventaja de esta medida es que es fácil de calcular y proporciona una comparación relativa entre pares de AGMOs, al permitir comparar los frentes de Paretos de ambos.

Para evaluar y determinar si existen diferencias significativas entre los resultados obtenidos por los distintos métodos, se ha optado por realizar un análisis estadístico [Dem06, GFLH09, GFLH10, GH08], mediante el uso de test no paramétricos, de acuerdo a las recomendaciones formuladas en [GFLH09]. En concreto, en este estudio se van a aplicar los siguientes test no paramétricos:

- El test de Friedman [Fri37], para realizar comparaciones múltiples y detectar si existen diferencias significativas entre los resultados generados por los métodos a comparar, y

- El test de Finner [Fin93], procedimiento a posteriori (post-hoc), que utilizaremos en caso que el test anterior detecte diferencias entre las muestras, para comparar el mejor método (algoritmo de control) contra el resto y detectar si la diferencia entre los métodos son o no significativas, de acuerdo a un determinado nivel de confianza.

Una descripción más amplia de estos tests estadísticos no paramétricos se encuentra en el Apéndice C.

Para realizar los test, se ha utilizado un nivel de confianza $\alpha = 0.05$. Además para que las diferencias existentes entre los distintos métodos se puedan comparar, adoptamos, tal y como se indicó en la Sección 2.5.1 del capítulo anterior, la diferencia normalizada DIFF (véase Expresión 2.8 del Capítulo 2) para realizar los test.

3.4.2. Primera Parte del Estudio Experimental

En la primera parte del estudio experimental se analiza el comportamiento práctico del primer modelo, estudiando en primer lugar la influencia de los diferentes valores de los umbrales en el mecanismo de mejora de la interpretabilidad con umbrales fijos propuesto (Subsección 3.4.2.1), y posteriormente, comparando las propuestas de un solo objetivo con la del multiobjetivo (Subsección 3.4.2.2.).

Los modelos considerados para esta primera parte del estudio experimental se resumen en la Tabla 3.4, donde, como podemos observar, la propuesta multiobjetivo con la mejora de la interpretabilidad, que denotaremos como MO-AD_I, se compara con dos modelos de un solo objetivo: SO-AD, que es la defuzzificación adaptativa estándar [CHMP04] orientada a precisión, y SO-AD_I que es la misma, pero añadiendo el mecanismo propuesto para mejorar la interpretabilidad.

Tabla 3.4: Modelos considerados para la comparación

Modelo	Descripción
	SDEs
SO-AD	Defuzzificación adaptativa [CHMP04]
$SO\text{-}AD_I$	Defuzzificación adaptativa con el mecanismo para mejorar la interpretabilidad
	SDEMO
$MO\text{-}AD_I$	Defuzzificación adaptativa con el mecanismo para mejorar la interpretabilidad

En el caso de las propuestas de defuzzificación adaptativa estudiadas, los valores de los parámetros de entrada considerados por los modelos de un solo objetivo (SO-AD y $SO\text{-}AD_I$) son: tamaño de la población de 61, 200000 evaluaciones, probabilidad de cruce 0.6 y 0.2 como probabilidad de mutación por cromosoma. En el caso del AGMO ($MO\text{-}AD_I$) son: tamaño de la población de 200 individuos, tamaño de la población externa de 200 individuos, 200000 evaluaciones y 0.2 como probabilidad de mutación.

3.4.2.1. Análisis de la Influencia de los Diferentes Valores de los Umbrales

En este apartado se analiza el comportamiento y se muestra los resultados obtenidos por el algoritmo multiobjetivo propuesto ($MO\text{-}AD_I$) con diferentes valores umbrales, con el fin de estudiar la influencia de estos valores en los criterios de precisión e interpretabilidad.

Para realizar este análisis, hemos elegido tres configuraciones de umbrales fijos diferentes:

- $U_B = 0.1$ y $U_A = 0.9$
- $U_B = 0.2$ y $U_A = 0.8$
- $U_B = 0.3$ y $U_A = 0.7$

que denotaremos respectivamente como $MO\text{-}AD_{I(0.1\text{-}0.9)}$, $MO\text{-}AD_{I(0.2\text{-}0.8)}$ y $MO\text{-}AD_{I(0.3\text{-}0.7)}$, para la propuesta multiobjetivo.

Las Tablas 3.5 y 3.6 muestran los resultados obtenidos en los tres puntos representativos del Pareto por las diferentes configuraciones de umbrales, utilizando como BRs iniciales las obtenidas por WM y FS-MOGUL, respectivamente.

Tabla 3.5: Resultados obtenidos por MO-AD$_I$ para diferentes configuraciones de umbrales usando la BR inicial de WM

		MAX $_{INT}$					MEDIA ($_{INT\ /\ PRE}$)					MAX $_{PRE}$				
Conjunto de Datos	MO-AD$_I$ U_B - U_A	ECM$_{ENT}$	ECM$_{TST}$	#R$_F$	R$_P$_MR$_{AC}$	(#R$_P$ MR$_{AC}$)	ECM$_{ENT}$	ECM$_{TST}$	#R$_F$	R$_P$_MR$_{AC}$	(#R$_P$ MR$_{AC}$)	ECM$_{ENT}$	ECM$_{TST}$	#R$_F$	R$_P$_MR$_{AC}$	(#R$_P$ MR$_{AC}$)
PLA	0.1 - 0.9	6.947800	6.940028	6.1	0.128	(0.00 1.02)	2.073691	2.091053	10.6	0.385	(2.23 2.48)	1.699226	**1.759935**	14.0	0.771	(10.80 3.25)
	0.2 - 0.8	7.042636	7.042633	6.0	0.125	(0.00 1.00)	2.237865	2.245128	10.6	0.346	(1.37 2.40)	1.769444	1.875019	13.9	0.675	(8.50 3.11)
	0.3 - 0.7	7.042636	7.042633	6.0	0.125	(0.00 1.00)	2.543587	2.545364	10.4	0.309	(0.60 2.31)	1.830941	1.900387	12.4	**0.551**	(6.13 2.75)
QUA	0.1 - 0.9	0.023570	0.024276	22.7	0.179	(4.93 2.12)	0.020914	0.022153	27.2	0.287	(10.60 3.01)	0.020675	**0.021944**	32.6	0.448	(22.67 3.79)
	0.2 - 0.8	0.024555	0.025273	19.9	0.138	(1.23 2.02)	0.021178	0.022430	25.6	0.237	(6.67 2.80)	0.020782	0.022043	32.1	0.416	(20.73 3.57)
	0.3 - 0.7	0.026361	0.027337	17.7	0.114	(0.13 1.80)	0.021700	0.022975	24.7	0.193	(3.13 2.62)	0.020897	0.022139	31.4	**0.380**	(17.37 3.49)
ELE	0.1 - 0.9	57867	60608	35.8	0.167	(7.37 3.10)	35428	39094	38.6	0.285	(16.30 4.48)	32518	**35633**	45.2	0.469	(33.27 5.96)
	0.2 - 0.8	83303	84897	27.2	0.082	(1.10 2.06)	44049	47997	34.0	0.159	(6.30 3.11)	32947	35747	45.3	0.440	(29.23 6.02)
	0.3 - 0.7	143152	150408	20.0	0.063	(0.03 1.77)	64459	70994	27.6	0.086	(1.17 2.16)	33433	35677	45.4	**0.421**	(25.97 6.19)
AU6	0.1 - 0.9	3.487613	6.568578	84.6	0.214	(6.60 6.07)	2.746415	6.482094	73.4	0.293	(28.57 5.57)	2.568845	6.457802	73.6	0.387	(48.77 5.77)
	0.2 - 0.8	3.715838	7.137473	61.2	0.143	(2.77 4.29)	2.787005	6.521554	64.7	0.216	(15.17 4.94)	2.623889	6.440329	72.0	0.348	(41.27 5.57)
	0.3 - 0.7	5.551319	9.068401	40.9	0.086	(0.27 2.78)	3.326684	7.252765	52.2	0.132	(3.43 3.84)	2.687347	**6.430410**	72.5	**0.325**	(35.53 5.61)
AU8	0.1 - 0.9	11.281517	13.152617	32.1	0.132	(3.00 3.93)	9.102294	11.28545	33.9	0.201	(10.80 4.45)	8.897037	**10.759520**	36.2	0.303	(23.00 4.99)
	0.2 - 0.8	16.321254	18.062810	19.0	0.070	(0.37 2.39)	9.601045	11.69286	26.3	0.124	(3.80 3.47)	8.991249	10.937447	35.1	0.276	(19.97 4.80)
	0.3 - 0.7	21.088986	23.062541	14.8	0.054	(0.00 1.93)	10.672853	12.84438	21.4	0.090	(1.13 2.91)	9.127924	11.186454	35.9	**0.255**	(17.23 4.73)
ANA	0.1 - 0.9	0.647718	0.646577	63.7	0.190	(1.77 1.43)	0.107990	0.109300	67.1	0.267	(3.67 1.94)	0.007488	**0.009588**	67.5	0.659	(59.50 1.98)
	0.2 - 0.8	0.871531	0.871570	54.9	0.151	(0.27 1.19)	0.337729	0.336884	60.4	0.216	(1.27 1.65)	0.007523	0.009669	65.6	0.585	(48.97 1.97)
	0.3 - 0.7	0.955815	0.960777	45.6	0.137	(0.03 1.09)	0.438822	0.440663	50.6	0.195	(0.70 1.52)	0.007551	0.009691	64.8	**0.531**	(41.37 1.96)
ABA	0.1 - 0.9	7.613683	7.712487	15.2	0.083	(2.00 2.58)	4.836747	4.858605	19.8	0.142	(6.23 3.66)	4.753375	**4.789237**	26.5	0.225	(13.47 4.79)
	0.2 - 0.8	12.693606	12.611176	11.2	0.050	(0.20 1.85)	5.533293	5.538769	15.3	0.090	(2.17 2.81)	4.790064	4.817361	25.9	0.213	(12.17 4.73)
	0.3 - 0.7	14.830294	14.893969	10.3	0.044	(0.00 1.66)	6.432483	6.432756	13.4	0.066	(0.37 2.42)	4.821530	4.845469	23.1	**0.184**	(9.37 4.39)
CON	0.1 - 0.9	70.753300	76.129910	83.8	0.131	(6.30 10.97)	55.1029	61.1530	71.5	0.169	(20.23 9.65)	50.3219	**57.5966**	59.9	0.219	(37.07 8.39)
	0.2 - 0.8	98.572969	104.99026	41.1	0.056	(1.83 5.03)	55.8998	64.5417	42.2	0.093	(9.70 5.87)	51.4730	58.4786	56.0	**0.190**	(30.87 7.70)
	0.3 - 0.7	128.97494	134.53928	25.9	0.030	(0.03 3.06)	73.2893	83.2969	33.2	0.049	(1.60 4.37)	52.9665	59.9898	60.0	0.198	(31.23 8.43)
STP	0.1 - 0.9	3.708981	3.794866	25.7	0.061	(3.20 5.04)	2.129880	2.227016	29.8	0.092	(7.83 6.25)	2.052056	**2.154450**	36.7	0.139	(15.43 7.94)
	0.2 - 0.8	4.652642	4.674927	17.0	0.037	(0.60 3.57)	2.321608	2.455662	23.8	0.064	(3.83 5.06)	2.115639	2.209151	34.1	0.116	(11.53 7.15)
	0.3 - 0.7	8.334997	8.401997	10.5	0.024	(0.07 2.44)	3.113456	3.240279	15.9	0.039	(1.13 3.58)	2.191832	2.272137	33.8	**0.106**	(9.67 6.91)
WAN	0.1 - 0.9	6.463753	7.172746	38.8	0.115	(12.30 13.24)	6.033333	6.780082	43.1	0.146	(18.50 15.22)	5.983361	**6.674879**	48.6	0.190	(27.90 17.52)
	0.2 - 0.8	9.346113	10.098808	31.0	0.060	(1.20 9.81)	6.214014	6.936514	38.5	0.092	(5.40 13.06)	6.040380	6.706163	46.7	0.159	(19.83 16.73)
	0.3 - 0.7	16.437768	17.227865	19.2	0.029	(0.03 5.03)	7.376043	8.159211	27.6	0.052	(0.67 8.66)	6.130478	6.859088	46.0	**0.140**	(14.50 16.28)
WIZ	0.1 - 0.9	3.846971	4.359892	30.5	0.147	(7.90 10.00)	2.718277	3.149809	36.1	0.211	(14.73 12.86)	2.629827	**3.068601**	43.2	0.297	(27.57 15.13)
	0.2 - 0.8	7.513500	8.062620	24.5	0.079	(0.87 6.87)	3.009577	3.614640	32.7	0.146	(5.47 11.02)	2.675106	3.230414	43.8	0.279	(23.43 15.32)
	0.3 - 0.7	24.760387	25.488658	14.8	0.034	(0.03 3.11)	5.940635	6.631707	19.3	0.062	(0.73 5.35)	2.734843	3.302554	45.1	**0.265**	(20.30 15.39)
MOR	0.1 - 0.9	1.590566	1.564395	9.2	0.042	(0.77 2.97)	0.201851	0.219410	12.9	0.079	(2.83 4.83)	0.153782	**0.165901**	16.5	0.144	(9.57 6.50)
	0.2 - 0.8	3.173316	3.214704	6.5	0.024	(0.13 1.81)	0.321694	0.342161	10.6	0.053	(1.37 3.54)	0.159248	0.171457	17.5	0.142	(8.90 6.69)
	0.3 - 0.7	3.180415	3.323322	5.6	0.021	(0.00 1.64)	0.551852	0.574422	8.5	0.039	(0.43 2.86)	0.163368	0.175955	18.2	**0.137**	(7.90 6.86)
TRE	0.1 - 0.9	1.720608	1.786207	10.3	0.045	(0.67 2.96)	0.248659	0.280729	14.2	0.086	(3.10 4.75)	0.202813	**0.232601**	20.3	0.177	(12.23 7.00)
	0.2 - 0.8	3.553366	3.630207	6.5	0.028	(0.00 2.07)	0.557518	0.555809	10.7	0.056	(1.00 3.61)	0.208263	0.237011	19.0	0.154	(9.67 6.58)
	0.3 - 0.7	4.611414	4.668308	5.5	0.025	(0.00 1.80)	1.005509	1.023034	7.9	0.039	(0.23 2.72)	0.212469	0.238197	18.9	**0.144**	(7.93 6.68)

Tabla 3.6: Resultados obtenidos por MO-AD$_I$ para diferentes configuraciones de umbrales usando la BR inicial de FS-MOGUL

		MAX $_{INT}$					MEDIA ($_{INT / PRE}$)					MAX $_{PRE}$				
Conjunto de Datos	$\frac{MO\text{-}AD_I}{U_B - U_A}$	ECM_{ENT}	ECM_{TST}	$\#R_F$	$R_P\_MR_{AC}$	$(\#R_P \; MR_{AC})$	ECM_{ENT}	ECM_{TST}	$\#R_F$	$R_P\_MR_{AC}$	$(\#R_P \; MR_{AC})$	ECM_{ENT}	ECM_{TST}	$\#R_F$	$R_P\_MR_{AC}$	$(\#R_P \; MR_{AC})$
PLA	0.1 - 0.9	1.574572	1.623150	21.9	0.130	(4.80 4.68)	1.171741	1.207446	29.3	0.216	(11.57 6.61)	1.163733	**1.201293**	40.5	0.379	(28.03 9.16)
	0.2 - 0.8	3.278771	3.305375	11.5	0.051	(0.23 2.37)	1.246104	1.273259	20.3	0.111	(2.83 4.40)	1.164596	1.201882	36.9	0.298	(19.17 8.12)
	0.3 - 0.7	5.290166	5.346323	7.0	0.029	(0.00 1.38)	1.754391	1.810790	13.2	0.063	(0.10 2.97)	1.166274	1.204826	33.5	**0.251**	(14.87 7.24)
QUA	0.1 - 0.9	0.021684	0.022334	199.7	0.292	(13.63 26.19)	0.017291	0.018149	177.8	0.385	(71.50 22.83)	0.017111	**0.018167**	170.8	0.553	(141.50 24.26)
	0.2 - 0.8	0.017468	0.018472	132.1	0.208	(20.67 16.32)	0.017125	0.018199	123.1	0.260	(43.27 16.55)	0.017071	0.018207	124.1	0.320	(65.73 17.58)
	0.3 - 0.7	0.017608	0.018610	86.4	0.129	(13.13 10.08)	0.017135	0.018257	98.0	0.174	(22.40 12.52)	0.017088	0.018183	117.2	**0.268**	(48.00 16.29)
ELE	0.1 - 0.9	64241	67141	56.0	0.129	(4.23 4.10)	42156	45526	49.5	0.228	(17.10 5.09)	39495	**42072**	52.0	0.352	(33.80 6.28)
	0.2 - 0.8	84698	87379	35.9	0.068	(1.40 2.35)	46395	50173	40.8	0.126	(6.07 3.57)	39976	42689	49.6	0.297	(26.70 5.68)
	0.3 - 0.7	108723	113721	23.2	0.048	(0.07 1.85)	61099	65626	29.1	0.078	(2.07 2.56)	40766	43063	48.5	**0.258**	(22.10 5.18)
AU6	0.1 - 0.9	3.821884	6.764193	143.4	0.197	(4.50 10.80)	3.296278	6.516126	129.9	0.266	(32.60 9.97)	2.941543	**6.263927**	111.0	0.389	(79.53 9.18)
	0.2 - 0.8	3.454157	7.474670	79.4	0.133	(10.20 6.07)	3.009759	6.955385	81.0	0.173	(21.03 6.51)	2.925950	6.544321	86.8	0.235	(38.40 7.15)
	0.3 - 0.7	4.202431	7.836509	58.7	0.083	(3.43 4.31)	3.124690	7.149861	69.9	0.126	(10.77 5.52)	2.976153	6.502033	88.3	**0.226**	(34.93 7.27)
AU8	0.1 - 0.9	10.610777	11.814385	24.6	0.120	(3.97 3.30)	7.526326	9.129575	27.7	0.185	(11.33 3.95)	7.386325	**9.000788**	33.3	0.283	(22.67 4.81)
	0.2 - 0.8	16.296509	18.007791	18.0	0.069	(0.63 2.28)	7.898196	9.633468	23.8	0.122	(4.43 3.28)	7.435334	9.047838	31.6	0.250	(18.63 4.57)
	0.3 - 0.7	19.910480	20.495670	14.7	0.056	(0.00 2.00)	9.872984	11.83105	19.3	0.081	(0.37 2.76)	7.505406	9.082736	32.1	**0.238**	(17.17 4.52)
ANA	0.1 - 0.9	0.044437	0.047909	174.4	0.188	(12.07 4.67)	0.013463	0.015529	154.4	0.250	(44.00 4.24)	0.007158	0.009335	133.0	0.358	(91.53 4.12)
	0.2 - 0.8	0.038210	0.040613	129.3	0.141	(12.93 3.22)	0.009456	0.011776	114.0	0.191	(27.63 3.68)	0.007033	**0.009237**	112.5	0.252	(50.90 3.84)
	0.3 - 0.7	0.126768	0.126480	85.7	0.094	(8.33 2.16)	0.015763	0.018568	92.5	0.125	(10.17 2.93)	0.007180	0.009461	108.3	**0.216**	(34.20 3.92)
ABA	0.1 - 0.9	10.034018	9.980401	15.6	0.098	(1.10 2.33)	6.252071	6.261997	20.2	0.180	(5.57 3.34)	6.198179	**6.219666**	24.9	0.311	(15.07 4.32)
	0.2 - 0.8	13.518071	13.474387	10.1	0.064	(0.03 1.70)	6.450398	6.484010	15.2	0.115	(1.40 2.69)	6.200991	6.221171	24.6	0.258	(10.83 4.02)
	0.3 - 0.7	13.471095	13.490648	9.1	0.058	(0.00 1.56)	6.933849	6.956984	13.8	0.099	(0.70 2.45)	6.204145	6.222632	22.9	**0.220**	(7.90 3.77)
CON	0.1 - 0.9	67.672232	71.789422	86.8	0.143	(5.13 12.02)	59.010937	65.03955	66.0	0.201	(25.23 9.74)	56.380503	**62.489223**	62.6	0.257	(39.37 9.66)
	0.2 - 0.8	80.296355	88.380934	40.9	0.071	(3.53 5.52)	58.333450	65.72600	45.4	0.113	(10.77 6.87)	56.679133	62.796797	55.0	0.195	(27.00 8.43)
	0.3 - 0.7	116.57328	123.10188	27.7	0.036	(0.30 3.45)	66.961798	73.82797	36.8	0.066	(2.67 5.44)	57.450287	63.305839	58.3	**0.189**	(24.53 8.79)
STP	0.1 - 0.9	5.405339	5.474713	13.3	0.096	(1.33 3.21)	4.259753	4.362481	18.6	0.167	(4.63 4.54)	4.159594	**4.255350**	25.0	0.287	(11.73 6.19)
	0.2 - 0.8	7.052710	7.174048	9.1	0.063	(0.13 2.41)	4.338720	4.473762	16.2	0.136	(3.17 3.98)	4.179792	4.288358	25.0	0.268	(9.90 6.25)
	0.3 - 0.7	8.748136	8.816542	7.9	0.058	(0.00 2.25)	4.455629	4.560821	14.7	0.125	(2.67 3.75)	4.194782	4.309624	24.9	**0.252**	(8.50 6.22)
WAN	0.1 - 0.9	26.310868	26.725976	9.6	0.088	(0.53 2.86)	14.522565	14.77936	13.3	0.193	(4.40 4.46)	13.774738	**14.009191**	16.1	0.361	(12.27 6.19)
	0.2 - 0.8	29.878818	30.684648	7.9	0.069	(0.00 2.43)	15.051468	15.54537	12.1	0.150	(2.43 4.00)	13.832013	14.079000	15.6	0.331	(10.67 5.96)
	0.3 - 0.7	32.988571	34.115293	7.9	0.065	(0.00 2.26)	15.540758	16.02159	11.5	0.122	(0.93 3.81)	13.948231	14.207306	15.2	**0.302**	(9.47 5.61)
WIZ	0.1 - 0.9	12.545764	13.543774	15.7	0.107	(2.57 4.18)	3.881428	4.467590	19.7	0.193	(7.50 6.11)	3.413844	**3.856717**	25.0	0.351	(19.80 8.12)
	0.2 - 0.8	34.049969	35.694754	11.6	0.059	(0.13 2.93)	4.965794	5.631655	16.9	0.121	(2.30 5.01)	3.463373	3.912941	24.4	0.314	(16.77 7.72)
	0.3 - 0.7	34.337238	36.299840	10.5	0.054	(0.00 2.74)	6.706427	7.391861	14.5	0.091	(0.87 4.18)	3.547122	4.014734	24.5	**0.291**	(14.73 7.61)
MOR	0.1 - 0.9	3.056375	3.052106	6.0	0.060	(0.03 1.89)	0.423308	0.439410	8.6	0.134	(2.57 2.93)	0.267971	**0.275199**	12.8	0.305	(9.03 5.06)
	0.2 - 0.8	3.325074	3.280224	5.6	0.055	(0.00 1.74)	0.502724	0.495026	7.8	0.118	(1.90 2.76)	0.275008	0.281564	13.0	0.302	(8.77 5.12)
	0.3 - 0.7	3.689980	3.747015	5.8	0.050	(0.00 1.58)	0.548342	0.553402	7.8	0.109	(1.50 2.69)	0.283597	0.290467	13.0	**0.293**	(8.13 5.15)
TRE	0.1 - 0.9	4.887197	5.066371	5.8	0.054	(0.17 1.80)	0.806103	0.792245	8.7	0.104	(0.87 3.17)	0.448043	0.436540	10.4	0.229	(7.20 4.17)
	0.2 - 0.8	5.307431	5.513163	5.4	0.048	(0.00 1.67)	0.931746	0.905101	8.4	0.092	(0.43 2.96)	0.450184	**0.436108**	10.3	0.218	(6.60 4.10)
	0.3 - 0.7	4.886871	4.991007	5.7	0.046	(0.00 1.62)	0.991394	0.975439	8.0	0.084	(0.23 2.81)	0.455389	0.437021	10.3	**0.211**	(6.17 4.10)

Todas las configuraciones han sido comparadas con el fin de determinar cuál de ellas es la óptima. Dado que hay que comparar de forma conjunta más de dos algoritmos, se ha utilizado el test de Friedman para realizar comparaciones múltiples, a fin de comprobar si alguno de los resultados presenta alguna desigualdad. En caso de encontrar alguna, averiguamos a posteriori mediante el test post-hoc de Finner cuál de los resultados promedio de los algoritmos asociados son estadísticamente diferentes.

Las Tablas 3.7 a 3.12 muestra las clasificaciones (rankings) de los diferentes modelos considerados en el estudio para las BR de WM y FS-MOGUL, obtenidas al aplicar el test de Friedman con un nivel de confianza $\alpha = 0.05$ (la probabilidad máxima que se está dispuesto a asumir de rechazar la hipótesis nula cuando es cierta, es de un 5%). Por tanto, el test de Friedman indican que existen diferencias significativas entre los resultados cuando el estadístico de contraste calculado por el test *p-fried* < 0.05.

Aunque no es uno de los objetivos, en el test también se ha tenido en cuenta la métrica número de reglas finales $\#R_F$, a fin de confirmar la relación entre el índice $R_P\_MR_{AC}$ y $\#R_F$.

Tabla 3.7: Ranking obtenido a través del test de Friedman para diferentes valores de umbrales en las métricas ECM_{TST}, $R_P\_MR_{AC}$ y $\#R_F$ en el punto de máxima precisión MAX_{PRE} usando la BR inicial de WM

Algoritmo	Ranking en ECM_{TST} (*p-fried*: 0.000196)	Ranking en $R_P\_MR_{AC}$ (*p-fried*:0.000006)	Ranking en $\#R_F$ (*p-fried*: 0.125315)
$MO\text{-}AD_{I(0.1\text{-}0.9)}$	1.1538	3	2.4615
$MO\text{-}AD_{I(0.2\text{-}0.8)}$	2.0769	1.9231	1.7692
$MO\text{-}AD_{I(0.3\text{-}0.7)}$	2.7692	1.0769	1.7692

Tabla 3.8: Ranking obtenido a través del test de Friedman para diferentes valores de umbrales en las métricas ECM_{TST}, $R_P\_MR_{AC}$ y $\#R_F$ en el punto de máxima precisión MAX_{PRE} usando la BR inicial de FS-MOGUL

Algoritmo	Ranking en ECM_{TST} (*p-fried*: 0.0000091)	Ranking en $R_P\_MR_{AC}$ (*p-fried*:0.000002)	Ranking en $\#R_F$ (*p-fried*: 0.001366)
$MO\text{-}AD_{I(0.1\text{-}0.9)}$	1.1538	3	2.8077
$MO\text{-}AD_{I(0.2\text{-}0.8)}$	2	2	1.7308
$MO\text{-}AD_{I(0.3\text{-}0.7)}$	2.8462	1	1.4615

Tabla 3.9: Ranking obtenido a través del test de Friedman para diferentes valores de umbrales en las métricas ECM_{TST}, $R_P\_MR_{AC}$ y $\#R_F$ en el punto de máxima interpretabilidad MAX $_{INT}$ usando la BR inicial de WM

Algoritmo	Ranking en ECM_{TST} (*p-fried*: 0.000004)	Ranking en $R_P\_MR_{AC}$ (*p-fried*:0.000004)	Ranking en $\#R_F$ (*p-fried*: 0.000004)
$MO\text{-}AD_{I(0.1\text{-}0.9)}$	1	3	3
$MO\text{-}AD_{I(0.2\text{-}0.8)}$	2.0385	1.9615	1.9615
$MO\text{-}AD_{I(0.3\text{-}0.7)}$	2.9615	1.0385	1.0385

Tabla 3.10: Ranking obtenido a través del test de Friedman para diferentes valores de umbrales en las métricas ECM_{TST}, $R_P\_MR_{AC}$ y $\#R_F$ en el punto de máxima interpretabilidad MAX $_{INT}$ usando la BR inicial de FS-MOGUL

Algoritmo	Ranking en ECM_{TST} (*p-fried*: 0.000912)	Ranking en $R_P\_MR_{AC}$ (*p-fried*:0.000002)	Ranking en $\#R_F$ (*p-fried*: 0.000012)
$MO\text{-}AD_{I(0.1\text{-}0.9)}$	1.3077	3	3
$MO\text{-}AD_{I(0.2\text{-}0.8)}$	1.9231	2	1.8462
$MO\text{-}AD_{I(0.3\text{-}0.7)}$	2.7692	1	1.1538

Tabla 3.11: Ranking obtenido a través del test de Friedman para diferentes valores de umbrales en las métricas ECM_{TST}, $R_P\_MR_{AC}$ y $\#R_F$ en el punto de mediana precisión MEDIA $_{(INT\ /\ PRE)}$ usando la BR inicial de WM

Algoritmo	Ranking en ECM_{TST} (*p-fried*: 0.000002)	Ranking en $R_P\_MR_{AC}$ (*p-fried*: 0.000002)	Ranking en $\#R_F$ (*p-fried*: 0.000002)
$MO\text{-}AD_{I(0.1\text{-}0.9)}$	1	3	3
$MO\text{-}AD_{I(0.2\text{-}0.8)}$	2	2	2
$MO\text{-}AD_{I(0.3\text{-}0.7)}$	3	1	1

Tabla 3.12: Ranking obtenido a través del test de Friedman para diferentes valores de umbrales en las métricas ECM_{TST}, $R_P\_MR_{AC}$ y $\#R_F$ en el punto de mediana precisión MEDIA $_{(INT\ /\ PRE)}$ usando la BR inicial de FS-MOGUL

Algoritmo	Ranking en ECM_{TST} (*p-fried*: 0.000006)	Ranking en $R_P\_MR_{AC}$ (*p-fried*: 0.000002)	Ranking en $\#R_F$ (*p-fried*: 0.000002)
$MO\text{-}AD_{I(0.1\text{-}0.9)}$	1.0769	3	3
$MO\text{-}AD_{I(0.2\text{-}0.8)}$	1.9231	2	2
$MO\text{-}AD_{I(0.3\text{-}0.7)}$	3	1	1

Observando las tablas, podemos ver que, en los tres puntos más representativos del Pareto: el más interpretable (MAX $_{INT}$), el punto mediano (MEDIA $_{(INT\ /\ PRE)}$) y el más preciso (MAX $_{PRE}$) existen diferencias entre los diferentes umbrales usados (los valores de los rankings en cada uno de los objetivos y en $\#R_F$ son distintos). No obstante, como denominador común, podemos observar que en esos tres puntos, el modelo MO-AD$_{I(0.1\text{-}0.9)}$ obtiene la mejor clasificación en precisión (ECM$_{TST}$) y MO-AD$_{I(0.3\text{-}0.7)}$ obtiene el mejor ranking en el índice $R_P\_MR_{AC}$ y la métrica $\#R_F$.

En casi todos los casos la hipótesis nula es rechazada (*p-fried* < 0.05), lo que indica que existen diferencias significativas entre los resultados observados en todos los conjuntos de datos. La única excepción se da en la métrica $\#R_F$ en el punto de máxima precisión MAX $_{PRE}$, cuando la BR utilizada es WM (Tabla 3.7), donde como podemos observar, el estadísticos de contraste *p-fried*=0.125315, supera el nivel de confianza $\alpha = 0.05$, lo que indica que se acepta la hipótesis nula y por tanto los resultados son similares.

Para determinar si las diferencias detectadas por el test de Friedman entre los distintos modelos son estadísticamente significativas se ha realizado una comparación por pares mediante el procedimiento a posteriori de Finner para comparar el método con mejor ranking (método de control) en cada caso con los restantes métodos. Las Tablas 3.13 a 3.18 presenta estos resultados, obtenidos al aplicar el test de Finner con un nivel de confianza $\alpha = 0.05$.

Tabla 3.13: Tabla de Finner con $\alpha = 0.05$ para los modelos en ECM$_{TST}$, $R_P\_MR_{AC}$ y $\#R_F$ en el punto de máxima precisión MAX $_{PRE}$ usando la BR de WM

	ECM$_{TST}$			$R_P\_MR_{AC}$			$\#R_F$		
i	Algoritmo	*p-finner*	Hipot	Algoritmo	*p-finner*	Hipot	Algoritmo	*p-finner*	Hipot
1	MO-AD I(0.3-0.7)	0.000076	Rech.	MO-AD I(0.1-0.9)	0.000002	Rech.	MO-AD I(0.1-0.9)	0.149097	Acep.
2	MO-AD I(0.2-0.8)	0.018603	Rech.	MO-AD I(0.2-0.8)	0.030984	Rech.	MO-AD I(0.2-0.8)	1	Acep.

Tabla 3.14: Tabla de Finner con $\alpha = 0.05$ para los modelos en ECM$_{TST}$, $R_P\_MR_{AC}$ y $\#R_F$ en el punto de máxima precisión MAX $_{PRE}$ usando la BR de FS-MOGUL

	ECM$_{TST}$			$R_P\_MR_{AC}$			$\#R_F$		
i	Algoritmo	*p-finner*	Hipot	Algoritmo	*p-finner*	Hipot	Algoritmo	*p-finner*	Hipot
1	MO-AD I(0.3-0.7)	0.000032	Rech.	MO-AD I(0.1-0.9)	0.000001	Rech.	MO-AD I(0.1-0.9)	0.001179	Rech.
2	MO-AD I(0.2-0.8)	0.030984	Rech.	MO-AD I(0.2-0.8)	0.010187	Rech.	MO-AD I(0.2-0.8)	0.492457	Acep.

Tabla 3.15: Tabla de Finner con $\alpha = 0.05$ para los modelos en ECM_{TST}, $R_P\_MR_{AC}$ y $\#R_F$ en el punto de máxima interpretabilidad MAX_{INT} usando la BR de WM

	ECM_{TST}			$R_P\_MR_{AC}$			$\#R_F$		
i	Algoritmo	*p-finner*	Hipot	Algoritmo	*p-finner*	Hipot	Algoritmo	*p-finner*	Hipot
1	MO-AD I(0.3-0.7)	0.000001	Rech.	MO-AD I(0.1-0.9)	0.000001	Rech.	MO-AD I(0.1-0.9)	0.000001	Rech.
2	MO-AD I(0.2-0.8)	0.008107	Rech.	MO-AD I(0.2-0.8)	0.018603	Rech.	MO-AD I(0.2-0.8)	0.018603	Rech.

Tabla 3.16: Tabla de Finner con $\alpha = 0.05$ para los modelos en ECM_{TST}, $R_P\_MR_{AC}$ y $\#R_F$ en el punto de máxima interpretabilidad MAX_{INT} usando la BR de FS-MOGUL

	ECM_{TST}			$R_P\_MR_{AC}$			$\#R_F$		
i	Algoritmo	*p-finner*	Hipot	Algoritmo	*p-finner*	Hipot	Algoritmo	*p-finner*	Hipot
1	MO-AD I(0.3-0.7)	0.000389	Rech.	MO-AD I(0.1-0.9)	0.000001	Rech.	MO-AD I(0.1-0.9)	0.000005	Rech.
2	MO-AD I(0.2-0.8)	0.116664	Acep.	MO-AD I(0.2-0.8)	0.010187	Rech.	MO-AD I(0.2-0.8)	0.047556	Rech.

Tabla 3.17: Tabla de Finner con $\alpha = 0.05$ para los modelos en ECM_{TST}, $R_P\_MR_{AC}$ y $\#R_F$ en el punto de mediana precisión $MEDIA_{(INT\ /\ PRE)}$ usando la BR de WM

	ECM_{TST}			$R_P\_MR_{AC}$			$\#R_F$		
i	Algoritmo	*p-finner*	Hipot	Algoritmo	*p-finner*	Hipot	Algoritmo	*p-finner*	Hipot
1	MO-AD I(0.3-0.7)	0.000001	Rech.	MO-AD I(0.1-0.9)	0.000001	Rech.	MO-AD I(0.1-0.9)	0.000001	Rech.
2	MO-AD I(0.2-0.8)	0.010187	Rech.	MO-AD I(0.2-0.8)	0.010187	Rech.	MO-AD I(0.2-0.8)	0.010787	Rech.

Tabla 3.18: Tabla de Finner con $\alpha = 0.05$ para los modelos en ECM_{TST}, $R_P\_MR_{AC}$ y $\#R_F$ en el punto de mediana precisión $MEDIA_{(INT\ /\ PRE)}$ usando la BR de FS-MOGUL

	ECM_{TST}			$R_P\_MR_{AC}$			$\#R_F$		
i	Algoritmo	*p-finner*	Hipot	Algoritmo	*p-finner*	Hipot	Algoritmo	*p-finner*	Hipot
1	MO-AD I(0.3-0.7)	0.000002	Rech.	MO-AD I(0.1-0.9)	0.000001	Rech.	MO-AD I(0.1-0.9)	0.000001	Rech.
2	MO-AD I(0.2-0.8)	0.030984	Rech.	MO-AD I(0.2-0.8)	0.010187	Rech.	MO-AD I(0.2-0.8)	0.010787	Rech.

En estas tablas, los algoritmos están ordenados de forma ascendente por el valor del estadístico de contraste calculado por el test, *p-finner*, obtenido con respecto al algoritmo con mejor ranking en el test de Friedman (algoritmo de control). A modo de ejemplo, en la Tabla 3.13, el test de Finner compara en el punto de máxima precisión (MAX $_{PRE}$) el algoritmo con el mejor ranking en ECM_{TST} según el test de Friedman, MO-$AD_{I(0.1-0.9)}$ (algoritmo de control), con los otros dos. Lo mismo hace en $R_P\_MR_{AC}$ y $\#R_F$, comparando el algoritmo con mejor ranking en ambas métricas MO-$AD_{I(0.3-0.7)}$, con los otros dos.

El test de Finner rechaza la hipótesis de igualdad (hipótesis nula) cuando el estadístico de contraste *p-finner* es < 0.05. Observando las tablas, podemos ver que el test de Finner rechaza la hipótesis de igualdad en precisión (ECM_{TST}) en los tres puntos más representativos del Pareto cuando el algoritmo de control es MO-$AD_{I(0.1-0.9)}$. La excepción ocurre en el punto más interpretable cuando la BR inicial es FS-MOGUL, donde los resultados del test indica que no hay diferencias significativas con respecto a MO-$AD_{I(0.2-0.8)}$, por lo que en dicho punto ambos modelos son estadísticamente iguales.

La hipótesis de igualdad también es rechazada con el modelo MO-$AD_{I(0.3-0.7)}$ para el índice $R_P\_MR_{AC}$ en todos los puntos representativos del Pareto. Con respecto a la métrica $\#R_F$, la hipótesis de igualdad es rechazada con el modelo MO-$AD_{I(0.3-0.7)}$ en el punto de máxima y mediana interpretabilidad MAX $_{INT}$ y MEDIA $_{(INT\ /\ PRE)}$, pero no en el punto de máxima precisión MAX $_{PRE}$.

En cualquier caso, tanto con la BR de WM como de FS-MOGUL, para todos los puntos representativos del Pareto, el mejor algoritmo (algoritmo de control) en MAX $_{PRE}$ es MO-$AD_{I(0.1-0.9)}$, mientras que en $R_P\_MR_{AC}$ y $\#R_F$ es MO-$AD_{I(0.3-0.7)}$.

Al examinar los resultados del análisis estadístico (Tablas 3.7 a 3.18), junto a las Tablas 3.5 y 3.6, se puede concluir lo siguiente:

- La mejor interpretabilidad se obtiene con los umbrales U_B = 0.3 y U_A = 0.7 (el índice de interpretabilidad y métricas, son mejores con ellos), mientras que la mejor precisión se obtiene con U_B = 0.1 y U_A = 0.9. Creemos que esto se debe a que la configuración más agresiva (U_B = 0.3 y U_A = 0.7) permite que el algoritmo de aprendizaje ataque el índice de interpretabilidad con más fuerza, reduciendo de forma simultánea las reglas con peso y las reglas activas, convergiendo rápidamente a las zonas con las soluciones más interpretables. Por el contrario, la configuración más conservadora (U_B = 0.1 y U_A = 0.9) obliga al algoritmo de aprendizaje a buscar en una zona menos interpretable, favoreciendo una más fácil convergencia con las soluciones más precisas. Este es un

resultado coherente porque un valor de umbral U_B = 0.3 reduce de forma considerable el número de reglas, y un valor de umbral U_A = 0.7 el de número de reglas con peso asociado, reduciendo ambos umbrales juntos el rango disponible para las reglas con peso al intervalo [0.3, 0.7], que es más pequeño que el intervalo [0.1, 0.9] que estaría disponible con la otra configuración. Al tener un menor rango de pesos asociados disponibles es lógico que el sistema obtenido sea menos preciso, al ser menos flexible.

- Los resultados obtenidos en el punto más preciso (MAX $_{PRE}$) por los tres umbrales son similares en cuanto a precisión. Como se ha comentado antes, la mejor precisión se obtiene con U_B = 0.1 y U_A = 0.9, pero cuando se usan las otras dos configuraciones de umbrales, la precisión no empeora tanto, mientras que la interpretabilidad si que mejora sustancialmente (por ejemplo, el número de reglas con peso disminuye bastante en la mayoría de los conjuntos de datos, llegando a ser aproximadamente la mitad del número total de reglas).

La Figura 3.4 muestra la comparación entre los frentes de Pareto medios obtenidos por el algoritmo multiobjetivo propuesto (MO-AD$_I$) con los diferentes configuraciones de umbrales estudiados, utilizando como BR inicial la obtenida por WM, para los problemas MOR y STO. Cada Pareto ha sido dibujado utilizando la media de los tres puntos más representativos (MAX $_{INT}$, MEDIA $_{(INT\ /\ PRE)}$ y MAX $_{PRE}$) de cada Pareto obtenido en las 30 pruebas realizadas para cada AGMO y conjunto de datos.

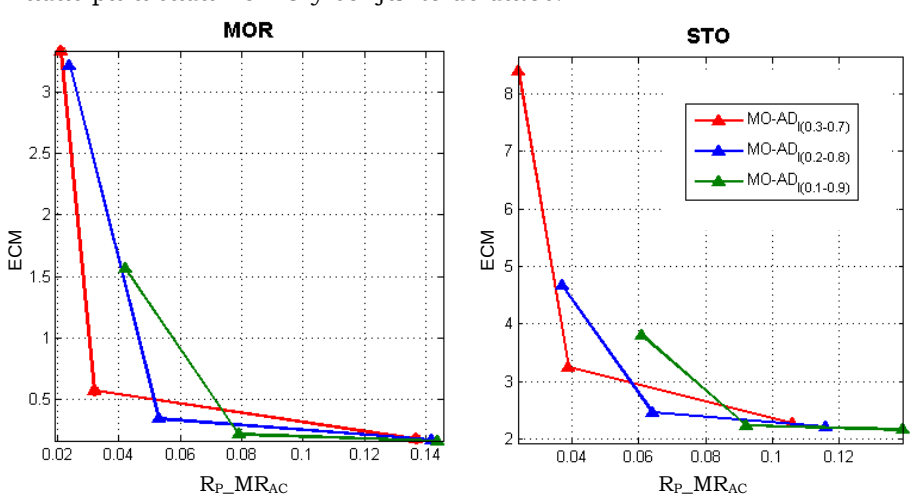


Figura 3.4: Frente de Pareto medio obtenido para los problemas STO y MOR en el plano precisión-interpretabilidad con BR inicial obtenida por WM.

Los resultados mostrados en dicha figura son extensivos al resto de conjuntos de datos utilizando como BR inicial tanto WM como FS-MOGUL. Como podemos ver, las aproximaciones de los frentes de Pareto obtenidos por U_B = 0.1 y U_A = 0.9 son en general más precisos que los obtenidos por U_B = 0.3 y U_A = 0.7 que son, como se esperaba, más interpretables. Además, se puede observar que los frentes de Pareto obtenidos por MO-$AD_{I(0.3-0.7)}$ son más amplios que los obtenidos con MO-$AD_{I(0.1-0.9)}$. Esto es coherente con la segunda conclusión previa obtenida estudiando las tablas: las mejores precisiones son alcanzadas por el modelo MO-$AD_{I(0.1-0.9)}$, pero MO-$AD_{I(0.3-0.7)}$ está cerca de éste en precisión (MAX $_{PRE}$), y alcanza el mayor nivel en interpretabilidad (MAX $_{INT}$).

3.4.2.2. Comparación de la Propuesta Multiobjetivo vs Propuestas de un Solo Objetivo

En esta subsección se analiza el rendimiento y funcionamiento de nuestra propuesta multiobjetivo contra las propuestas de un solo objetivo.

Para realizar el análisis se ha elegido la configuración U_B = 0.1 y U_A = 0.9, por ser la que mejores resultados logra en precisión, como hemos visto antes, para así poder compararlas con las propuestas de un solo objetivo, que utilizan como único objetivo, uno basado en la precisión.

Es evidente que no tendría sentido considerar el índice de interpretabilidad en las propuestas de un solo objetivo que están orientadas a precisión, ya que este índice afecta a la interpretabilidad y éste no es un objetivo en ellos (asumiendo que los enfoques orientados a la precisión tienen los peores valores de interpretabilidad medida mediante el índice $R_P\_MR_{AC}$). Sin embargo, con el objetivo de comparar el punto más preciso en la propuesta multiobjetivo con los de un único objetivo, hemos introducido también estos índices en las propuestas de un sólo objetivo (SO-AD y SO-AD_I) con el fin de ver cómo la precisión de dicho índice se ve afectada.

La Tabla 3.19 muestra los resultados medios alcanzados por las propuestas de un solo objetivo SO-AD y SO-$AD_{I(0.1-0.9)}$ junto con la propuesta multiobjetivo MO-$AD_{I(0.1-0.9)}$. Los resultados de MO-$AD_{I(0.1-0.9)}$ corresponden a los obtenidos en el punto de máxima precisión (MAX $_{PRE}$) de las Tablas 3.5 y 3.6.

Tabla 3.19: Resultados obtenidos por el modelo monoobjetivo estándar orientado a precisión (SO-AD) vs modelo monoobjetivo ($SO\text{-}AD_{I(0.1\text{-}0.9)}$) y modelo multiobjetivo ($MO\text{-}AD_{I(0.1\text{-}0.9)}$) con el mecanismo de mejora de la interpretabilidad propuesto utilizando BR de WM y FS-MOGUL

		WM						FS-MOGUL					
Datos	Modelo	ECM_{ENT}	ECM_{TST}	$\#R_F$	$R_{P\_}MR_{AC}$	($\#R_P$	MR_{AC})	ECM_{ENT}	ECM_{TST}	$\#R_F$	$R_{P\_}MR_{AC}$	($\#R_P$	MR_{AC})
PLA	SO-AD	1.694533	1.749531	14.80	0.913	(14.80	3.30)	1.274698	1.295137	75.40	0.904	(75.40	19.25)
	$SO\text{-}AD_I$	1.705751	1.765311	14.33	0.813	(11.97	3.27)	1.252672	1.269589	58.97	0.645	(52.83	14.02)
	$MO\text{-}AD_I$	1.699226	**1.759935**	14.0	0.771	(10.80	3.25)	1.163733	**1.201293**	40.5	0.379	(28.03	9.16)
QUA	SO-AD	0.021303	0.022412	53.60	0.892	(53.60	6.27)	0.017394	0.018187	227.60	0.829	(227.60	32.89)
	$SO\text{-}AD_I$	0.021163	0.022351	42.40	0.633	(38.17	4.43)	0.017366	0.018170	205.97	0.720	(197.87	28.48)
	$MO\text{-}AD_I$	0.020675	**0.021944**	32.6	0.448	(22.67	3.79)	0.017111	**0.018167**	170.8	0.553	(141.50	24.26)
ELE	SO-AD	36455	38917	65.00	0.880	(65.00	10.66)	43517	47035	88.80	0.759	(88.80	10.04)
	$SO\text{-}AD_I$	36546	39295	58.17	0.738	(54.03	9.02)	43444	46938	80.43	0.662	(74.63	9.37)
	$MO\text{-}AD_I$	32518	**35633**	45.2	0.469	(33.27	5.96)	39495	**42072**	52.0	0.352	(33.80	6.28)
AU6	SO-AD	2.917724	**6.002718**	116.00	0.777	(116.00	9.10)	3.252887	6.565788	170.60	0.731	(170.60	13.54)
	$SO\text{-}AD_I$	2.914147	6.034773	104.67	0.673	(99.17	8.06)	3.245635	6.629009	154.97	0.640	(146.47	12.36)
	$MO\text{-}AD_I$	2.568845	6.457802	73.6	0.387	(48.77	5.77)	2.941543	**6.263927**	111.0	0.389	(79.53	9.18)
AU8	SO-AD	9.222880	10.51600	70.60	0.771	(70.60	9.65)	7.804103	8.910248	78.80	0.812	(78.80	11.05)
	$SO\text{-}AD_I$	9.204716	10.72178	59.90	0.621	(56.07	7.97)	7.796410	9.112414	66.13	0.648	(61.93	9.04)
	$MO\text{-}AD_I$	8.897037	**10.75952**	36.2	0.303	(23.00	4.99)	7.386325	**9.000788**	33.3	0.283	(22.67	4.81)
ANA	SO-AD	0.008360	0.010359	72.40	0.752	(72.40	2.01)	0.023771	0.026515	211.20	0.740	(211.20	7.00)
	$SO\text{-}AD_I$	0.010606	0.012647	70.30	0.698	(64.83	2.00)	0.020551	0.023209	189.73	0.617	(183.13	5.34)
	$MO\text{-}AD_I$	0.007488	**0.009588**	67.5	0.659	(59.50	1.98)	0.007158	**0.009335**	133.0	0.358	(91.53	4.12)
ABA	SO-AD	5.101607	5.078695	68.20	0.912	(68.20	15.64)	6.374271	6.385199	50.20	0.854	(50.20	9.40)
	$SO\text{-}AD_I$	5.077808	5.059224	56.57	0.707	(51.90	12.40)	6.235069	6.249418	40.30	0.614	(36.87	6.59)
	$MO\text{-}AD_I$	4.753375	**4.789237**	26.5	0.225	(13.47	4.79)	6.198179	**6.219666**	24.9	0.311	(15.07	4.32)
CON	SO-AD	58.142348	62.98518	135.40	0.692	(135.40	19.63)	61.601930	65.901353	124.40	0.697	(124.40	19.30)
	$SO\text{-}AD_I$	57.845984	62.69151	118.10	0.569	(110.30	16.45)	61.289333	65.672337	108.17	0.568	(100.80	15.99)
	$MO\text{-}AD_I$	50.3219	**57.5966**	59.9	0.219	(37.07	8.39)	56.380503	**62.489223**	62.6	0.257	(39.37	9.66)
STP	SO-AD	3.784825	3.810319	122.80	0.849	(122.80	36.25)	4.371739	4.459575	45.60	0.870	(45.60	14.49)
	$SO\text{-}AD_I$	3.556481	3.571853	98.40	0.643	(91.97	27.91)	4.302168	4.399084	34.57	0.567	(30.03	9.32)
	$MO\text{-}AD_I$	2.052056	**2.154450**	36.7	0.139	(15.43	7.94)	4.159594	**4.255350**	25.0	0.287	(11.73	6.19)
WAN	SO-AD	7.775341	8.005151	156.00	0.845	(156.00	60.23)	14.300552	14.573703	33.40	0.864	(33.40	13.09)
	$SO\text{-}AD_I$	7.626420	7.883073	136.73	0.713	(129.27	52.19)	14.194849	14.484166	25.43	0.622	(23.00	10.00)
	$MO\text{-}AD_I$	5.983361	**6.674879**	48.6	0.190	(27.90	17.52)	13.774738	**14.009191**	16.1	0.361	(12.27	6.19)
WIZ	SO-AD	3.331755	3.574799	104.80	0.883	(104.80	35.21)	4.034584	4.544982	52.40	0.863	(52.40	18.26)
	$SO\text{-}AD_I$	3.316957	3.583501	92.80	0.750	(87.43	30.59)	3.943292	4.418400	42.47	0.659	(39.60	14.19)
	$MO\text{-}AD_I$	2.629827	**3.068601**	43.2	0.297	(27.57	15.13)	3.413844	**3.856717**	25.0	0.351	(19.80	8.12)
MOR	SO-AD	0.329428	0.328112	77.60	0.868	(77.60	29.29)	0.361349	0.364585	31.40	0.857	(31.40	11.25)
	$SO\text{-}AD_I$	0.263595	0.273204	58.50	0.633	(54.50	22.42)	0.346304	0.347567	21.20	0.557	(19.43	7.80)
	$MO\text{-}AD_I$	0.153782	**0.165901**	16.5	0.144	(9.57	6.50)	0.267971	**0.275199**	12.8	0.305	(9.03	5.06)
TRE	SO-AD	0.428898	0.432897	75.00	0.884	(75.00	28.12)	0.595556	0.587049	33.00	0.827	(33.00	11.39)
	$SO\text{-}AD_I$	0.374909	0.378188	57.00	0.653	(53.30	21.73)	0.578306	0.573042	23.00	0.553	(20.63	8.36)
	$MO\text{-}AD_I$	0.202813	**0.232601**	20.3	0.177	(12.23	7.00)	0.448043	**0.436540**	10.4	0.229	(7.20	4.17)

Para comparar la propuesta multiobjetivo con las propuestas de un solo objetivo, realizamos un análisis estadístico en los diferentes índices y métricas para chequear si hay diferencias significativas.

Las Tablas 3.20 y 3.21 muestra los rankings (obtenidos a partir del test de Friedman con un nivel de confianza $\alpha = 0.05$) de los diferentes modelos considerados en el estudio para las BR de WM y FS-MOGUL, respectivamente.

Tabla 3.20: Rankings obtenidos mediante el test de Friedman para diferentes valores de umbrales en las métricas ECM_{TST}, $R_{P}\_MR_{AC}$ y $\#R_F$ en el punto de máxima precisión MAX $_{PRE}$ usando la BR inicial de WM

Algoritmo	Ranking en ECM_{TST} (*p-fried*: 0.024914)	Ranking en $R_{P}\_MR_{AC}$ (*p-fried*:0.000002)	Ranking en $\#R_F$ (*p-fried*: 0.000002)
$MO\text{-}AD_{I(0.1\text{-}0.9)}$	1.3846	1	1
$SO\text{-}AD_{I(0.1\text{-}0.9)}$	2.3077	2	2
SO-AD	2.3077	3	3

Tabla 3.21: Ranking obtenido a través del test de Friedman para diferentes valores de umbrales en las métricas ECM_{TST}, $R_{P}\_MR_{AC}$ y $\#R_F$ en el punto de máxima precisión MAX $_{PRE}$ usando la BR inicial de FS-MOGUL

Algoritmo	Ranking en ECM_{TST} (*p-fried*: 0.0000072)	Ranking en $R_{P}\_MR_{AC}$ (*p-fried*:0.000002)	Ranking en $\#R_F$ (*p-fried*: 0.000002)
$MO\text{-}AD_{I(0.1\text{-}0.9)}$	1.0769	1	1
$SO\text{-}AD_{I(0.1\text{-}0.9)}$	2.1538	2	2
SO-AD	2.7692	3	3

Los *p*-valores (*p-fried*) calculados por el test de Friedman, tanto con la BR de WM como con FS-MOGUL, al ser todos < 0.05, indican que existen diferencias estadísticas entre los resultados en ECM_{TST}, $R_{P}\_MR_{AC}$ y $\#R_F$ respectivamente. En todos los casos $MO\text{-}AD_{I(0.1\text{-}0.9)}$ es el mejor en el ranking y en todos los casos, el test de Finner aplicado a posteriori (Tablas 3.22 y 3.23) rechaza la hipótesis nula con todos los métodos de un solo objetivo, lo que corrobora que las diferencias entre los métodos son estadísticamente significativas y que por tanto se puede afirmar que el modelo multiobjetivo $MO\text{-}AD_{I(0.1\text{-}0.9)}$ es estadísticamente mejor que los modelos de un solo objetivo $SO\text{-}AD_{I(0.1\text{-}0.9)}$ y SO-AD, tanto con la BR inicial de WM como con FS-MOGUL.

Tabla 3.22: Tabla Finner con $\alpha = 0.05$ para los modelos en ECM_{TST}, $R_{P}\_MR_{AC}$ y $\#R_F$ en el punto de máxima precisión MAX_{PRE} usando la BR de WM

	ECM_{TST}			$R_{P}\_MR_{AC}$			$\#R_F$		
i	Algoritmo	*p-finner*	Hipot	Algoritmo	*p-finner*	Hipot	Algoritmo	*p-finner*	Hipot
1	SO-AD	0.03686	Rech.	SO-AD	0.000001	Rech.	SO-AD	0.000001	Rech.
2	$SO\text{-}AD_{I(0.1\text{-}0.9)}$	0.03686	Rech.	$SO\text{-}AD_{I(0.1\text{-}0.9)}$	0.010787	Rech.	$SO\text{-}AD_{I(0.1\text{-}0.9)}$	0.010787	Rech.

Tabla 3.23: Tabla Finner con $\alpha = 0.05$ para los modelos en ECM_{TST}, $R_{P}\_MR_{AC}$ y $\#R_F$ en el punto de máxima precisión MAX_{PRE} usando la BR de FS-MOGUL

	ECM_{TST}			$R_{P}\_MR_{AC}$			$\#R_F$		
i	Algoritmo	*p-finner*	Hipot	Algoritmo	*p-finner*	Hipot	Algoritmo	*p-finner*	Hipot
1	SO-AD	0.000032	Rech.	SO-AD	0.000001	Rech.	SO-AD	0.000001	Rech.
2	$SO\text{-}AD_{I(0.1\text{-}0.9)}$	0.006040	Rech.	$SO\text{-}AD_{I(0.1\text{-}0.9)}$	0.010787	Rech.	$SO\text{-}AD_{I(0.1\text{-}0.9)}$	0.010787	Rech.

De esta forma, observando los resultados de la Tabla 3.19 y el análisis estadístico (Tablas 3.20, 3.21, 3.22 y 3.23), podemos concluir lo siguiente:

- La propuesta multiobjetivo, en el punto de máxima precisión (MAX_{PRE}), estadísticamente mejora la precisión de las propuestas de un solo objetivo orientadas a precisión. Este es un resultado importante, ya que confirma que el uso del mecanismo propuesto para mejorar la interpretabilidad de los métodos de defuzzificación adaptativa aplicado a los AGMO mejora no sólo la interpretabilidad sino también la precisión, respecto a las propuestas monoobjetivos.

- Como era de esperar, las medidas de interpretabilidad de la propuesta multiobjetivo para el punto de máxima precisión (MAX_{PRE}), son mejores que las propuestas de un solo objetivo orientadas a precisión.

- Teniendo en cuenta únicamente las propuestas de un solo objetivo, el método $SO\text{-}AD_{I(0.1\text{-}0.9)}$ obtiene mejores resultados que SO-AD. De esta forma, el mecanismo para mejorar la interpretabilidad propuesto también consigue mejorar la precisión de los métodos de un solo objetivo cuando usa como umbrales los valores $U_B = 0.1$ y $U_A = 0.9$.

La Figura 3.5 muestra los tres puntos representativos del Pareto (MAX $_{INT}$, MEDIA $_{(INT\ /\ PRE)}$ y MAX $_{PRE}$) y la aproximación del frente del Pareto de la propuesta multiobjetivo MO-AD$_{I(0.1\text{-}0.9)}$, comparadas con la solución obtenida por las propuestas de un solo objetivo SO-AD y SO-AD$_{I(0.1\text{-}0.9)}$, para el problema TRE, utilizando como BR inicial la obtenida por WM. Los resultados mostrados en dicha figura, obtenidos como la media de los resultados alcanzados en las 30 pruebas realizadas por cada propuesta, son extensivos al resto de conjuntos de datos utilizando como BR inicial tanto WM como FS-MOGUL.

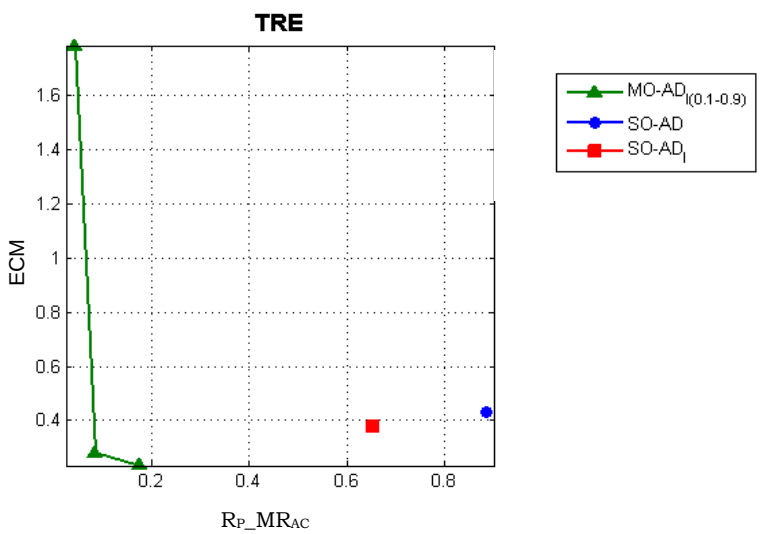


Figura 3.5: Frente de Pareto medio obtenido por el problema TRE en el plano precisión-interpretabilidad con como BR inicial la obtenida por WM.

A fin de demostrar que el número de reglas finales $\#R_F$ evoluciona de la misma manera que el índice $R_P\_MR_{AC}$, en el frente de Pareto aproximado generado por MO-AD$_{I(0.1\text{-}0.9)}$ mostramos en la Figura 3.6 un ejemplo representativo del problema WIZ, con ambos índices.

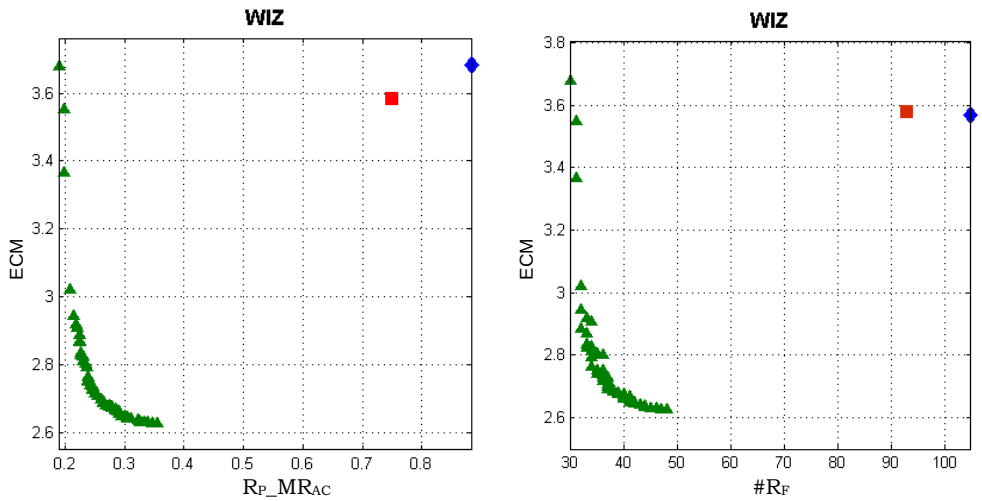


Figura 3.6: Ejemplo de Frente de Pareto obtenido con el problema WIZ.

Las soluciones de MO-$AD_{I(0.1\text{-}0.9)}$ han sido representadas en dos dimensiones mediante la proyección de dichas soluciones en el plano precisión-número de reglas finales ($\#R_F$) y en el plano precisión-número de reglas con peso ($\#R_P$) junto con la solución de un solo objetivo alcanzadas por SO-AD y SO-$AD_{I(0.1\text{-}0.9)}$. A fin de mostrar toda la información, las soluciones dominadas obtenidas a partir de las respectivas proyecciones no han sido eliminadas de dichas figuras. Este tipo de proyecciones ha sido usada por algunos investigadores para representar gráficamente las soluciones obtenidas cuando se optimiza simultáneamente tres o más objetivos [ADH+09, GAH10].

3.4.3. Segunda Parte del Estudio Experimental

En esta subsección analizamos el comportamiento práctico del segundo modelo propuesto que hace uso de los umbrales adaptativos. Para ello, se analiza el comportamiento práctico del segundo modelo (umbrales adaptativos) comparándolo con el primer modelo (umbrales fijos).

Los modelos considerados para esta segunda parte del estudio experimental se resumen en la Tabla 3.24.

Tabla 3.24: Modelos considerados para la comparación

Modelo	**Descripción**
	SDEs
MO-AD_I	Defuzzificación adaptativa con el mecanismo de umbrales fijo para mejorar la interpretabilidad
MO-$AD_{I(A)}$	Defuzzificación adaptativa con el mecanismo de umbrales adaptativos para mejorar la interpretabilidad

Para llevar a cabo el estudio, la propuesta multiobjetivo con el mecanismo de mejora de la interpretabilidad mediante el uso de umbrales adaptativos (MO-$AD_{I(A)}$), es comparada con las dos configuraciones de umbrales de la propuesta multiobjetivo con el mecanismo de mejora de la interpretabilidad de umbrales fijos con las que se obtenía las soluciones más precisas (MO-$AD_{I(0.1-0.9)}$) y las soluciones más interpretables (MO-$AD_{I(0.3-0.7)}$).

Con respecto a la configuración de los AGMOs, la configuración usada es la misma que la empleada en la primera parte del estudio experimental (véase el apartado 3.4.1. anterior).

Los valores de los parámetros de entrada considerados por los modelos multiobjetivo usados (MO-$AD_{I(A)}$ y MO-AD_I) son: tamaño de la población de 200 individuos, tamaño de la población externa de 200 individuos, 200000 evaluaciones y 0.2 como probabilidad de mutación.

Las Tablas 3.25 y 3.26 muestran los resultados obtenidos por las dos configuraciones de la propuesta multiobjetivo con umbrales fijos MO-$AD_{I(0.1-0.9)}$ y MO-$AD_{I(0.3-0.7)}$ con las que se obtiene respectivamente, las soluciones más precisas y las soluciones más interpretables y los resultados obtenidos por la propuesta con umbrales adaptativos MO-$AD_{I(A)}$, utilizando como BRs iniciales las obtenidas por WM y FS-MOGUL, en los tres puntos más representativos del Pareto: el más interpretable (MAX $_{INT}$), el mediano (MEDIA $_{(INT\ /\ PRE)}$) y el más preciso (MAX $_{PRE}$).

Tabla 3.25: Resultados obtenidos por MO-AD_I para diferentes configuraciones de umbrales y por MO-$AD_{I(A)}$ usando la BR inicial de WM

		MAX $_{INT}$						MEDIA ($_{INT\ /\ PRE}$)						MAX $_{PRE}$					
Conjunto de Datos	MO-AD_I U_B - U_A	ECM_{ENT}	ECM_{TST}	$\#R_F$	R_P_MR_{AC}	($\#R_P$	MR_{AC})	ECM_{ENT}	ECM_{TST}	$\#R_F$	R_P_MR_{AC}	($\#R_P$	MR_{AC})	ECM_{ENT}	ECM_{TST}	$\#R_F$	R_P_MR_{AC}	($\#R_P$	MR_{AC})
PLA	0.1 - 0.9	6.947800	6.940028	6.1	0.128	(0.00	1.02)	2.073691	2.091053	10.6	0.385	(2.23	2.48)	1.699226	1.759935	14.0	0.771	(10.80	3.25)
	0.3 - 0.7	7.042636	7.042633	6.0	0.125	(0.00	1.00)	2.543587	2.545364	10.4	0.309	(0.60	2.31)	1.830941	1.900387	12.4	0.551	(6.13	2.75)
	Adaptativo	3.516398	3.587155	5.00	0.132	(0.00	1.05)	2.157240	2.168258	9.67	0.344	(1.70	2.30)	1.690599	1.745825	14.07	0.783	(11.13	3.26)
QUA	0.1 - 0.9	0.023570	0.024276	22.7	0.179	(4.93	2.12)	0.020914	0.022153	27.2	0.287	(10.60	3.01)	0.020675	0.021944	32.6	0.448	(22.67	3.79)
	0.3 - 0.7	0.026361	0.027337	17.7	0.114	(0.13	1.80)	0.021700	0.022975	24.7	0.193	(3.13	2.62)	0.020897	0.022139	31.4	0.380	(17.37	3.49)
	Adaptativo	0.071487	0.071611	6.10	0.018	(0.00	0.28)	0.036666	0.037420	9.97	0.063	(0.23	0.97)	0.020688	0.022138	32.97	0.450	(23.70	3.66)
ELE	0.1 - 0.9	57867	60608	35.8	0.167	(7.37	3.10)	35428	39094	38.6	0.285	(16.30	4.48)	32518	35633	45.2	0.469	(33.27	5.96)
	0.3 - 0.7	143152	150408	20.0	0.063	(0.03	1.77)	64459	70994	27.6	0.086	(1.17	2.16)	33433	35677	45.4	0.421	(25.97	6.19)
	Adaptativo	4515380	4510411	13.03	0.014	(0.00	0.39)	308668	326805	17.60	0.049	(0.00	1.39)	32809	35539	45.00	0.482	(32.57	6.49)
AU6	0.1 - 0.9	3.487613	6.568578	84.6	0.214	(6.60	6.07)	2.746415	6.482094	73.4	0.293	(28.57	5.57)	2.568845	6.457802	73.6	0.387	(48.77	5.77)
	0.3 - 0.7	5.551319	9.068401	40.9	0.086	(0.27	2.78)	3.326684	7.252765	52.2	0.132	(3.43	3.84)	2.687347	6.430410	72.5	0.325	(35.53	5.61)
	Adaptativo	4.825753	8.955610	47.60	0.109	(1.70	3.33)	3.024116	7.687152	56.40	0.189	(13.70	4.26)	2.609892	6.387879	78.97	0.441	(58.43	6.20)
AU8	0.1 - 0.9	11.281517	13.152617	32.1	0.132	(3.00	3.93)	9.102294	11.28545	33.9	0.201	(10.80	4.45)	8.897037	10.759520	36.2	0.303	(23.00	4.99)
	0.3 - 0.7	21.088986	23.062541	14.8	0.054	(0.00	1.93)	10.672853	12.84438	21.4	0.090	(1.13	2.91)	9.127924	11.186454	35.9	0.255	(17.23	4.73)
	Adaptativo	19.514876	20.801083	12.77	0.045	(0.30	1.52)	9.739728	11.83919	22.57	0.120	(5.23	2.97)	8.887143	10.698300	37.17	0.325	(25.63	5.13)
ANA	0.1 - 0.9	0.647718	0.646577	63.7	0.190	(1.77	1.43)	0.107990	0.109300	67.1	0.267	(3.67	1.94)	0.007488	0.009588	67.5	0.659	(59.50	1.98)
	0.3 - 0.7	0.955815	0.960777	45.6	0.137	(0.03	1.09)	0.438822	0.440663	50.6	0.195	(0.70	1.52)	0.007551	0.009691	64.8	0.531	(41.37	1.96)
	Adaptativo	0.360127	0.360551	38.87	0.054	(0.00	0.43)	0.104234	0.110433	49.33	0.160	(2.07	1.17)	0.007347	0.009452	66.80	0.656	(59.13	1.98)
ABA	0.1 - 0.9	7.613683	7.712487	15.2	0.083	(2.00	2.58)	4.836747	4.858605	19.8	0.142	(6.23	3.66)	4.753375	4.789237	26.5	0.225	(13.47	4.79)
	0.3 - 0.7	14.830294	14.893969	10.3	0.044	(0.00	1.66)	6.432483	6.432756	13.4	0.066	(0.37	2.42)	4.821530	4.845469	23.1	0.184	(9.37	4.39)
	Adaptativo	11.779818	11.777811	5.30	0.027	(0.03	1.01)	5.504223	5.571520	9.70	0.066	(1.20	2.19)	4.761915	4.779487	25.87	0.232	(13.97	4.91)
CON	0.1 - 0.9	70.753300	76.129910	83.8	0.131	(6.30	10.97)	55.1029	61.1530	71.5	0.169	(20.23	9.65)	50.3219	57.5966	59.9	0.219	(37.07	8.39)
	0.3 - 0.7	128.97494	134.53928	25.9	0.030	(0.03	3.06)	73.2893	83.2969	33.2	0.049	(1.60	4.37)	52.9665	59.9898	60.0	0.198	(31.23	8.43)
	Adaptativo	121.13485	127.93996	28.87	0.037	(0.53	3.53)	62.424558	69.86624	38.17	0.071	(4.97	5.38)	50.580760	57.292015	64.60	0.254	(44.00	9.34)
STP	0.1 - 0.9	3.708981	3.794866	25.7	0.061	(3.20	5.04)	2.129880	2.227016	29.8	0.092	(7.83	6.25)	2.052056	2.154450	36.7	0.139	(15.43	7.94)
	0.3 - 0.7	8.334997	8.401997	10.5	0.024	(0.07	2.44)	3.113456	3.240279	15.9	0.039	(1.13	3.58)	2.191832	2.272137	33.8	0.106	(9.67	6.91)
	Adaptativo	15.327830	15.680952	11.33	0.019	(0.00	1.96)	2.568037	2.731294	19.27	0.048	(2.27	4.00)	2.080961	2.147385	36.77	0.135	(14.63	7.88)
WAN	0.1 - 0.9	6.463753	7.172746	38.8	0.115	(12.30	13.24)	6.033333	6.780082	43.1	0.146	(18.50	15.22)	5.983361	6.674879	48.6	0.190	(27.90	17.52)
	0.3 - 0.7	16.437768	17.227865	19.2	0.029	(0.03	5.03)	7.376043	8.159211	27.6	0.052	(0.67	8.66)	6.130478	6.859088	46.0	0.140	(14.50	16.28)
	Adaptativo	13.890007	13.886866	26.77	0.052	(0.67	8.81)	6.702816	7.416840	34.43	0.085	(5.10	11.93)	6.033621	6.791114	46.00	0.165	(21.27	16.96)
WIZ	0.1 - 0.9	3.846971	4.359892	30.5	0.147	(7.90	10.00)	2.718277	3.149809	36.1	0.211	(14.73	12.86)	2.629827	3.068601	43.2	0.297	(27.57	15.13)
	0.3 - 0.7	24.760387	25.488658	14.8	0.034	(0.03	3.11)	5.940635	6.631707	19.3	0.062	(0.73	5.35)	2.734843	3.302554	45.1	0.265	(20.30	15.39)
	Adaptativo	19.506770	20.197547	17.27	0.056	(0.23	5.01)	3.220616	3.690454	25.80	0.115	(3.57	8.98)	2.643355	3.061785	44.60	0.311	(29.27	15.63)
MOR	0.1 - 0.9	1.590566	1.564395	9.2	0.042	(0.77	2.97)	0.201851	0.219410	12.9	0.079	(2.83	4.83)	0.153782	0.165901	16.5	0.144	(9.57	6.50)
	0.3 - 0.7	3.180415	3.323322	5.6	0.021	(0.00	1.64)	0.551852	0.574422	8.5	0.039	(0.43	2.86)	0.163368	0.175955	18.2	0.137	(7.90	6.86)
	Adaptativo	2.080311	2.089618	5.23	0.020	(0.03	1.60)	0.339201	0.355028	8.27	0.041	(0.53	3.01)	0.165634	0.186429	17.07	0.148	(9.60	6.80)
TRE	0.1 - 0.9	1.720608	1.786207	10.3	0.045	(0.67	2.96)	0.248659	0.280729	14.2	0.086	(3.10	4.75)	0.202813	0.232601	20.3	0.177	(12.23	7.00)
	0.3 - 0.7	4.611414	4.668308	5.5	0.025	(0.00	1.80)	1.005509	1.023034	7.9	0.039	(0.23	2.72)	0.212469	0.238197	18.9	0.144	(7.93	6.68)
	Adaptativo	3.887324	3.855478	5.20	0.019	(0.00	1.41)	0.796229	0.820500	7.90	0.040	(0.33	2.74)	0.223408	0.229770	22.17	0.199	(13.30	8.09)

Tabla 3.26: Resultados obtenidos por MO-AD$_I$ para diferentes configuraciones de umbrales y por MO-AD$_{I(A)}$ usando la BR inicial de FS-MOGUL

		MAX $_{INT}$					MEDIA ($_{INT\ /\ PRE}$)					MAX $_{PRE}$				
Conjunto de Datos	MO-AD$_I$ U_B - U_A	ECM$_{ENT}$	ECM$_{TST}$	#R$_F$	R$_P$_MR$_{AC}$	(#R$_P$ MR$_{AC}$)	ECM$_{ENT}$	ECM$_{TST}$	#R$_F$	R$_P$_MR$_{AC}$	(#R$_P$ MR$_{AC}$)	ECM$_{ENT}$	ECM$_{TST}$	#R$_F$	R$_P$_MR$_{AC}$	(#R$_P$ MR$_{AC}$)
PLA	0.1 - 0.9	1.574572	1.623150	21.9	0.130	(4.80 4.68)	1.171741	1.207446	29.3	0.216	(11.57 6.61)	1.163733	1.201293	40.5	0.379	(28.03 9.16)
	0.3 - 0.7	5.290166	5.346323	7.0	0.029	(0.00 1.38)	1.754391	1.810790	13.2	0.063	(0.10 2.97)	1.166274	1.204826	33.5	0.251	(14.87 7.24)
	Adaptativo	3.231820	3.335011	12.67	0.059	(0.50 2.65)	1.281801	1.301098	21.33	0.129	(4.40 4.73)	1.163860	1.200922	41.47	0.387	(28.50 9.41)
QUA	0.1 - 0.9	0.021684	0.022334	199.7	0.292	(13.63 26.19)	0.017291	0.018149	177.8	0.385	(71.50 22.83)	0.017111	0.018167	170.8	0.553	(141.50 24.26)
	0.3 - 0.7	0.017608	0.018610	86.4	0.129	(13.13 10.08)	0.017135	0.018257	98.0	0.174	(22.40 12.52)	0.017088	0.018183	117.2	0.268	(48.00 16.29)
	Adaptativo	0.017483	0.018617	99.30	0.160	(16.63 12.46)	0.017136	0.018311	105.77	0.205	(28.47 14.31)	0.017086	0.018236	122.67	0.311	(62.93 17.35)
ELE	0.1 - 0.9	64241	67141	56.0	0.129	(4.23 4.10)	42156	45526	49.5	0.228	(17.10 5.09)	39495	42072	52.0	0.352	(33.80 6.28)
	0.3 - 0.7	108723	113721	23.2	0.048	(0.07 1.85)	61099	65626	29.1	0.078	(2.07 2.56)	40766	43063	48.5	0.258	(22.10 5.18)
	Adaptativo	4026527	4027142	29.33	0.028	(0.40 0.99)	157856	155209	34.07	0.076	(1.97 2.51)	39974	42599	58.60	0.434	(43.93 7.25)
AU6	0.1 - 0.9	3.821884	6.764193	143.4	0.197	(4.50 10.80)	3.296278	6.516126	129.9	0.266	(32.60 9.97)	2.941543	6.263927	111.0	0.389	(79.53 9.18)
	0.3 - 0.7	4.202431	7.836509	58.7	0.083	(3.43 4.31)	3.124690	7.149861	69.9	0.126	(10.77 5.52)	2.976153	6.502033	88.3	0.226	(34.93 7.27)
	Adaptativo	4.124317	8.088891	66.67	0.104	(6.47 4.94)	3.086823	7.151888	77.37	0.176	(24.03 6.14)	2.949025	6.371336	109.33	0.379	(77.10 8.94)
AU8	0.1 - 0.9	10.610777	11.814385	24.6	0.120	(3.97 3.30)	7.526326	9.129575	27.7	0.185	(11.33 3.95)	7.386325	9.000788	33.3	0.283	(22.67 4.81)
	0.3 - 0.7	19.910480	20.495670	14.7	0.056	(0.00 2.00)	9.872984	11.83105	19.3	0.081	(0.37 2.76)	7.505406	9.082736	32.1	0.238	(17.17 4.52)
	Adaptativo	18.357268	19.391021	10.00	0.036	(0.27 1.20)	8.196575	9.935652	19.33	0.106	(4.20 2.76)	7.398018	8.961176	34.93	0.305	(24.73 5.10)
ANA	0.1 - 0.9	0.044437	0.047909	174.4	0.188	(12.07 4.67)	0.013463	0.015529	154.4	0.250	(44.00 4.24)	0.007158	0.009335	133.0	0.358	(91.53 4.12)
	0.3 - 0.7	0.126768	0.126480	85.7	0.094	(8.33 2.16)	0.015763	0.018568	92.5	0.125	(10.17 2.93)	0.007180	0.009461	108.3	0.216	(34.20 3.92)
	Adaptativo	0.143600	0.143536	115.67	0.078	(3.40 2.05)	0.033059	0.036042	122.23	0.119	(11.07 2.72)	0.007096	0.009207	125.23	0.321	(77.90 3.97)
ABA	0.1 - 0.9	10.034018	9.980401	15.6	0.098	(1.10 2.33)	6.252071	6.261997	20.2	0.180	(5.57 3.34)	6.198179	6.219666	24.9	0.311	(15.07 4.32)
	0.3 - 0.7	13.471095	13.490648	9.1	0.058	(0.00 1.56)	6.933849	6.956984	13.8	0.099	(0.70 2.45)	6.204145	6.222632	22.9	0.220	(7.90 3.77)
	Adaptativo	11.046818	11.051219	5.73	0.036	(0.07 0.94)	6.615973	6.628668	10.77	0.088	(1.67 1.92)	6.197562	6.219388	22.13	0.276	(13.50 3.80)
CON	0.1 - 0.9	67.672232	71.789422	86.8	0.143	(5.13 12.02)	59.010937	65.03955	66.0	0.201	(25.23 9.74)	56.380503	62.489223	62.6	0.257	(39.37 9.66)
	0.3 - 0.7	116.57328	123.10188	27.7	0.036	(0.30 3.45)	66.961798	73.82797	36.8	0.066	(2.67 5.44)	57.450287	63.305839	58.3	0.189	(24.53 8.79)
	Adaptativo	102.84723	110.31467	26.50	0.038	(0.47 3.56)	61.63860	67.58694	37.37	0.078	(4.90 5.72)	56.389478	62.40864	58.00	0.227	(33.40 9.06)
STP	0.1 - 0.9	5.405339	5.474713	13.3	0.096	(1.33 3.21)	4.259753	4.362481	18.6	0.167	(4.63 4.54)	4.159594	4.255350	25.0	0.287	(11.73 6.19)
	0.3 - 0.7	8.748136	8.816542	7.9	0.058	(0.00 2.25)	4.455629	4.560821	14.7	0.125	(2.67 3.75)	4.194782	4.309624	24.9	0.252	(8.50 6.22)
	Adaptativo	12.726416	12.816972	5.17	0.028	(0.00 1.07)	4.374793	4.494771	13.10	0.110	(2.67 3.18)	4.165515	4.247484	24.87	0.284	(11.53 6.17)
WAN	0.1 - 0.9	26.310868	26.725976	9.6	0.088	(0.53 2.86)	14.522565	14.77936	13.3	0.193	(4.40 4.46)	13.774738	14.009191	16.1	0.361	(12.27 6.19)
	0.3 - 0.7	32.988571	34.115293	7.9	0.065	(0.00 2.26)	15.540758	16.02159	11.5	0.122	(0.93 3.81)	13.948231	14.207306	15.2	0.302	(9.47 5.61)
	Adaptativo	63.395682	62.167748	5.17	0.036	(0.03 1.20)	14.83103	15.04641	9.77	0.141	(2.57 3.46)	13.770163	13.98595	15.83	0.363	(12.60 6.25)
WIZ	0.1 - 0.9	12.545764	13.543774	15.7	0.107	(2.57 4.18)	3.881428	4.467590	19.7	0.193	(7.50 6.11)	3.413844	3.856717	25.0	0.351	(19.80 8.12)
	0.3 - 0.7	34.337238	36.299840	10.5	0.054	(0.00 2.74)	6.706427	7.391861	14.5	0.091	(0.87 4.18)	3.547122	4.014734	24.5	0.291	(14.73 7.61)
	Adaptativo	68.862557	68.324376	7.23	0.029	(0.03 1.50)	11.29783	11.72132	11.77	0.093	(1.80 3.86)	3.419498	3.854768	24.50	0.358	(20.27 8.28)
MOR	0.1 - 0.9	3.056375	3.052106	6.0	0.060	(0.03 1.89)	0.423308	0.439410	8.6	0.134	(2.57 2.93)	0.267971	0.275199	12.8	0.305	(9.03 5.06)
	0.3 - 0.7	3.689980	3.747015	5.8	0.050	(0.00 1.58)	0.548342	0.553402	7.8	0.109	(1.50 2.69)	0.283597	0.290467	13.0	0.293	(8.13 5.15)
	Adaptativo	2.081246	2.046311	5.07	0.044	(0.00 1.38)	0.364971	0.368820	7.30	0.137	(3.07 2.77)	0.293933	0.306642	12.93	0.321	(9.23 5.47)
TRE	0.1 - 0.9	4.887197	5.066371	5.8	0.054	(0.17 1.80)	0.806103	0.792245	8.7	0.104	(0.87 3.17)	0.448043	0.436540	10.4	0.229	(7.20 4.17)
	0.3 - 0.7	4.886871	4.991007	5.7	0.046	(0.00 1.62)	0.991394	0.975439	8.0	0.084	(0.23 2.81)	0.455389	0.437021	10.3	0.211	(6.17 4.10)
	Adaptativo	3.762724	3.686490	5.00	0.043	(0.00 1.49)	0.906458	0.912706	6.73	0.080	(0.17 2.68)	0.449993	0.436562	10.90	0.246	(7.93 4.38)

Observando dichas tablas podemos ver que la solución más precisa del mecanismo de umbrales adaptativos MO-$AD_{I(A)}$ está muy cerca de la solución más precisa obtenida con la propuesta de umbrales fijos MO-AD_I y generalmente, la solución más interpretable (usando el índice propuesto) obtenida por MO-$AD_{I(A)}$ es normalmente más interpretable que la más interpretable de MO-$AD_{I(0.3\text{-}0.7)}$, aunque a veces dicha solución no puede ser usada por ser su precisión demasiado mala.

El hecho anteriormente comentado deja patente que el análisis de los tres puntos más representativos no es suficiente para mostrar la utilidad de la propuesta multiobjetivo adaptativa respecto a la propuesta multiobjetivo con umbrales fijos, por lo que se hace necesario el uso de otras herramientas para estudiar el comportamiento y poder comparar ambos mecanismos (fijo y adaptativo).

Por otro lado, nos podemos hacer una idea de que es lo que ocurre si vemos gráficamente los Paretos obtenidos por cada una de las propuestas.

La Figura 3.7 muestra los frentes de Paretos obtenidos por la propuesta multiobjetivo de umbrales adaptativos MO-$AD_{I(A)}$, y por las propuestas multiobjetivo con umbrales fijos MO-$AD_{I(0.1\text{-}0.9)}$, y MO-$AD_{I(0.3\text{-}0.7)}$, para un ejemplo concreto del problema MOR.

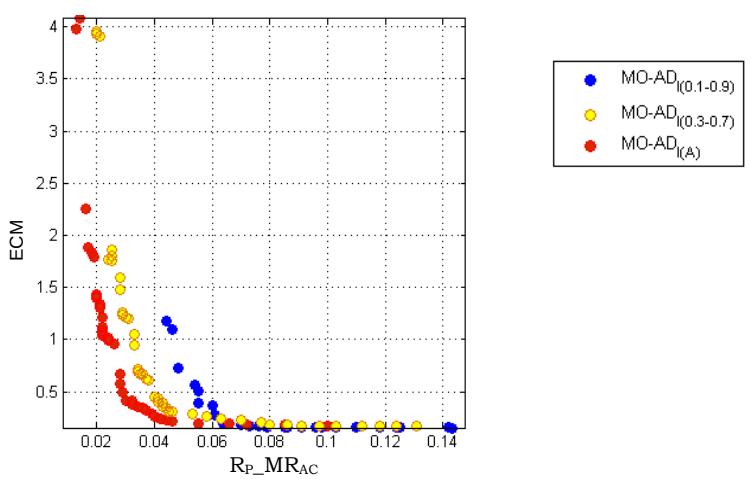


Figura 3.7: Frentes de Paretos y umbrales obtenidos en un ejemplo del problema MOR

Una comparación visual de los frentes de Pareto obtenidos por cada una de las propuestas pone de relieve el interés de la solución obtenida por MO-$AD_{I(A)}$ al estar su curva por debajo de las otras dos en la mayor parte del Pareto, lo que implica que para el mismo nivel de precisión obtiene soluciones más interpretables.

Para comprobar si lo que ocurre en el gráfico anterior es extrapolable o no al resto de ejemplos y conjuntos de datos, usamos el ratio CS [ZDT00], como herramienta para comparar los frentes de Pareto obtenidos por las diferentes propuestas multiobjetivo. Los ratios obtenidos son mostrados en la Tabla 3.27, donde los mejores resultados han sido resaltados en negrita. Observando dicha tabla podemos ver claramente que MO-$AD_{I(A)}$ es mejor que MO-$AD_{I(0.1\text{-}0.9)}$ y MO-$AD_{I(0.3\text{-}0.7)}$, para casi todos los problemas.

Tabla 3.27: Ratios CS obtenidos en el plano precisión-interpretabilidad

	WM				FS-MOGUL			
Conjunto de Datos	Adaptativo vs. 0.1_0.9	0.1_0.9 vs. Adaptativo	Adaptativo vs. 0.3_0.7	0.3_0.7 vs. Adaptativo	Adaptativo vs. 0.1_0.9	0.1_0.9 vs. Adaptativo	Adaptativo vs. 0.3_0.7	0.3_0.7 vs. Adaptativo
PLA	**0.733**	0.205	**0.788**	0.119	**0.825**	0.373	0.363	**0.708**
QUA	0.278	**0.671**	0.416	**0.491**	**0.815**	0.200	0.381	**0.875**
ELE	0.263	**0.554**	0.243	**0.641**	**0.563**	0.321	0.144	**0.815**
AU6	**0.675**	0.445	**0.811**	0.211	**0.605**	0.502	0.299	**0.815**
AU8	**0.683**	0.363	**0.709**	0.447	**0.806**	0.337	**0.803**	0.399
ANA	**0.830**	0.051	**0.966**	0.001	**0.904**	0.008	**0.613**	0.227
ABA	**0.726**	0.157	**0.906**	0.079	**0.822**	0.140	**0.922**	0.112
CON	**0.643**	0.182	**0.554**	0.424	**0.912**	0.063	**0.678**	0.397
STP	**0.470**	0.329	**0.595**	0.420	**0.675**	0.307	**0.845**	0.180
WAN	**0.654**	0.267	**0.531**	0.478	**0.910**	0.149	**0.937**	0.099
WIZ	**0.613**	0.297	**0.591**	0.253	**0.807**	0.198	**0.886**	0.105
MOR	**0.627**	0.187	**0.794**	0.164	**0.792**	0.154	**0.860**	0.125
TRE	**0.439**	0.292	**0.677**	0.297	**0.772**	0.105	**0.802**	0.127

La Figura 3.8 muestra un ejemplo de los umbrales encontrados por el mecanismo de umbrales adaptativos. Probablemente las zonas más interesantes de soluciones son aquellas con el índice de interpretabilidad inferior ($R_{P}\_MR_{AC}$ alrededor de 0.04) la cual tiene uno de los errores más bajos (cercano a 0) y los valores de umbrales están comprendidos entre U_B = 0.25 y U_A = 0.75, y entre U_B = 0.30 y U_A = 0.70.

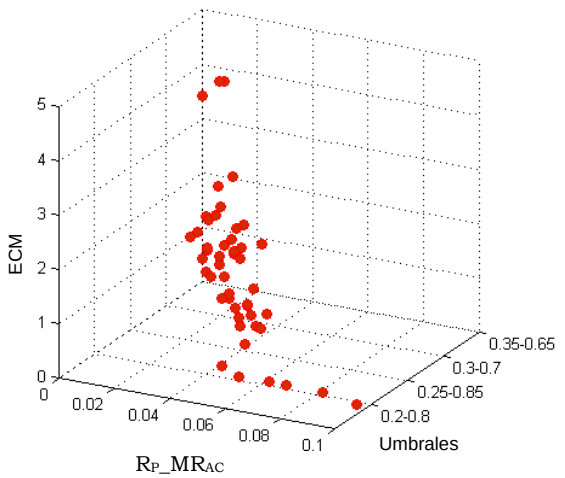


Figura 3.8: Umbrales encontrados por el mecanismo de umbrales adaptativos en un ejemplo del problema MOR

Por todo ello podemos concluir lo siguiente:

- La propuesta multiobjetivo basada en el uso del mecanismo adaptativo para mejorar la interpretabilidad de los SBRDs con defuzzificación adaptativa, MO-AD$_{I(A)}$, permite obtener, en una sola ejecución del algoritmo, un conjunto de soluciones que van desde la más interpretable hasta la más precisa sin necesidad de configurar los umbrales para obtener una u otra (según sea el caso).

- El conjunto de soluciones obtenidas con la propuesta MO-AD$_{I(A)}$, mejora a aquellas obtenidas con la propuesta multiobjetivo de umbrales fijos, convirtiéndose así en la opción de diseño más preferible.

3.5. Sumario

En este capítulo hemos propuesto una forma de mejorar la interpretabilidad de los sistemas con defuzzificación adaptativa mejorando también su precisión, mediante la introducción de un mecanismo para reducir, en función de unos determinados valores umbrales, el número de reglas, el número de reglas con pesos y el número de reglas activadas

simultáneamente con cada ejemplo, proponiendo un índice de interpretabilidad que tiene en cuenta todos estos factores de forma conjunta, propiciando que distintos aspectos relacionados con la interpretabilidad en un sistema difuso con defuzzificación adaptativa queden englobados en un único objetivo.

A partir de dicho mecanismo se ha desarrollado dos modelos evolutivos multiobjetivos para maximizar la precisión e interpretabilidad. Un primer modelo, que aprende los parámetros de la defuzzificación adaptativa mediante unos valores de umbrales fijos proporcionados por un experto, y un segundo modelo, autoadaptativo, en el que los valores de los umbrales son aprendidos en el mismo proceso de aprendizaje de forma conjunta con los parámetros de la defuzzificación.

Para analizar ambos modelos propuestos se ha llevado a cabo un estudio experimental en el que se han utilizado trece conjuntos de datos reales de regresión de diferentes características, con dos BRs iniciales distintas. El estudio se ha dividido en dos partes, cada una de ellas centrada en uno de los modelos.

En la primera, el modelo multiobjetivo de umbrales fijos ha sido comparado con un par de modelos de un solo objetivo (uno que utiliza la defuzzificación adaptativa clásica y otro que la utiliza con el mecanismo para mejorar la interpretabilidad propuesto). Además se ha analizado el comportamiento práctico del modelo estudiando la influencia que tienen distintos valores de umbrales en el equilibrio entre interpretabilidad y precisión.

En la segunda se ha analizado el comportamiento del modelo autoadaptativo con respecto al modelo con umbrales fijos, comparándolo con las dos configuraciones de umbrales con las que se obtiene las soluciones más precisas y más interpretables.

Al igual que en el capítulo anterior, el estudio experimental ha sido validado con test estadísticos no paramétricos, derivándose las siguientes conclusiones:

- Las soluciones más interpretables se obtienen con valores de umbrales altos y las más precisas con valores de umbrales bajos. No obstante, todas son capaces de encontrar la solución más precisa con un nivel similar de precisión, aunque obviamente con un nivel de interpretabilidad diferente.

- Las propuestas multiobjetivos mejoran la precisión y la interpretabilidad de las propuestas de un solo objetivo orientadas a precisión. Además el mecanismo propuesto también es aplicable a los métodos de un solo objetivo consiguiendo mejorar la precisión de éstos cuando usan la configuración de umbral adecuada.

- El modelo autoadaptativo mejora las soluciones obtenidas con el modelo de umbrales fijos, convirtiéndose así en la opción de diseño preferible.

Capítulo 4

Incremento de la Escalabilidad en AGMOs con Inferencia Adaptativa para Problemas de Alta Dimensionalidad

La mejora de la escalabilidad es un problema de actualidad desde hace unos años en buena parte de las áreas de la Inteligencia Computacional, generalmente, en las que tratan con conjuntos de datos: una vez los modelos que se han pretendido construir han mejorado progresivamente a lo largo de los años en cuanto a sus niveles de precisión e interpretabilidad, el nuevo reto aún sin resolver es abordar problemas con mayores conjuntos de datos.

Es cierto por otro lado que el concepto de *conjunto de datos grande* puede tener distinta consideración dependiendo del área particular en la que nos encontremos, e incluso es un concepto que parece evolucionar con el tiempo, pues lo que hace unos años se consideraba un conjunto muy grande, puede que hoy en día *no lo sea tanto* con toda seguridad. Tradicionalmente, los conjuntos de datos que se manejan en modelado difuso son más pequeños que los utilizados típicamente en minería de datos, como la clasificación, etc.

En el presente capítulo de esta tesis abordamos el aumento de volumen de los conjuntos de datos con los que se aprende o ajusta un SDE que utiliza parámetros en algún componente de su MI, concretamente en el operador de conjunción del SI [AFHMP07]. Cuanto mayor es el conjunto de datos, mucho mayor es el número de reglas generadas por el modelo [CA98, Jin00] y consecuentemente mayor es el número de parámetros que tiene que aprender/ajustar el SI adaptativo. Esto nos conduce a espacios de búsqueda más grandes que deben ser afrontados por el algoritmo evolutivo, lo que unido al mayor tamaño del conjunto de datos de entrenamiento, provoca una disminución considerable de su ritmo de convergencia.

Para solucionar este problema en este capítulo desarrollamos un mecanismo que permite recuperar los tiempos de respuesta razonables a pesar del incremento del tamaño de los ejemplos de entrenamiento. Dejamos como trabajo futuro arquitecturas que exploten procesamientos paralelos o distribuidos para afrontar volúmenes de datos aún mayores.

Por último, cabe destacar que si bien a lo largo de esta tesis siempre hemos trabajado en el ámbito del equilibrio entre precisión e interpretabilidad, este objetivo cambia ligeramente cuando los conjuntos de datos son grandes, es decir, un sistema con gran cantidad de ejemplos es un sistema con un número de reglas más elevado, y eso ya es necesariamente un inconveniente para la interpretabilidad. Por tanto aunque el objetivo relacionado con la interpretabilidad es tenido en cuenta, ante este tipo de problemas el sistema se encuentra inevitablemente algo más sesgado a una mejor precisión por la relativa disminución del interés intrínseco por la interpretabilidad.

Para materializar esta propuesta, el capítulo se organiza de la siguiente manera: En primer lugar, en la Sección 4.1 se introduce y justifica el problema a resolver. En la Sección 4.2 se presenta el nuevo operador de conjunción adaptativo y se describe en detalle el mecanismo para acelerar los cálculos de la inferencia que puede ser implementado gracias a este operador. A continuación en la Sección 4.3 se presenta nuestra propuesta, describiendo en detalle el modelo de aprendizaje evolutivo asociado, sus principales características y operadores genéticos considerados. La Sección 4.4 desarrolla el estudio experimental del método propuesto y muestra los resultados obtenidos, para finalmente en la Sección 4.5, ofrecer algunas conclusiones y observaciones finales.

4.1. Introducción

El conjunto de parámetros de los operadores de conjunción adaptativos de los sistemas difusos son generalmente aprendidos usando un algoritmo evolutivo con un esquema de codificación real. Cuando el conjunto de datos contiene muchas instancias o ejemplos y/o muchas variables o características el número de reglas generadas por el modelo suele crecer de forma exponencial [CA98, Jin00] provocando un aumento considerable del número de parámetros que debe aprender el modelo. Como consecuencia de todo ello, la convergencia del algoritmo evolutivo se ralentiza de forma considerable debido [FAN+13, Her08] a:

1. La complejidad del espacio de búsqueda.

2. La escalabilidad, originado sobre todo por el gran número de instancias implicadas en estos conjuntos de datos que deben ser evaluadas.

En la mayoría de los casos, las diferentes técnicas existentes para realizar el aprendizaje o mejora de la precisión e interpretabilidad en el sentido de la complejidad de los SBRDs lingüísticos no se pueden utilizar al no ser eficaces y generar modelos muy complejos que necesitan tiempos de ejecución extremadamente grandes y/o gran cantidad de memoria.

Por todo ello, uno de los retos más importantes hoy en día, en el ámbito del uso de AGMOs con MI adaptativo y en el área del modelado difuso lingüístico en general, es el diseño de sistemas para problemas de regresión con conjuntos de datos de alta dimensionalidad y/o gran escalabilidad. Debemos enfatizar que, en el contexto de este tipo de problemas, el uso de AGMOs van encaminados a obtener modelos con diferentes equilibrios entre precisión y complejidad, debido a que al tratar con un gran número de reglas, la interpretabilidad es considerada en segundo plano.

En este tipo de problemas se hace necesario la implementación de algún tipo de mecanismo que permita, bien reducir el número de cálculos que hay que realizar o bien acelerar lo máximo posible el tiempo necesario para realizarlos.

Algunos autores abordan este problema proponiendo tres tipos de soluciones: la primera es la reducción de datos [ADM12, AGH11, SGGT12], utilizando un conjunto reducido de instancias representativas en lugar del

conjunto completo de datos de entrenamiento con el fin de reducir el esfuerzo computacional, o mediante el aprendizaje conjunto de las variables utilizadas, el particionamiento lingüístico y el conjunto de reglas, a fin de reducir el número y la complejidad de las reglas obtenidas. La segunda es el uso de propuestas de aproximaciones de la función de aptitud con el fin de reducir el tiempo de ejecución mediante el cálculo aproximado de dicha función en lugar del cálculo completo de ésta [CLM10, CLM11, Jin05] y la última es la aplicación de metodologías paralelas o distribuidas [BdVMP09, dVBMP09, RABH09]. Además, recientemente otros autores han propuesto el uso de métodos híbridos que combinan la aplicación de metodologías paralelas junto con la reducción de datos [IMN13].

En el presente capítulo, el problema planteado lo abordamos proponiendo un nuevo operador de conjunción adaptativo discreto (en lugar de continuo), más eficiente y rápido, que pretende reducir lo máximo posible tanto el espacio de búsqueda como los tiempos necesarios para realizar los cálculos de la inferencia, el cual es usado en un modelo evolutivo multiobjetivo centrado exclusivamente en la parametrización de los operadores de conjunción del SI.

La razón de usar un SI adaptativo basado exclusivamente en la parametrización del operador de conjunción es explicada en la Sección D.1.4.1 del Apéndice D, donde se habla de la escasa influencia en el proceso de inferencia del operador de implicación adaptativo frente a la del operador de conjunción adaptativo, así como de las ventajas de usar para dicha parametrización parámetros individuales para cada regla, los cuales son aprendidos a la vez que se aprende la BR, en lugar de utilizar un único parámetro para adaptar globalmente el comportamiento del SI, ya que de esta forma el operador de conjunción puede adaptarse individualmente a cada regla, al tener cada una de ellas un parámetro individual asociado.

La ventaja de esta técnica es que puede ser complementaria a cualquiera de las otras técnicas mencionadas anteriormente. De esta forma, la conjunción adaptativa, y por tanto la propuesta a desarrollar en este capítulo podría ser usada en combinación con las metodologías indicadas anteriormente para acelerar los cálculos y mejorar la precisión e interpretabilidad en el sentido de la complejidad de los sistemas con conjuntos de datos con alta dimensionalidad y/o gran escalabilidad, lo que lo hace aún más interesante.

4.2. OCA_{PAD}: Una Propuesta para un Operador de Conjunción Adaptativo para Problemas de Alta Dimensionalidad

Una vez planteado el problema de la complejidad y el escalado en los sistemas con inferencia adaptativa en conjuntos de datos de alta dimensionalidad y los objetivos que se pretenden optimizar en el nuevo modelo, introduciremos en esta sección una parte esencial del mismo como es el operador de conjunción adaptativo utilizado, antes de exponer el nuevo modelo propuesto en la siguiente sección.

Esta sección por tanto, proporciona una descripción detallada de dicho operador que denotamos como OCA_{PAD} (operador de conjunción adaptativo para problemas de alta dimensionalidad), así como los detalles de cómo con este operador se puede acelerar los cálculos cuando usamos un sistema difuso lingüístico con una BC que no se modifica en todo el proceso.

Tal y como se ha indicado en las secciones anteriores, los inconvenientes derivados de la utilización de operadores de conjunción adaptativos (sobre todo cuando utilizan un parámetro para cada regla) en problemas de alta dimensionalidad son consecuencia de la gran cantidad de reglas derivadas por el elevado número de variables y datos que tienen estos problemas.

Debemos tener en cuenta que los operadores clásicos de conjunción adaptativa (véase Tabla D.2 del Apéndice D) utilizan parámetros continuos en sus expresiones y los rangos de valores de dichos parámetros pueden ser muy extenso (sobre todo en las t_normas adaptativas de Dombi y Frank) lo que unido a la complejidad de las expresiones necesarias para calcular las citadas t-normas hace que el cálculo de la función de aptitud (fitness) sea muy costosa y lenta, ralentizando el proceso de búsqueda. Todo ello trae como consecuencia que los algoritmos de optimización empleados para aprender los valores de dichos parámetros tengan un doble problema debido a la amplitud del espacio de búsqueda: primero, necesitan más tiempo para alcanzar las soluciones y segundo, la calidad de las soluciones encontradas es menor. Además a todo ello se une el hecho de que el cálculo de las citadas t-normas es costoso en tiempo, lo que empeora aún más la situación.

Para solventar este doble problema, en este capítulo se propone utilizar el operador OCA_{PAD}, una nueva t-norma de conjunción adaptativa que usa valores discretos para sus parámetros. El operador propuesto permite

seleccionar entre distintas t-normas clásicas, en función del valor que tomen los parámetros discretos de dicha t-norma, de modo que las reglas que tengan asociado el mismo valor en dicho parámetro, utilizarán la misma t-norma para su adaptación.

La ventaja principal de nuestra propuesta radica en que el algoritmo utilizado para la optimización de los parámetros, sólo debe buscar para cada regla entre diferentes posibilidades discretas limitadas en lugar de entre valores continuos, como ocurre al usar t-normas adaptativas clásicas, reduciendo el espacio de búsqueda de forma considerable. Por otro lado, al utilizar siempre las mismas t_normas permite precalcular los valores de éstas, lo que permite su almacenamiento en una tabla de consulta rápida para su posterior reutilización con el consecuente ahorro de tiempo al no tener que calcularlas una y otra vez.

En las siguientes subsecciones se explicará más detalladamente la nueva t_norma y el mecanimo para acelerar su cálculo.

4.2.1. El Selector de T-normas

El mecanismo propuesto, consistente en seleccionar la t-norma clásica más apropiada para cada regla es menos flexible y tiene menos grados de libertad que las existentes basadas en el cálculo de los valores continuos de las t-normas adaptativas. Esta carencia queda compensada con la reducción tan enorme del espacio de búsqueda que se consigue al tener que seleccionar entre unos pocos valores discretos en lugar de entre valores continuos, que hace que la convergencia alcanzada por el algoritmo evolutivo sea mayor, permitiendo obtener resultados de mayor calidad.

Para formar parte del operador de conjunción adaptativo propuesto se han seleccionado varias t-normas clásicas combinadas con algunas t-normas calculadas a partir de valores específicos obtenidos a partir de una t-norma adaptativa. Concretamente se han seleccionado las siguientes:

- Como t-normas clásicas, han sido seleccionadas (véase Tabla D.1 del Apéndice D) las t-normas del Mínimo, Hamacher y Producto Algebraico, ya que son las más ampliamente empleadas como operadores de conjunción debido a su simplicidad computacional y a su buen comportamiento práctico [CHP00].

- Como base para la creación del resto de t-normas específicas, se ha seleccionado la t-norma adaptativa de Dubois (véase Expresión D.9 del Apéndice D), teniendo en cuenta que se calcula de manera más eficiente y muestra mayor precisión en comparación con las otras t-normas de Dombi y Frank (véase Expresión D.10 y D.11). Considerando la relación entre las t-normas clásicas y los valores de los parámetros de las t-normas adaptativas (véase Tabla D.3 del Apéndice D), se han creado ocho t-normas, a partir de la t-norma de Dubois con α = 0.1, 0.2, 0.3, 0.4, 0.6, 0.7, 0.8, 0.9, de modo que todas se encontrarán entre el Mínimo y el Producto Algebraico (aquellas que tienen mejor rendimiento y funcionamiento).

De esta forma, el nuevo operador de conjunción adaptativo permite seleccionar entre un total de once t-normas distintas:

$$TOCA_{PAD}(\alpha) \in [T_{Min}, Dub_{0.1}, Dub_{0.2}, Dub_{0.3}, Dub_{0.4}, T_{Ham}, Dub_{0.6}, Dub_{0.7}, Dub_{0.8}, Dub_{0.9}, T_{Alg}]$$

Las ventajas de usar esta nueva t-norma adaptativa es múltiple:

- Por un lado, se reduce el espacio de búsqueda, lo que garantiza una convergencia rápida. El algoritmo de aprendizaje sólo debe aprender una de estas once posibles t-normas para cada regla, lo que supone menos flexibilidad que una t-norma continua cuyos valores se encuentren en el mismo intervalo como es el caso de la de Dubois, pero que se compensa con la reducción del espacio de búsqueda, que producirá mejores resultados con un menor esfuerzo computacional cuando el número de reglas sea muy alto.

- Por otro, al restringir el intervalo de variación entre las mejores t-normas se evita una sobreadaptación de las reglas, lo que permite un aprendizaje general y evita que se produzca sobreaprendizaje. Además, permite una fácil preselección de t-normas prometedores, interesantes para combinar con otras técnicas de aprendizaje.

- Por último, esta t-norma permite acelerar de forma drástica los tiempos empleados en los cálculos, al permitir precalcular y almacenar en una tabla de consulta rápida los grados de emparejamiento correspondientes a las once t-normas posibles (las t-normas adaptativas clásicas al trabajar con valores continuos poseen infinitos valores, lo que hace imposible su precálculo).

Debido a la importancia de este último punto, pasamos a continuación a describir de forma detallada el mecanismo para acortar el tiempo empleado y acelerar dichos cálculos.

4.2.2. Implementación del Mecanismo para Evaluar el Operador de Conjunción Adaptativo Propuesto

Como comentamos en los capítulos anteriores, el proceso de evaluación en un SBRD lingüístico con fuzzificación puntual se basa en calcular la salida del SI (véase Expresión D.8 del Apéndice D) y posteriormente aplicar el método de defuzzificación. Para realizar dicha inferencia, el SI debe primero calcular el grado de emparejamiento (h_i) de cada regla, empleando el operador de conjunción $C(.)$ sobre los puntos de corte $\mu_{Ai}(x_i)$ resultantes de la fuzzificación puntual de cada regla, $h_i = C(\mu_{A1}(x_1), \ldots, \mu_{Am}(x_m))$, y posteriormente inferir aplicando el operador de implicación $I(\cdot)$, utilizando el grado de emparejamiento calculado para cada regla y la parte consecuente de dicha regla.

De esta forma, durante la fase de aprendizaje evolutivo de los parámetros de los operadores de conjunción adaptativos, para cada evaluación de los cromosomas se necesita calcular el grado de emparejamiento de cada regla. El tiempo requerido para esta evaluación viene determinado, entre otros factores, por el número de variables y el número de reglas de la BR utilizada. En problemas de alta dimensionalidad, el SI adaptativo clásico puede necesitar una enorme cantidad de tiempo para obtener o derivar miles de reglas, ya que dicho cálculo hay que hacerlo de forma repetitiva cada vez que aplicamos cada regla a cada ejemplo del conjunto de entrenamiento, de forma que cuantas más variables tenga las reglas, más reglas haya y más ejemplos tenga el conjunto de datos del problema, más tiempo será el necesario para realizar dichos cálculos. Piénsese además que en el proceso de aprendizaje de un determinado modelo evolutivo estos cálculos hay que hacerlos para cada iteración del AE y para cada SBRD codificado en los cromosomas de dicho algoritmo con lo que el número de veces que hay que realizarlos es enorme.

Por tanto, en problemas de alta dimensionalidad (y también en los de gran escalado) se hace imperiosamente necesario la implementación de algún tipo de mecanismo que permita o bien reducir el número de veces que dichos cálculos deben hacerse o bien acelerar lo máximo posible el tiempo necesario para realizarlos. Algunas de las técnicas usadas en la literatura científica tratan de reducir el número de ejemplos de datos sobre los que se aplica los cálculos preseleccionando un subconjunto de datos que represente fielmente el

comportamiento del conjunto de datos completo, mientras que otras tratan de reducir el número de reglas del sistema (cuanto menor sea el número de ejemplos y/o menor sea el número de reglas menos tiempo se tardará en inferir y calcular la salida global del sistema). Independientemente de la técnica utilizada, también es conveniente reducir dicho tiempo acelerando de alguna forma el tiempo necesario para realizar los cálculos. En el caso concreto que nos ocupa, el operador de conjunción adaptativo propuesto en nuestro modelo permite acelerar de forma drástica el tiempo invertido en ello, mediante el precálculo y almacenamiento de parte de los cálculos necesarios para realizar la inferencia en una tabla de acceso directo de consulta rápida. La ventaja de nuestro mecanismo es que puede combinarse con otras técnicas, como las mencionadas anteriormente ya que son complementarias a éstas, lo cual lo hace idóneo para ser utilizado en este tipo de problemas.

El mecanismo propuesto se basa en la siguiente idea: precalcular y almacenar directamente en una tabla de consulta rápida, para cada ejemplo y cada regla posible, los valores de los *grados de emparejamiento* resultantes de evaluar el operador de conjunción $C(\mu_{A1}(x_1), \ldots, \mu_{Am}(x_m))$ con las once t-normas posibles que permite nuestro operador de conjunción adaptativo.

De esta forma, podemos ahorrarnos tener que realizar el cálculo de la fuzzificación puntual $\mu_{Ai}(x_i)$ de cada regla con cada ejemplo del conjunto de entrenamiento (cuantas más variables antecedentes A_i tenga la regla más tiempo se tarda en realizar el cálculo) y tener que realizar el cálculo de la conjunción cada vez que inferimos.

Debemos tener en cuenta que al basarse la nueva t-norma adaptativa propuesta en tres t-normas no adaptativas (Mínimo, Hamacher y Producto Algebraico) y en otras ocho adaptativas con valores fijos, podemos precalcular tanto unas como otras (las primeras por no tener parámetros asociados y las segundas por conocerse de antemano los valores fijos de dichos parámetros).

Dicho cálculo se realiza una única vez al principio del algoritmo, para cada ejemplo del conjunto de datos de entrenamiento y para cada regla candidata de la BR inicial, almacenándose en la tabla de consulta rápida únicamente los once *grados de emparejamiento* posibles, que es lo que necesitamos para aplicar la inferencia. A la hora de inferir, en función del valor del parámetro del operador de conjunción adaptativo asociado a una determinada regla, seleccionamos directamente de dicha tabla el *grado de emparejamiento* correspondiente a dicha regla (resultante de aplicar la t-norma clásica determinada por el parámetro) ahorrando tiempo por partida doble al no tener que calcular para cada regla ni la fuzzificación puntual ni la conjunción. En

función del número de reglas activas que tenga el modelo simulado, sólo tendríamos que aplicar el operador de implicación sobre los grados de emparejamiento de dichas reglas y defuzzificar empleando el correspondiente operador de defuzzificación.

Los requerimientos de memoria para llevar a cabo esta idea tal cual son enormes, ya que sería necesario crear una tabla tridimensional de un tamaño igual al producto *ejemplos* x *reglas* x 11 *t-normas*. A modo de ejemplo en un problema como el Elevators [AFSG+09] con 16599 ejemplos y 4322 reglas serian necesarios unos 4.7 Gb de espacio de almacenamiento (considerando un modelo de validación cruzada de 5 particiones y datos de tipo double para almacenar los precálculos).

Este problema se resuelve almacenando en memoria sólo las t-normas de aquellas reglas que se disparan para cada ejemplo, creando una tabla bidimensional (*ejemplos* x *reglas*), en el que cada elemento *e (ejemplo)*, *r (regla)* de la tabla puede tener o no (dependiendo de si la regla *r* correspondiente se dispara o no para el ejemplo *e*) una estructura con los correspondientes grados de emparejamiento para cada una de las once t-normas posibles. De esta forma, si un ejemplo no se dispara para una regla concreta la estructura no es reservada y los cálculos del grado de emparejamiento no son realizados, con el consiguiente ahorro de tiempo y, sobre todo, de memoria. El mecanismo propuesto se lleva a cabo por medio de los siguientes pasos:

1. Realizar inicialmente un procedimiento de pre-procesamiento, donde para cada ejemplo e_i en *E (conjunto de datos de entrenamiento)* y para cada regla R_j de la BR inicial:

   1.1. Calcular la fuzzificación puntual de cada regla $\mu_{A1}(x_1), \ldots, \mu_{Am}(x_m)$

   1.2. Calcular, para aquellas reglas que se disparan, los once *grados de emparejamiento* posibles de la parte antecedente de la regla (uno por cada operador de conjunción discreto posible)

   1.3. Reservar memoria y guardar en la tabla de consulta rápida los once *grados de emparejamiento* de las reglas que se disparan.

2. En el proceso de evaluación la propuesta sólo necesita buscar en la tabla de consulta rápida los valores correspondientes, en función del parámetro del operador de conjunción adaptativo asociado a cada regla, que indica cuál de las once posibles t-normas es la utilizada. Así, para cada ejemplo, indexamos el operador de conjunción de cada regla activa para obtener directamente de la tabla su correspondiente *grado de*

emparejamiento (h_i). Si para ese ejemplo concreto, la regla no se dispara, no tendrá ningún *grado de emparejamiento* asociado y no habrá que hacer ningún cálculo adicional. En aquellas reglas que se disparan se selecciona su correspondiente grado de emparejamiento y se calcula su *centro de gravedad (CG_i)*, al utilizar como método de defuzzificación el modo B-FITA.

3. Finalmente, la salida del sistema es calculada, aplicando el método de defuzzificación sobre los valores h_i y CG_i anteriores de las reglas que se disparan.

Analizando los pasos del mecanismo, se deduce claramente que, aquellos conjuntos de datos que contienen muchas reglas disparadas por cada ejemplo necesitan más tiempo para realizar sus cálculos que aquellas otros en las que el número de reglas disparadas es menor, ya que el cálculo del centro de gravedad sólo es necesario hacerlo en aquellas reglas que se disparan. Como consecuencia de ello, el tiempo empleado en el aprendizaje del algoritmo no depende tanto del número de ejemplos y reglas, sino más bien del número de ejemplos y número de reglas disparadas por cada ejemplo.

4.3. Un Sistema de Inferencia Adaptativo Multiobjetivo Rápido y Escalable

Esta sección describe el modelo evolutivo propuesto para el SI adaptativo que utiliza el operador de conjunción OCA_{PAD} presentado en la sección anterior, que al permitir acelerar los cálculos enormemente, es ideal para ser utilizado en problemas de alta dimensionalidad. En este sentido, aunque diferentes técnicas de optimización podría ser consideradas para el aprendizaje de los parámetros del SI adaptativo, en este trabajo como en los anteriores, hemos considerado utilizar un AGMO para esta tarea, por permitir mejorar tanto la precisión como la simplicidad del sistema (esencial para manejar problemas de alta dimensionalidad).

El esquema de aprendizaje considerado para obtener los parámetros óptimos con un AGMO efectivo está basado en dos aspectos principales: el aprendizaje evolutivo del operador de conjunción y la implementación eficiente del modelo de aprendizaje.

4.3.1. Convergencia y Escalabilidad del Modelo Propuesto

El uso del operador de conjunción adaptativo propuesto, OCA_{PAD}, permite una implementación de software eficaz y rápida que permite obtener el modelo más rápidamente solventando dos de los problemas más importantes que presentan los conjuntos de datos de alta dimensionalidad:

- El gran número de evaluaciones necesarias para alcanzar la convergencia. Este problema se resuelve mediante el aprendizaje de sólo once operadores discretos para cada regla en lugar del dominio continuo de las t-normas adaptativas clásica (reducción del espacio de búsqueda).

- La gran cantidad de tiempo requerida para evaluar la salida del SBRD. Este problema se resuelve precalculando y almacenando en una tabla de consulta rápida los grados de emparejamientos posibles, con el consiguiente ahorro de tiempo, al no tener que calcular tampoco la fuzzificación de cada regla en cada evaluación.

Teniendo en cuenta la discusión o análisis anterior, el algoritmo propuesto se compone de tres componentes principales:

- El operador adaptativo propuesto.

- Un AGMO eficaz basado en NSGA-II [DAPM02], con dos objetivos a minimizar (el error del sistema y el número de reglas), a fin de aprender los parámetros prometedores.

- Una implementación eficiente y rápida del SI adaptativo con este operador de conjunción adaptativo propuesto.

4.3.2. Algoritmo y Modelo Evolutivo Multiobjetivo Propuesto

Una vez explicada las ventajas del nuevo operador de conjunción adaptativo propuesto, en cuanto a reducción del espacio de búsqueda y del tiempo necesario para realizar los cálculos, pasamos a continuación a describir en detalle los componentes del modelo evolutivo multiobjetivo que hace uso de dicho operador, aprendiendo los parámetros óptimos de éste junto con la selección de reglas.

En este sentido, el modelo propuesto se basa en un AGMO con dos objetivos a minimizar: el número de reglas (#*R*), para mejorar la interpretabilidad en el sentido de la complejidad, y el error del sistema, a fin de mejorar la precisión. En este trabajo, nuestro modelo utiliza un AGMO basado en el conocido NSGA-II [DAPM02] con las mejoras descritas en el capítulo anterior (véase Sección 3.3.1.3 del Capítulo 3) que permitían mejorar la capacidad de búsqueda del algoritmo, haciendo que se centrara en la zona más reducida e interesante del Pareto donde las soluciones alcanzan una precisión más alta con el menor número de reglas posibles, es decir, aquella donde las soluciones son más precisas y más interpretables.

En las siguientes subsecciones, planteamos los principales componentes y parámetros de este algoritmo. Para ello, en primer lugar, se describirá la codificación de los cromosomas de la población, a continuación se indicará cómo se genera la población inicial y por último se indicarán los operadores genéticos de cruce y mutación utilizados.

4.3.2.1. Esquema de Codificación y Población Inicial

Para representar los parámetros del modelo evolutivo propuesto se utiliza un esquema de codificación entero de tamaño N (donde N representa el número de reglas de la BR inicial de la que se aprende) en el que se codifica el operador de conjunción de las diferentes reglas,

$$C = (I_1, \ldots, I_N) \mid I_i \in \{0, 11\} \qquad (4.1)$$

Cada gen I_i representa el parámetro utilizado por la regla i-ésima y toma valores enteros en el intervalo [0, 11]. Los valores de 1 a 11 codifican las once t-normas diferentes que es posible utilizar, mientras que el valor 0 indica que la regla correspondiente no es usada.

La población inicial se compone de dos subconjuntos diferentes de individuos: Un individuo de la población inicial tiene todos los genes inicializados al valor 1 a fin de iniciar el proceso evolutivo con todas las reglas activas con la t-norma del mínimo como operador de conjunción. Los restantes individuos de la población inicial se inicializan aleatoriamente, de forma que a cada gen se le asigna una de las posibles t-normas, es decir, sin ningún 0, a fin de no descartar a priori ninguna de las reglas, al igual que el primer individuo de la población (todos los individuos inicialmente tienen todas las reglas de la BR inicial).

4.3.2.2. Objetivos

En este algoritmo se utiliza estos objetivos a minimizar:

- El número de reglas, siendo (#*R*) el número de reglas finales en el sistema, como una medida de la complejidad (simplicidad).
- El ECM(*S*) (véase Expresión 2.7 del Capítulo 2), que mide la precisión del sistema.

donde *S* denota el modelo difuso cuya SI utiliza el operador de conjunción adaptativo OCA_{PAD}, la t-norma del mínimo como operador de inferencia, y el centro de gravedad ponderado por el grado de emparejamiento como método de defuzzificación.

4.3.2.3. Operadores de Cruce y Mutación

En nuestro modelo utilizamos tanto operadores genéticos de cruce como de mutación para obtener la población descendiente a partir de la población de los padres. El operador de cruce es aplicado a todos los individuos que forman la población de padres, mientras que el operador de mutación es aplicado con una determinada probabilidad P_m asignada como entrada del algoritmo.

El operador de cruce utilizado es HUX [Esh91]. Este operador intercambia exactamente la mitad de los genes que son diferentes en los padres, asegurando la distancia máxima entre los hijos y sus padres (exploración). Todos los individuos que forman la población de padres son cruzados de dos en dos, generándose por cada cruce dos descendientes, que sustituyen a sus padres. Para cada uno de ellos, el operador de mutación cambia aleatoriamente el valor de un gen con probabilidad P_m, asignándole otro valor entero distinto en el intervalo [0, 11].

4.4. Estudio Experimental y Análisis de los Resultados

Con el fin de evaluar la utilidad y eficacia de la metodología propuesta, diseñada para realizar un aprendizaje rápido en problemas de alta dimensionalidad, se han utilizado 17 problemas de regresión de complejidades diferentes, cubriendo un rango de problemas que van desde 6 a 85 variables de entrada (DEL y TIC, respectivamente) y desde 337 a 20640 ejemplos (BAS y CAL).

La Tabla 4.1 resume las principales características de los diferentes problemas considerados en este estudio y muestra los vínculos a las páginas web del proyecto KEEL [AFSG+09], Torgo y repositorio UCI de donde se pueden descargar. Para cada problema se muestra su nombre, el número de ejemplos que componen el conjunto de datos asociado, el número de variables que tiene, así como el repositorio de donde se puede descargar. Los problemas están ordenados de menor a mayor complejidad. Todos los problemas son complejos de por sí en términos de tarea de modelado, bien porque poseen muchas variables, bien porque tienen muchos datos o por ambas cosas. No obstante, los más complejos de todos son ELV, CA, AIL y TIC (parte inferior de la tabla). Que nosotros sepamos, estos problemas nunca se han resuelto utilizando SI adaptativos difusos, debido a la gran cantidad de tiempo necesario para evaluar cada individuo (instancia del problema) y al número mínimo de evaluaciones necesarias para alcanzar la convergencia de los SDEs. Además, el gran número de reglas que se obtiene para este tipo de problemas con los algoritmos ad-hoc de aprendizaje basado en datos como WM, hacen más difícil el diseño de sus modelos respectivos. Por todo ello, estos problemas representan un reto importante para este tipo de algoritmos.

Tabla 4.1: Conjunto de datos considerados para el estudio experimental

Conjunto de Datos	Nombre	Instancias	Variables	Repositorio
Delta_elv	DEL	9517	6	Keel
Abalone	ABA	4177	8	Keel
California	CAL	20640	8	Keel
Concrete	CON	1030	8	Keel
Kinematics Robot	KIN	8192	8	Torgo
Puma8nh	PUM	8192	8	Torgo
Stock	STP	950	9	Keel
Weather Ankara	WAN	1609	9	Keel
Wine-red	WINR	1599	11	UCI
Wine-white	WINW	4898	11	UCI
Forest Fires	FOR	517	12	Keel
Mortgage	MOR	1049	15	Keel
Baseball	BAS	337	16	Keel
Elevators	ELV	16599	18	Keel
Computer-Activity	CA	8192	21	Keel
Airelons	AIL	13750	40	Keel
The Insurance Company	TIC	9822	85	Keel

Keel: http://sci2s.ugr.es/keel/datasets.php
Torgo: http://www.liaad.up.pt/~ltorgo/Regression/DataSets.html
UCI: http://www.ics.uci.edu/~mlearn

Antes de mostrar y analizar los resultados obtenidos en el estudio experimental vamos a describir como han sido configurados los conjuntos de datos y la metodología de comparación utilizada en dicho estudio.

4.4.1. Configuración del Estudio Experimental

En esta sección se describe el montaje experimental, incluyendo una breve descripción de los modelos y los test estadísticos no paramétricos consideradas para realizar las comparaciones.

Para analizar el comportamiento práctico de la metodología propuesta, se ha llevado a cabo un estudio experimental en el que el modelo propuesto, se ha comparado contra varios modelos multiobjetivo. Para ello, tres conocidos operadores clásicos de conjunción adaptativa, cuyos SI adaptativos han sido estudiados en [MPH07], han sido considerados para la comparación: los operadores de conjunción adaptativa que utilizan la t-norma adaptativa de Dubois, Dombi y Frank [Miz89]. Denotamos como OCA_{DUB}, OCA_{DOM} y OCA_{FRA} los modelos multiobjetivos respectivos basados en dichas t-normas. La razón por la que han sido elegidos estos operadores estriba en el hecho de ser los más conocidos, eficientes y más utilizados. De hecho, en el Capítulo 2 presentamos un AGMO para mejorar la cooperación entre la BR y los operadores de conjunción donde se utilizó la t-norma adaptativa de Dubois por tener ésta el mejor comportamiento computacional.

El modelos propuesto, al trabajar con valores discretos, tiene menos grados de libertad que los otros tres contra los que nos comparamos, que usan los operadores clásicos de conjunción adaptativa, que trabajan con valores continuos, por lo que teóricamente estos últimos deben obtener los resultados más precisos. Por tanto, la precisión alcanzada en ellos representa una buena referencia de dicho objetivo, para nuestro algoritmo propuesto, sobre todo cuando lo aplicamos a problemas que no son de alta dimensionalidad (particularmente en el caso de OCA_{DUB}, que es el más eficiente de los tres computacionalmente y en rendimiento).

Para competir con el modelo propuesto en un enfoque multiobjetivo, los tres modelos OCA_{DUB}, OCA_{DOM} y OCA_{FRA} utilizan un mecanismo de selección de reglas, es decir, un cromosoma con un doble esquema de codificación (C_C + C_S) donde:

- C_C codifica los α_i parámetros del operador de conjunción adaptativo, mediante N parámetros (genes) con codificación real, uno para cada regla R_i de la BR lingüística inicial. El universo de discurso de estos parámetros depende de la t-norma de conjunción adaptativa correspondiente que estemos utilizando: [0, 1] para la t-norma de Dubois, [0, 10] para la de Dombi y [0, 100] para la t-norma de Frank. Estos intervalos han sido seleccionados tras la realización de varias pruebas. Téngase en cuenta que excepto Dubois, cuyo único intervalo posible es el indicado, las t-normas de Dombi y Frank pueden tomar valores entre 0 y ∞ (véase Tabla D.2 y D.3 del Apéndice D).

- C_S codifica la selección de reglas, mediante una cadena binaria de N genes, cada uno representado a cada regla candidata R_i de la BR lingüística inicial. Cada gen podrá tomar los valores 1 ó 0, dependiendo de si la regla correspondiente es seleccionada o no.

La población inicial se inicializa aleatoriamente en la parte correspondiente a los ODAs (C_C) con la excepción de un único cromosoma, cuyos N genes son inicializados de la siguiente manera:

- Cuando usamos la t-norma de Dubois, se inicializan a 0 con el fin de hacer esta t-norma equivalente a la t-norma del mínimo.

- Cuando usamos la t-norma de Dombi, se inicializan a 1 con el fin de hacerla equivalente a la t-norma del mínimo.

- Cuando usamos la t-norma de Frank, se inicializan a 0.5 con el fin de hacerla equivalente a la de Hamacher.

En cuanto a la parte correspondiente a la selección de reglas (C_S), los N genes de todos los individuos de la población inicial, sin excepción, son inicializados a 1, a fin de no descartar a priori ninguna de las reglas (todos los individuos inicialmente tienen todas las reglas de la BR inicial).

El operador de cruce empleado depende de la parte del cromosoma donde sea aplicado. En la parte real C_C, que codifica los parámetros de los operadores difusos se utiliza el cruce BLX-0.5 [Esh91, HLS03] mientras que en la parte binaria C_S, que codifica la selección de reglas se utiliza el cruce HUX [Esh91].

Todos los individuos que forman la población de padres son cruzados de dos en dos, generándose por cada cruce cuatro descendientes como resultado de combinar los dos cromosomas de la parte C_S con los dos de la parte C_C (los

dos mejores sustituyen a sus padres). Respecto al operador de mutación, éste cambia aleatoriamente el valor de un gen en la parte C_S y en la C_C con probabilidad P_m.

Para todos los modelos, las particiones lingüísticas consideradas están compuestas por cinco términos lingüísticos con forma triangular. Además, para obtener el conjunto inicial de reglas lingüísticas candidatas (BR inicial) que tomamos como referencia, se han utilizado los mismos algoritmos (WM y FS-MOGUL) utilizados en el Capítulo 3.

No se ha utilizado ningún otro método de los propuestos en la literatura científica, debido a los problemas que presentan cuando se ejecutan con un número elevado de variables como las que tienen los conjuntos de datos utilizados en este trabajo (véase la parte inferior de la Tabla 4.1).

La Tabla 4.2 resume las principales características de los modelos estudiados.

Tabla 4.2: Modelos considerados en el estudio experimental

Referencia	Método	Tipo de aprendizaje
[WM92]	WM	Generación de reglas ad-hoc guiada por datos con 5 etiquetas
[CH97]	FS-MOGUL	Generación de reglas ad-hoc guiada por datos con 5 etiquetas
[AFHMP07]	OCA_{DUB}	Conjunción adaptativa con t-norma adaptativa de Dubois
[AFHMP07]	OCA_{DOM}	Conjunción adaptativa con t-norma adaptativa de Dombi
[AFHMP07]	OCA_{FRA}	Conjunción adaptativa con t-norma adaptativa de Frank
	OCA_{PAD}	Nuestra propuesta

Los resultados medios de los SBRDs de referencia iniciales obtenidos con WM y FS-MOGUL se muestran en las Tablas 4.3 y 4.4, donde ECM_{ENT} y ECM_{TST} representan el ECM obtenido en entrenamiento y test respectivamente y $\#R$ representa el número medio de reglas.

Al igual que en la Tabla 4.1, los problemas están ordenados de forma ascendente por el número de variables que posee (dimensionalidad). Estos métodos obtienen la BR inicial sobre la que trabajan nuestros modelos propuestos.

Tabla 4.3: Resultado inicial usando la BR generada por WM. Los resultados de $ECM_{ENT/TST}$ de la tabla se deben multiplicar respectivamente por 10^{-6}, 10^{8}, 10^{-4}, 10^{2}, 10^{4}, 10^{-6}, 10^{-8} y 10^{-4} en los problemas DEL, CAL, KIN, FOR, BAS, ELV, AIL y TIC

Nombre	**#R**	**ECM_{ENT}**	**ECM_{TST}**
DEL	679.8	1.624	1.684
ABA	199.0	3.341	3.477
CAL	623.8	38.323	38.712
CON	309.8	35.388	47.644
KIN	6423.2	47.688	120.523
PUM	6435.6	3.281	9.256
STP	265.4	1.453	1.487
WAN	456.8	4.878	6.128
WINR	714.2	0.229	0.251
WINW	1019.8	0.296	0.306
FOR	374.6	14.350	342.347
MOR	198.8	0.128	0.134
BAS	252.6	7.821	64.852
ELV	4322.0	11.591	12.271
CA	1539.4	8.449	12.440
AIL	6697.0	2.431	2.950
TIC	6597.8	73.428	904.269

Tabla 4.4: Resultado inicial usando la BR generada por FS-MOGUL. Los resultados de $ECM_{ENT/TST}$ de la tabla se deben multiplicar respectivamente por 10^{-6}, 10^{8}, 10^{-4}, 10^{2}, 10^{4}, 10^{-6}, 10^{-8} y 10^{-4} en los problemas DEL, CAL, KIN, FOR, BAS, ELV, AIL y TIC

Nombre	**#R**	**ECM_{ENT}**	**ECM_{TST}**
DEL	589.8	2.979	3.426
ABA	190.0	14.461	14.646
CAL	713.2	132.370	133.311
CON	645.2	74.397	99.676
KIN	10000.0	132.381	282.011
PUM	10000.0	11.247	23.925
STP	156.0	3.297	5.332
WAN	296.8	15.017	18.553
WINR	652.4	0.406	0.642
WINW	577.2	0.585	0.628
FOR	747.2	27.750	751.659
MOR	98.8	0.475	0.488
BAS	500.0	16.060	128.301
ELV	1747.6	61.887	64.067
CA	970.8	25.550	35.050
AIL	2722.4	6.408	19.013
TIC	10000.0	928.853	1939.588

Para todos nuestros experimentos, hemos seguido la misma metodología utilizada en los capítulos anteriores. Esto es:

- se ha considerado un *modelo de validación cruzada de 5 particiones*, con 6 semillas diferentes (véase Sección 2.5.1 del Capítulo 2),

- Para cada conjunto de datos y para cada prueba, se ha generado el frente de Pareto aproximado y luego, se ha calculado los resultados promedio de los 30 frentes,

- Para comparar los diferentes modelos multiobjetivos hemos considerado sólo los tres puntos del frente de Pareto más representativos: el punto más preciso (MAX $_{PRE}$), el punto mediano (MEDIO $_{INT\ /\ PRE}$) y el punto más interpretable (MAX $_{INT}$) o más simple, con idea de por un lado, tener la posibilidad de comparar estadísticamente las soluciones más óptimas en cada uno de los objetivos, es decir, en MAX $_{PRE}$ y en MAX $_{INT}$, y por otro, poder mostrar la tendencia del resto del frente de Pareto obtenido por los diferentes modelos,

- Como herramienta complementaria para comparar los frentes de Pareto hemos utilizado el ratio CS [ZDT00] (véase Subsección 3.3.1 y Expresión 3.10 del Capítulo 3), y por último

- Se ha realizado un análisis estadístico para determinar si existen diferencias significativas entre los resultados obtenidos por los distintos métodos, previa normalización de los datos obtenidos mediante la diferencia normalizada DIFF (véase Expresión 2.8 del Capítulo 2), utilizando los test no paramétricos de Friedman [Fri37] y de Finner [Fin93], con un nivel de confianza, $\alpha = 0.05$.

Los valores de los parámetros de entrada considerados para todos los modelos (OCA_{PAD}, OCA_{DUB}, OCA_{DOM} y OCA_{FRA}) han sido los siguientes: tamaño de la población de 61 individuos, tamaño de la población externa de 61 individuos, 200000 evaluaciones y 100% de probabilidad de cruce. En el caso de OCA_{PAD} la probabilidad de mutación por cromosoma es de 0.3, mientras que en el resto de modelos es de 0.2 para la parte entera y parte real respectivamente, por ser ésta la probabilidad que mejor comportamiento presenta en estos modelos.

4.4.2. Resultados y Análisis

Para analizar los resultados realizamos dos estudios: uno en el punto de máxima precisión (MAX $_{PRE}$) y otro en el frente de Pareto completo. Además, complementamos el análisis con un estudio de los tiempos computacionales y de la escalabilidad de los algoritmos empleados.

Para ello en primer lugar comparamos las soluciones más precisas de nuestra propuesta con respecto a las obtenidas por las otras tres contra las que nos comparamos. A continuación mostramos los costes computacionales de los diferentes algoritmos y analizamos la escalabilidad del planteamiento propuesto. Por último, para cada conjunto de datos del estudio experimental comparamos los frentes de Pareto de nuestra propuesta con respecto a la de las otras tres. Para ello, trazamos los frentes de Pareto promedio. Estas gráficas proporcionan información fiable sobre la forma y características del frente de Pareto obtenido, lo que nos permite comprobar y analizar la tendencia y el tipo de correlación entre el error de entrenamiento y el error de test.

4.4.2.1. Resultados y Análisis de la Solución más Precisa

En esta sección se presenta y analiza los resultados obtenidos en el punto de máxima precisión (MAX $_{PRE}$) por el modelo OCA_{PAD} propuesto frente a los resultados obtenidos por los modelos que utilizan las t-normas clásicas adaptativas como operador de conjunción OCA_{DUB}, OCA_{DOM} y OCA_{FRA}.

Las Tablas 4.5 y 4.6 muestran los resultados obtenidos por los modelos estudiados cuando se utiliza respectivamente, la BR inicial generada por WM y FS-MOGUL.

Los valores de las tablas, agrupados en columnas según el modelo utilizado, representan los resultados medios obtenidos por cada algoritmo en todos los conjuntos de datos estudiados. Para cada uno de ellos, la primera columna muestra el número medio de reglas (#*R*), mientras que la segunda y tercera columnas muestran respectivamente el ECM promedio alcanzado en entrenamiento y test.

Tabla 4.5: Resultados medios de los diferentes algoritmos en complejidad y precisión (entrenamiento y test) para el punto MAX $_{PRE}$ usando como BR inicial la generada por WM. Los resultados de $ECM_{ENT/TST}$ de la tabla se deben multiplicar respectivamente por 10^{-6}, 10^{8}, 10^{-4}, 10^{2}, 10^{4}, 10^{-6}, 10^{-8} y 10^{-4} en DEL, CAL, KIN, FOR, BAS, ELV, AIL y TIC

Datos	OCA_{PAD}			OCA_{DUB}			OCA_{DOM}			OCA_{FRA}		
	#R	ECM_{ENT}	ECM_{TST}	#R	ECM_{ENT}	ECM_{TST}	#R	ECM_{ENT}	ECM_{TST}	#R	ECM_{ENT}	ECM_{TST}
DEL	256.13	1.0423	1.2177	294.10	1.0445	1.2223	242.80	1.0873	1.2405	240.93	1.1158	1.3514
ABA	75.07	2.3757	2.5950	82.57	2.3744	2.6052	73.63	2.4892	2.6621	79.13	2.3462	2.6505
CAL	215.13	21.0229	21.9481	239.70	21.0328	21.9599	172.70	24.6511	25.3395	232.47	20.6081	22.1077
CON	156.47	15.2526	31.2032	179.00	15.0353	31.1829	164.63	18.9389	35.0770	174.20	15.8395	36.5558
KIN	4968.40	30.5995	108.0379	5197.27	26.9888	109.2050	5057.03	33.1820	92.8559	5865.83	36.8488	114.9363
PUM	5050.47	2.0069	7.6021	5291.20	1.4989	8.6435	5088.40	2.4072	7.6352	5406.47	1.4594	8.9901
STP	95.77	0.4777	0.6004	113.13	0.4693	0.6003	89.40	0.6133	0.7815	94.83	0.5999	0.8328
WAN	212.14	1.6969	4.4078	230.87	1.6659	4.4439	188.07	2.3553	4.6864	204.60	2.4138	5.6817
WINR	195.33	0.1355	0.2189	219.83	0.1355	0.2183	163.40	0.1591	0.2241	244.57	0.1165	0.2358
WINW	334.27	0.2098	0.2570	367.90	0.2076	0.2567	261.67	0.2349	0.2673	428.27	0.1875	0.2638
FOR	179.63	9.6025	284.2226	179.10	9.5502	289.0661	174.27	9.6009	305.6473	168.47	9.3414	308.6922
MOR	75.73	0.0427	0.0586	81.90	0.0427	0.0593	68.37	0.0469	0.0630	53.43	0.1163	0.1203
BAS	129.47	4.9020	57.7261	136.23	4.8270	61.4822	127.70	5.4407	57.5143	132.33	5.2430	63.1273
ELV	2676.47	5.1144	6.6920	-	-	-	-	-	-	-	-	-
CA	707.63	3.1529	9.4692	-	-	-	-	-	-	-	-	-
AIL	4723.23	1.2918	2.1573	-	-	-	-	-	-	-	-	-
TIC	6314.40	69.8025	898.5079	-	-	-	-	-	-	-	-	-

Tabla 4.6: Resultados medios de los diferentes algoritmos en complejidad y precisión (entrenamiento y test) para el punto MAX $_{PRE}$ usando como BR inicial la de FS-MOGUL. Los resultados de $ECM_{ENT/TST}$ de la tabla se deben multiplicar respectivamente por 10^{-6}, 10^{8}, 10^{-4}, 10^{2}, 10^{4}, 10^{-6}, 10^{-8} y 10^{-4} en DEL, CAL, KIN, FOR, BAS, ELV, AIL y TIC

Datos	OCA_{PAD}			OCA_{DUB}			OCA_{DOM}			OCA_{FRA}		
	#R	ECM_{ENT}	ECM_{TST}	#R	ECM_{ENT}	ECM_{TST}	#R	ECM_{ENT}	ECM_{TST}	#R	ECM_{ENT}	ECM_{TST}
DEL	228.30	1.0833	1.2311	260.83	1.0894	1.2377	228.17	1.1196	1.2437	181.13	1.1460	1.3214
ABA	74.50	2.3844	2.5992	80.73	2.3832	2.5743	68.93	2.4348	2.5903	72.07	2.3890	2.6604
CAL	229.20	24.0739	25.0128	248.87	24.0855	25.0122	200.90	26.9570	27.7096	242.57	25.0647	26.1682
CON	223.03	15.4176	32.4437	258.63	15.9592	32.9127	238.50	19.5425	36.2815	232.33	15.5709	33.8978
KIN	7767.70	52.6087	147.8134	8257.33	52.1064	148.2074	8247.87	55.2433	140.9773	8902.63	65.7062	163.4085
PUM	7654.63	3.8650	10.8437	8263.17	3.7888	10.8523	8264.67	4.1565	10.9331	8358.67	4.0480	11.8748
STP	60.80	0.8821	1.0596	62.90	0.8914	1.0688	55.77	1.0135	1.2139	49.50	1.0640	1.3996
WAN	111.70	3.3672	6.5059	124.53	3.5356	6.7360	104.73	3.7667	7.0111	89.03	4.9432	9.1283
WINR	225.40	0.1386	0.2313	261.30	0.1393	0.2325	221.53	0.1627	0.2338	211.30	0.1315	0.2488
WINW	226.70	0.2328	0.2694	269.90	0.2321	0.2687	222.57	0.2479	0.2746	217.47	0.2285	0.2812
FOR	228.67	2.2664	477.7946	211.90	2.2086	487.5386	226.37	3.8820	483.7357	186.27	2.0400	488.6680
MOR	50.20	0.1097	0.1348	47.67	0.1145	0.1395	40.43	0.1019	0.1245	31.20	0.3064	0.3680
BAS	172.60	5.1023	81.8764	184.73	5.1309	81.3165	176.73	5.6304	78.2598	142.20	5.4827	82.8408
ELV	782.77	8.0021	10.6360	-	-	-	-	-	-	-	-	-
CA	325.23	5.9463	12.9131	-	-	-	-	-	-	-	-	-
AIL	1501.57	1.6639	2.7844	-	-	-	-	-	-	-	-	-
TIC	6586.37	464.4229	969.9089	-	-	-	-	-	-	-	-	-

En los problemas ELV, CA, AIL y TIC (los más complejos en dimensionalidad y escalabilidad) no se muestra ningún valor para los modelos OCA_{DUB}, OCA_{DOM} y OCA_{FRA} ya que el gran número de variables e instancias que poseen, unido a la gran cantidad de reglas iniciales que tienen, -véase la parte inferior de las Tablas 4.1, 4.3 y 4.4-, han hecho imposible su ejecución (debido a la enorme cantidad de tiempo que necesitan para ello) a pesar del esfuerzo llevado a cabo para optimizar y mejorar los correspondientes algoritmos que implementan dichos modelos.

Para comparar todos los modelos estudiados con el fin de determinar si el método propuesto está funcionando correctamente o no en función del error en el conjunto de test y el número de reglas, se ha realizado un análisis estadístico para determinar si existen diferencias significativas entre los resultados obtenidos.

La Tabla 4.7 muestra la clasificación (ranking) de los diferentes modelos considerados en este estudio para las BR de WM (parte izquierda de la tabla) y FS-MOGUL (parte derecha de la tabla), obtenidas al aplicar el test de Friedman con un nivel de confianza $\alpha = 0.05$.

Tabla 4.7: Ranking obtenido por el test de Friedman en las métricas ECM_{TST} y $\#R$ en el punto de máxima precisión MAX_{PRE} con la BR inicial de WM y FS-MOGUL

Algoritmo	BR WM		BR FS-MOGUL	
	Ranking en ECM_{TST} (*p-fried*: 0.000004)	Ranking en $\#R$ (*p-fried*: 0.000404)	Ranking en ECM_{TST} (*p-fried*: 0.000014)	Ranking en $\#R$ (*p-fried*: 0.004474)
OCA_{PAD}	1.3529	1.9412	1.4706	2.2353
OCA_{DUB}	2.1765	3.4706	2.2353	3.4412
OCA_{DOM}	2.8824	1.8235	2.6471	2.3824
OCA_{FRA}	3.5882	2.7647	3.6471	1.9412

Los resultados del test de Friedman nos indica que hay diferencias significativas en ECM_{TST} y en $\#R$ entre los resultados observados en todos los conjuntos de datos (*p-fried* < 0.05), con independencia de la BR inicial (WM o FS-MOGUL) utilizada. Observando la tabla podemos ver que el modelo OCA_{PAD} es el que obtiene la mejor clasificación en precisión (ECM_{TST}) con ambas BRs, mientras que en complejidad o interpretabilidad ($\#R$) la mejor clasificación es obtenida por OCA_{DOM} con la BR de WM y por OCA_{FRA} en caso de usar la BR de FS-MOGUL, aunque en ambos casos, nuestra propuesta, que es la segunda mejor en el ranking, obtiene unos valores muy cercano a ellos.

En todos los casos es posible aplicar a posteriori el procedimiento post-hoc de Finner para comparar el modelo con mejor clasificación en cada caso con

los restantes métodos, para determinar si las diferencias detectadas por el test de Friedman son o no estadísticamente significativas. Las Tablas 4.8 y 4.9 presentan los resultados de dicha comparación.

Tabla 4.8: Tabla de Finner con $\alpha = 0.05$ para los modelos en ECM_{TST} y $\#R$ en el punto de máxima precisión MAX_{PRE} usando como BR inicial la de WM
Algoritmo de control: OCA_{PAD} para ECM_{TST} y OCA_{DOM} para $\#R$

	ECM_{TST}			$\#R$		
i	Algoritmo	*p-finner*	Hipótesis	Algoritmo	*p-finner*	Hipótesis
1	OCA_{DUB}	0.062915	Aceptada	OCA_{PAD}	0.790482	Aceptada
2	OCA_{DOM}	0.000829	Rechazada	OCA_{FRA}	0.049896	Rechazada
3	OCA_{FRA}	0.000001	Rechazada	OCA_{DUB}	0.000599	Rechazada

Tabla 4.9: Tabla de Finner con $\alpha = 0.05$ para los modelos en ECM_{TST} y $\#R$ en el punto de máxima precisión MAX_{PRE} usando como BR inicial la de FS-MOGUL
Algoritmo de control: OCA_{PAD} para ECM_{TST} y OCA_{FRA} para $\#R$

	ECM_{TST}			$\#R$		
i	Algoritmo	*p-finner*	Hipótesis	Algoritmo	*p-finner*	Hipótesis
1	OCA_{DUB}	0.084177	Aceptada	OCA_{PAD}	0.506555	Aceptada
2	OCA_{DOM}	0.011808	Rechazada	OCA_{DOM}	0.438139	Aceptada
3	OCA_{FRA}	0.000003	Rechazada	OCA_{DUB}	0.002115	Rechazada

En estas tablas, los algoritmos están ordenados con respecto al p-valor obtenido con respecto al algoritmo de control. El test de Finner rechaza la hipótesis de igualdad (hipótesis nula) cuando el *p-finner* es < 0.05. Observando las tablas, podemos ver que el test de Finner rechaza la hipótesis de igualdad en precisión (ECM_{TST}) cuando el algoritmo de control es OCA_{PAD}, tanto para WM como para FS-MOGUL, excepto para OCA_{DUB}, por lo que no podemos afirmar que con respecto a ese objetivo nuestra propuesta sea estadísticamente mejor que esa. No obstante OCA_{PAD} es mejor que OCA_{DUB} en número de reglas ($\#R$).

En cuanto a complejidad o número de reglas ($\#R$) los algoritmos de control son OCA_{DOM} y OCA_{FRA} para WM y FS-MOGUL respectivamente. En estos casos el test de Finner no rechaza (acepta) la hipótesis de igualdad cuando se compara con OCA_{PAD}. Por tanto, no hay diferencias significativas entre los resultados, por lo que con respecto a dicho objetivo, OCA_{PAD} podría considerarse similar a los algoritmos de control.

Analizando los resultados que se muestran en las Tablas 4.5 y 4.6, y las evidencias estadísticas obtenidas en las Tablas 4.7, 4.8 y 4.9 podemos destacar que en el punto de máxima precisión (MAX_{PRE}):

- OCA_{PAD} es el modelo más preciso y obtiene los mejores resultados en $ECM_{TST.}$

- Aunque la precisión alcanzada por OCA_{DUB} es muy similar, no apreciándose de hecho diferencias estadísticas, nuestra propuesta obtiene modelos mucho más simples.

- En cuanto al número de reglas (#*R*), no existen diferencias significativas entre las obtenidas por el mejor método (OCA_{DOM} en WM y OCA_{FRA} en FS-MOGUL) y nuestra propuesta OCA_{PAD}, alcanzando la nuestra precisiones mucho más altas.

4.4.2.2. Estudio de los Tiempos Computacionales y de la Escalabilidad

En esta sección, se analiza cómo el uso de OCA_{PAD} permite reducir considerablemente los tiempos de ejecución con respecto al uso de los operadores clásicos de conjunción adaptativos. Estos tiempos han sido obtenidos con un equipo basado en un procesador Intel® Core™ 2 Quad Q9650 (12M cache, 3,00 GHz, 1333 MHz FSB) con 4 GB de memoria y un sistema operativo Linux de 64-bit usando sólo uno de los cuatro núcleos.

Las Tablas 4.10 y 4.11 muestran los tiempos medios de ejecución de las 30 ejecuciones empleados por los modelos estudiados cuando se utiliza respectivamente, la BR inicial de WM y FS-MOGUL. Los tiempos mostrados en las tablas no tienen en cuenta el tiempo requerido para obtener la BR inicial. Además de los tiempos, también se muestran la ganancia de tiempo obtenida al emplear nuestra propuesta respecto a las otras (calculado como el cociente resultante de dividir el tiempo empleado por nuestra propuesta entre los tiempos empleados por los diferentes modelos con los que nos comparamos). Para los problemas ELV, CA, AIL y TIC sólo se muestran los tiempos obtenidos por nuestra propuesta ya que como comentamos en la sección anterior, su ejecución ha sido imposible para el resto de modelos debido a la enorme cantidad de tiempo que se necesitaría para ello.

Tabla 4.10: Tiempo medio de ejecución de los diferentes modelos en horas, minutos y segundos (hh:mm:ss) usando como BR inicial la de WM

Conjunto de Datos	OCA_{PAD}	OCA_{DUB}	Ganancia de tiempo	OCA_{DOM}	Ganancia de tiempo	OCA_{FRA}	Ganancia de tiempo
DEL	03:03:42	25:40:36	8.4	93:48:13	30.6	102:26:21	33.5
ABA	00:29:06	02:59:07	6.2	10:38:39	21.9	12:01:17	24.8
CAL	07:13:20	56:13:34	7.8	246:52:09	34.2	268:52:28	37.2
CON	00:07:17	00:56:59	7.8	02:26:26	20.1	03:11:06	26.2
KIN	12:19:00	131:40:32	10.7	331:41:08	26.9	361:13:10	29.3
PUM	11:35:49	135:11:52	11.7	334:51:08	28.9	371:43:43	32.1
STP	00:07:45	01:03:42	8.2	04:30:02	34.8	04:44:10	36.7
WAN	00:21:45	03:36:42	10.0	15:39:19	43.2	18:11:26	50.2
WINR	00:42:02	08:21:49	11.9	43:54:40	62.7	46:50:19	66.9
WINW	03:35:14	57:59:37	16.2	339:36:34	94.7	363:27:32	101.3
FOR	00:04:08	00:36:32	8.8	01:58:10	28.6	02:15:28	32.7
MOR	00:07:24	01:26:36	11.7	07:55:42	64.3	08:16:32	67.2
BAS	00:02:02	00:35:46	17.6	03:23:03	99.9	03:44:45	110.6
ELV	28:17:34	-	-	-	-	-	-
CA	07:34:59	-	-	-	-	-	-
AIL	39:20:49	-	-	-	-	-	-
TIC	15:27:27	-	-	-	-	-	-

Tabla 4.11: Tiempo medio de ejecución de los diferentes modelos en horas, minutos y segundos (hh:mm:ss) usando como BR inicial la de FS-MOGUL

Conjunto de Datos	OCA_{PAD}	OCA_{DUB}	Ganancia de tiempo	OCA_{DOM}	Ganancia de tiempo	OCA_{FRA}	Ganancia de tiempo
DEL	03:05:42	22:56:01	7.4	84:49:42	27.4	94:44:31	30.6
ABA	00:29:27	02:56:51	6.0	11:31:26	23.5	12:35:10	25.6
CAL	05:36:19	40:32:53	7.2	150:35:13	26.9	164:10:55	29.3
CON	00:14:12	01:50:15	7.8	04:38:20	19.6	06:11:42	26.2
KIN	18:14:59	211:19:54	11.6	480:14:50	26.3	534:27:57	29.3
PUM	18:23:55	197:17:38	10.7	471:32:51	25.6	515:19:24	28.0
STP	00:04:43	00:37:04	7.9	02:44:02	34.8	02:51:08	36.3
WAN	00:14:38	02:17:53	9.4	10:13:59	41.9	11:44:54	48.1
WINR	00:30:33	06:21:23	12.5	32:16:38	63.4	34:17:32	67.4
WINW	01:40:05	27:33:21	16.5	157:50:35	94.6	172:14:05	103.3
FOR	00:08:06	01:09:55	8.6	03:47:30	28.1	04:42:53	35.0
MOR	00:03:35	00:36:52	10.3	03:34:38	59.8	03:36:07	60.2
BAS	00:03:33	01:06:46	18.8	06:18:47	106.7	06:58:51	118.0
ELV	09:12:11	-	-	-	-	-	-
CA	03:13:09	-	-	-	-	-	-
AIL	12:11:09	-	-	-	-	-	-
TIC	20:26:54	-	-	-	-	-	-

Observando las tablas podemos ver que el método OCA_{PAD} es entre 6.0 y 18.8 veces más rápido que OCA_{DUB} (el más rápido de las otras propuestas), lo que permite ahorrar una considerable cantidad de tiempo. En cuanto a los conjuntos de datos más complejos ELV, CA, AIL y TIC, aunque nuestra propuesta OCA_{PAD} necesita una gran cantidad de tiempo, consigue obtener modelos que son imposibles de alcanzar con el resto de propuestas con las que nos comparamos.

Analizando los resultados de dichas tablas podemos concluir lo siguiente:

- La reducción de tiempo del modelo propuesto es muy significativa, para todos los conjuntos de datos y para todos los modelos.

- La influencia del número de reglas y ejemplos es muy importante. Las diferencias significativas que se observan entre los resultados obtenidos usando la BR de WM y FS-MOGUL son debidas a la diferencia entre el número de reglas iniciales que dichos métodos producen.

- La ganancia de tiempo obtenida por nuestra propuesta OCA_{PAD} depende, entre otros motivos, de la combinación de varios factores: complejidad del conjunto de datos, número de variables, número de reglas iniciales y número de ejemplos en el conjunto de entrenamiento, número de reglas disparadas por cada ejemplo y la posibilidad de reducir el número de reglas durante la ejecución (pensamos que cuando la selección de reglas tiene dificultades para reducir el conjunto de reglas iniciales, OCA_{PAD} reduce más tiempo que los otros modelos debido a que la dificultad en los tradicionales modelos de conjunción adaptativa es mayor y más problemática).

Para que el lector pueda apreciar la influencia de dichos factores en la ganancia de tiempo obtenida, en la siguiente tabla se muestran de forma conjunta todos ellos:

Tabla 4.12: Influencia de varios factores en la ganancia de tiempo de OCA_{PAD} cuando se usa las BRs iniciales de WM y FS-MOGUL

Conjunto de Datos	Variables	Ejemplos	BR WM			BR FS-MOGUL		
			#R	#R disparadas por ejemplos	Tiempo OCA_{PAD}	#R	#R disparadas por ejemplos	Tiempo OCA_{PAD}
DEL	6	9517	679.8	45.1	03:03:42	589.8	47.3	03:05:42
ABA	8	4177	199.0	19.9	00:29:06	190.0	22.3	00:29:27
CAL	8	20640	623.8	59.6	07:13:20	713.2	32.0	05:36:19
CON	8	1030	309.8	8.8	00:07:17	645.2	16.5	00:14:12
KIN	8	8192	6423.2	9.5	12:19:00	10000.0	11.5	18:14:60
PUM	8	8192	6435.6	9.6	11:35:49	10000.0	10.4	18:23:55
STP	9	950	265.4	19.7	00:07:45	156.0	12.7	00:04:43
WAN	9	1609	456.8	36.7	00:21:45	296.8	22.1	00:14:38
WINR	11	1599	714.2	78.8	00:42:02	652.4	46.8	00:30:33
WINW	11	4898	1019.8	169.9	03:35:14	577.2	63.9	01:40:05
FOR	12	517	374.6	5.1	00:04:08	747.2	8.9	00:08:06
MOR	15	1049	198.8	21.8	00:07:24	98.8	10.9	00:03:35
BAS	16	337	252.6	7.5	00:02:02	500.0	9.9	00:03:33
ELV	18	16599	4322.0	208.7	28:17:34	1747.6	64.7	09:12:11
CA	21	8192	1539.4	168.1	07:34:60	970.8	57.1	03:13:09
AIL	40	13750	6697.0	417.3	39:20:49	2722.4	83.7	12:11:09
TIC	85	9822	6597.8	1.4	15:27:27	10000.0	2.2	20:26:54

4.4.2.3. Análisis de los Frentes de Pareto

En esta sección, se analiza la efectividad del algoritmo propuesto en las soluciones restantes que se obtienen en los frentes de Pareto. Para ello, comparamos para cada problema, cada uno de los 30 frentes de Pareto obtenidos con nuestra propuesta con los frentes de Pareto respectivos obtenidos con las otras, utilizando para ello el ratio CS. De este modo, podemos analizar las diferencias entre las soluciones obtenidas por el algoritmo OCA_{PAD} respecto a la alcanzada con los algoritmos OCA_{DUB}, OCA_{DOM} y OCA_{FRA} en términos de los frentes de Pareto.

Este criterio ya fue usado en el estudio experimental del Capítulo 3 por proporcionar información más fiable que la proporcionada por el criterio clásico utilizado también en los capítulos anteriores, consistente en comparar el promedio de puntos de máxima interpretabilidad (MAX_{INT}), de precisión e interpretabilidad media ($MEDIA_{(INT\ /\ PRE)}$) y de máxima precisión (MAX_{PRE}) de los Paretos obtenidos con cada propuesta.

En las Tablas 4.13 y 4.14 se muestran los ratios CS (más concretamente, los promedios de los 30 ratios CS) obtenidos por cada problema. Cada tabla muestra, cada dos columnas, el resultado de comparar nuestra propuesta con las otras (que usan los operadores adaptativos de conjunción clásicos) y viceversa, resaltándose en negrita los resultados con mejor comportamiento. Obsérvese que en esta dos tablas sólo mostramos los primeros 13 conjuntos de datos, al no haberse podido ejecutar los modelos de conjunción adaptativa clásicos OCA_{DUB}, OCA_{DOM} y OCA_{FRA} para los restantes cuatro.

Tabla 4.13: Ratios CS de todos los frentes de Pareto para la BR de WM

Conjunto de Datos	**OCA_{PAD} vs. OCA_{DUB}**	**OCA_{DUB} vs. OCA_{PAD}**	**OCA_{PAD} vs. OCA_{DOM}**	**OCA_{DOM} vs. OCA_{PAD}**	**OCA_{PAD} vs. OCA_{FRA}**	**OCA_{FRA} vs. OCA_{PAD}**
DEL	**0.719**	0.000	**0.599**	0.123	**0.630**	0.000
ABA	**0.777**	0.229	**0.874**	0.114	**0.849**	0.155
CAL	**0.657**	0.005	**0.050**	0.000	**0.690**	0.026
CON	**0.677**	0.145	**0.873**	0.116	**0.936**	0.048
KIN	**0.693**	0.000	**0.000**	0.000	**0.971**	0.000
PUM	**0.861**	0.000	**0.700**	0.000	**1.000**	0.000
STP	**0.587**	0.112	**0.806**	0.029	**0.899**	0.027
WAN	**0.500**	0.024	**0.313**	0.165	**0.612**	0.068
WINR	**0.474**	0.062	0.008	**0.122**	**0.800**	0.000
WINW	**0.565**	0.000	**0.000**	**0.000**	**0.833**	0.000
FOR	**0.547**	0.273	**0.550**	0.342	**0.554**	0.302
MOR	**0.580**	0.311	**0.491**	0.457	**0.266**	0.000
BAS	**0.682**	0.224	**0.538**	0.432	**0.715**	0.254

Tabla 4.14: Ratios CS de todos los frentes de Pareto para la BR de FS-MOGUL

Conjunto de Datos	OCA_{PAD} vs. OCA_{DUB}	OCA_{DUB} vs. OCA_{PAD}	OCA_{PAD} vs. OCA_{DOM}	OCA_{DOM} vs. OCA_{PAD}	OCA_{PAD} vs. OCA_{FRA}	OCA_{FRA} vs. OCA_{PAD}
DEL	**0.794**	0.008	**0.790**	0.128	**0.031**	0.000
ABA	**0.586**	0.380	0.361	**0.629**	**0.770**	0.229
CAL	**0.540**	0.087	**0.157**	0.000	**0.291**	0.021
CON	**0.739**	0.000	**0.989**	0.004	**0.755**	0.151
KIN	**0.533**	0.000	**0.000**	0.000	**0.994**	0.000
PUM	**0.586**	0.000	**0.560**	0.000	**1.000**	0.000
STP	**0.652**	0.337	**0.814**	0.138	**0.754**	0.075
WAN	**0.687**	0.178	**0.552**	0.331	**0.262**	0.057
WINR	**0.640**	0.006	**0.387**	0.290	**0.465**	0.045
WINW	**0.515**	0.000	**0.739**	0.171	**0.781**	0.006
FOR	**0.071**	0.067	**0.391**	0.122	0.000	**0.098**
MOR	**0.827**	0.194	0.155	**0.833**	**0.666**	0.008
BAS	**0.451**	0.272	**0.423**	0.326	0.004	**0.593**

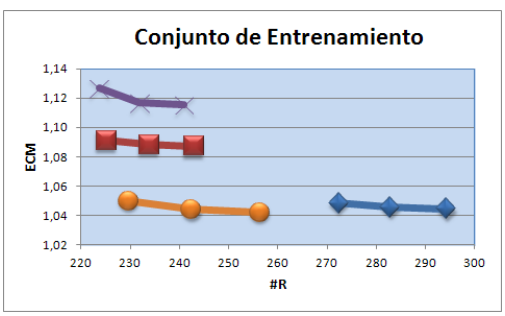
Los resultados muestran que independientemente de la BR empleada (WM o FS-MOGUL), en la mayoría de los problemas, nuestra propuesta OCA_{PAD} logra un ratio CS mayor que los modelos basados en las t-normas adaptativas clásicas con las que nos comparamos, lo que indica claramente que las soluciones obtenidas por nuestra propuesta dominan a las obtenidas por las otras, siendo por tanto, mejor que aquellas.

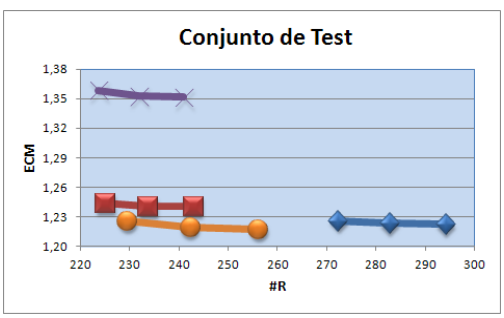
La Figura 4.1 muestra los frentes de Pareto medios obtenidos por cada una de las propuestas (OCA_{PAD}, OCA_{DUB}, OCA_{DOM} y OCA_{FRA}) en los conjuntos de datos más representativos estudiados. Las gráficas han sido dibujadas conectando entre si los tres puntos más representativos de cada Pareto (MAX_{INT}, $MEDIA_{(INT\ /\ PRE)}$ y MAX_{PRE}) descritos al principio de esta sección, con el fin de mostrar los hipotéticos frentes de Pareto.

Una comparación visual de los frentes de Pareto obtenidos por cada una de las propuestas pone de relieve el interés de la solución obtenida por OCA_{PAD} al estar su curva por debajo de las otras tres en la mayor parte del Pareto, lo que implica que para el mismo nivel de precisión obtiene soluciones más interpretables. Además podemos ver que en la mayoría de los conjuntos de datos, las soluciones en test de OCA_{PAD} son mejores que en entrenamiento, respecto a las obtenidos por las otras propuestas (t-normas adaptativas clásicas), lo que indica que nuestra propuesta aprende mejor que las otras.

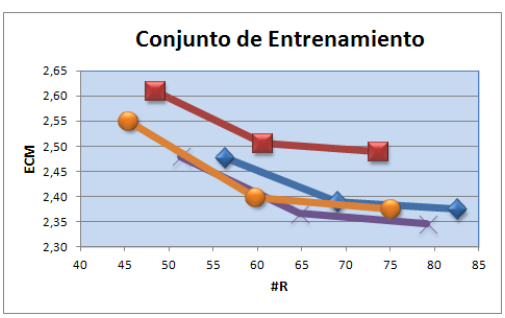

OCADUB
OCADOM
OCAFRA
OCAPAD

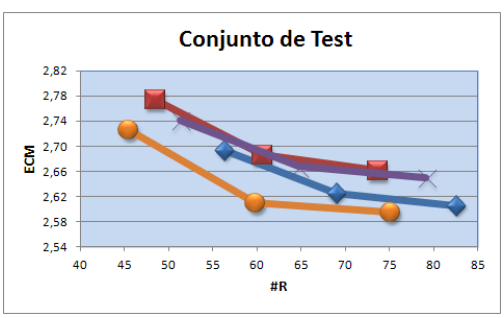
DEL


Conjunto de Entrenamiento
ECM
1,14
1,12
1,10
1,08
1,06
1,04
1,02
220 230 240 250 260 270 280 290 300
#R

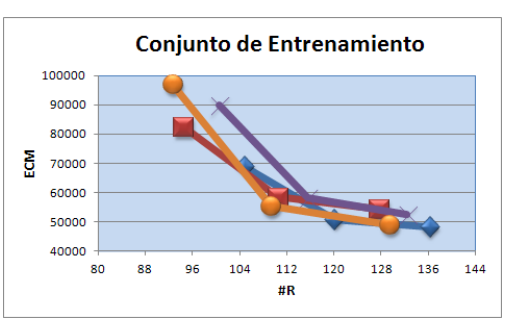

Conjunto de Test
ECM
1,38
1,35
1,32
1,29
1,26
1,23
1,20
220 230 240 250 260 270 280 290 300
#R

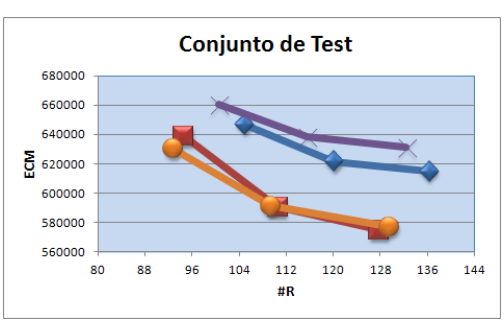
ABA


Conjunto de Entrenamiento
ECM
2,65
2,60
2,55
2,50
2,45
2,40
2,35
2,30
40 45 50 55 60 65 70 75 80 85
#R


Conjunto de Test
ECM
2,82
2,78
2,74
2,70
2,66
2,62
2,58
2,54
40 45 50 55 60 65 70 75 80 85
#R

BAS


Conjunto de Entrenamiento
ECM
100000
90000
80000
70000
60000
50000
40000
80 88 96 104 112 120 128 136 144
#R


Conjunto de Test
ECM
680000
660000
640000
620000
600000
580000
560000
80 88 96 104 112 120 128 136 144
#R

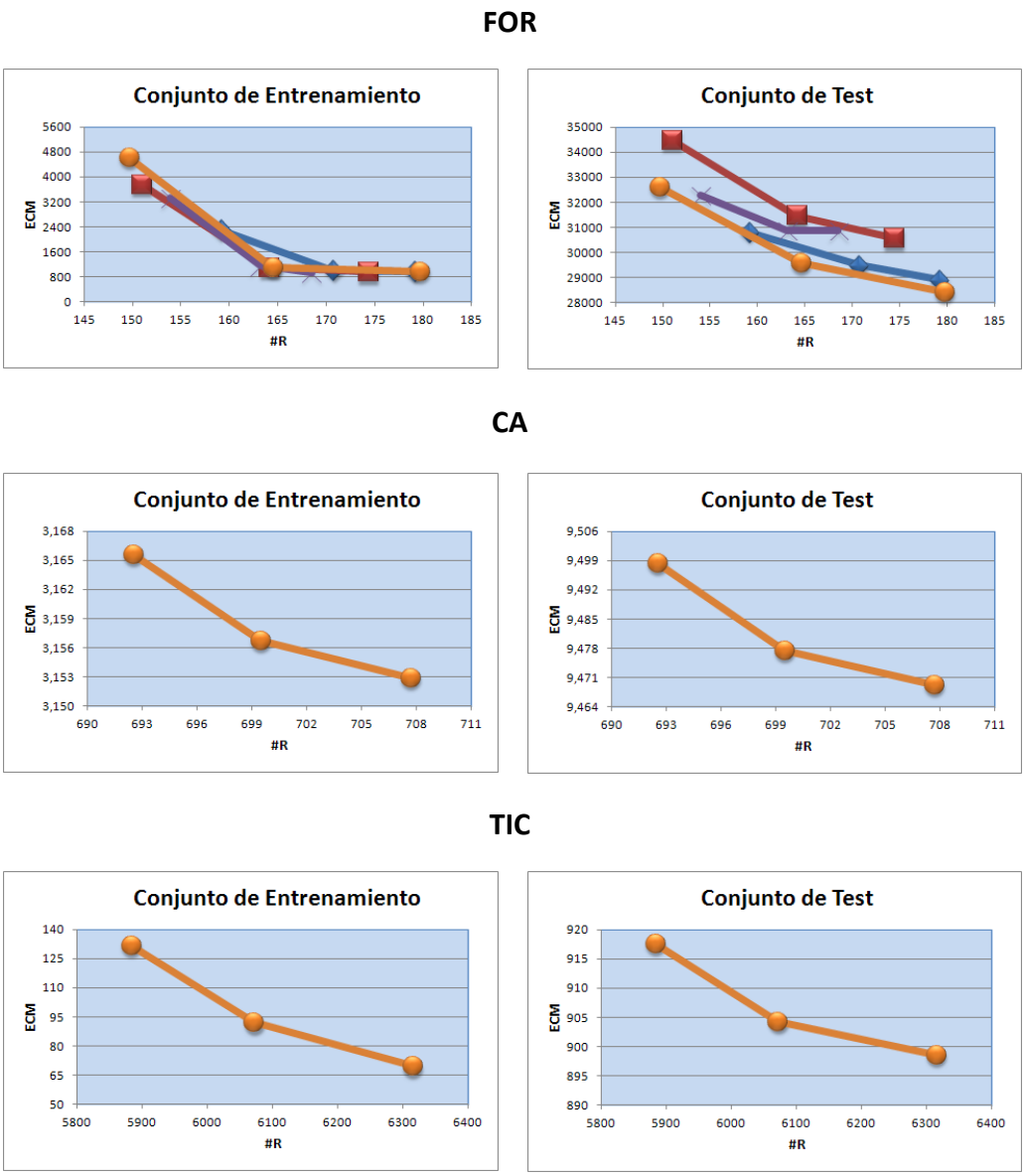


Figura 4.1: Frentes de Paretos medios obtenidos por OCA_{PAD}, OCA_{DUB}, OCA_{DOM} y OCA_{FRA} en diferentes problemas usando la BR inicial de WM

Este es un punto muy importante a destacar de nuestra propuesta, sobre las otras. Esto es, existe una correlación muy alta entre los valores en el conjunto de entrenamiento y los valores en el test. Para cada uno de los conjunto de datos considerado se puede apreciar que las soluciones promedio en el conjunto de test son muy similares a las soluciones obtenidas en el conjunto de entrenamiento. Además, podemos ver que no hay ningún sobreajuste o sobreaprendizaje en las diferentes partes de los frentes de Pareto obtenidos. Este comportamiento es más notable para los conjuntos de datos de alta dimensionalidad.

4.5. Sumario

En este capítulo, hemos propuesto un SDEMO de inferencia adaptativa eficaz para problemas de alta dimensionalidad. Nuestra propuesta, basada en un novedoso operador de conjunción adaptativo denominado OCA_{PAD} permite aumentar la escalabilidad para este tipo de problemas, al trabajar con valores discretos en lugar de con valores continuos, lo que le permite ser más rápido y eficiente que los operadores adaptativos clásicos.

Este operador, por sus características, es ideal para ser usado en problemas de alta dimensionalidad ya que permite por un lado precalcular parte del proceso, de forma que, mediante una eficaz implementación que aprovecha los cálculos ya realizados, acorta el tiempo necesario para realizar la inferencia. Por otro lado, al tener un espacio de búsqueda menor como consecuencia de trabajar con valores discretos, consigue reducir el tiempo necesario empleado en ajustar sus parámetros, haciendo que la convergencia del algoritmo que lo utilice sea más rápida. De esta forma, se consigue obtener mejores resultados, tanto en precisión como en escalabilidad, para este tipo de problemas que los alcanzados con los operadores de conjunción adaptativos clásicos más utilizados en la literatura.

A partir de dicho operador y del mecanismo para acelerar los cálculos de la inferencia, se ha desarrollado un modelo evolutivo multiobjetivo para maximizar la precisión e interpretabilidad en el sentido de la complejidad de los sistemas difusos lingüísticos que funciona de forma más eficiente y rápida que los otros, siendo por tanto ideal para ser usado en problemas de alta dimensionalidad.

Se ha llevado a cabo un extenso estudio experimental en el que se han utilizado diecisiete conjuntos de datos de complejidades diferentes con dos BRs iniciales distintas para comparar el nuevo operador adaptativo propuesto con los operadores adaptativos más conocidos, a través de los respectivos modelos multiobjetivos que hacen uso de dichos operadores. Al igual que en los capítulos anteriores, los resultados del estudio experimental han sido validados con test estadísticos no paramétricos, derivando las siguientes conclusiones:

- OCA_{PAD} mejora en la mayoría de los casos, o al menos iguala, la precisión alcanzada por los modelos que usan operadores de conjunción adaptativos clásicos, obteniendo modelos con una complejidad similar a ellos.

- La reducción de tiempo de aprendizaje de OCA_{PAD} es importante y mayor en los conjuntos de datos más complejos (en dimensionalidad). Además el modelo propuesto puede ser aplicado a conjuntos de datos que son imposibles de modelar con los otros operadores de inferencia adaptativos con los que nos comparamos.

- En la mayoría de los problemas, las soluciones obtenidas por OCA_{PAD} dominan a las obtenidas por las otras, en casi todo el frente del Pareto, lo que indica que nuestra propuesta obtiene por regla general para el mismo nivel de interpretabilidad soluciones más precisas en todas las soluciones con diferente equilibrio entre los objetivos precisión e interpretabilidad.

En resumen, el método propuesto obtiene soluciones más rápidamente sin apenas sobreaprendizaje, consiguiendo mejorar la precisión a la vez que se mantiene la complejidad del modelo obtenido, siendo por tanto ideal para ser aplicado a conjuntos de datos con alta dimensionalidad. Otro interesante aspecto del modelo propuesto es que es complementario y por tanto puede ser usado en combinación con otras metodologías para mejorar la calidad de los modelos difusos obtenidos (por ejemplo, con las basadas en la mejora de la BC) lo que lo hace aún más interesante.

Comentarios Finales

Dedicamos esta sección a la presentación de un resumen de los resultados obtenidos y conclusiones que aporta esta tesis. Asimismo presentamos las publicaciones asociadas a esta tesis y comentamos posibles trabajos futuros en la línea de trabajo de la misma u otras líneas de investigación que se pueden derivar

A. Resumen y Conclusiones

La principal virtud de los SBRDs lingüísticos es su interpretabilidad mientras que su principal inconveniente es su menor precisión comparativamente hablando con otros tipos de SBRDs. Hoy en día, en realidad, ambas cualidades suelen ser necesarias, y por este motivo en los últimos años se ha desarrollado con éxito diversas propuestas de tipo evolutivas multiobjetivo en toda una gama de SDEs (tanto de aprendizaje como de post-procesamiento) constituyéndose prácticamente como una especialidad en sí misma.

En este marco, esta tesis desarrolla nuevas propuestas de aprendizaje y ajuste en el ámbito de los operadores adaptativos del SI y la ID, presentando distintos algoritmos para el aprendizaje de SBRDs compactos y precisos que hace uso de ellos. Los siguientes apartados resumen brevemente los resultados obtenidos y presentan algunas conclusiones sobre los mismos.

A.1. Aprendizaje Cooperativo de los Operados Difusos Adaptativos y de la BR con AGMOs

Se ha propuesto un modelo evolutivo multiobjetivo para el aprendizaje cooperativo y conjunto de los ODAs que intervienen en el SI y la ID, junto con la BR, con el doble objetivo de minimizar tanto el error del sistema como el número de reglas. El aprendizaje de ambos elementos conjuntamente propicia que ambos cooperen, alcanzando una colaboración entre ambas partes del modelo que permite obtener un conjunto de soluciones con diferentes equilibrios óptimos, más precisos que los modelos que aprenden secuencialmente cada uno de ellos incluso manteniendo la filosofía multiobjetivo. Para conseguir este modelo se han empleado dos conocidos AGMOs de segunda generación (SPEA2 y NSGA-II) a los que se les ha añadido mejoras para centrar la búsqueda del algoritmo en la región más apropiada del frente de Pareto para este particular.

El modelo ha sido verificado mediante un estudio experimental en el que participan nueve conjuntos de datos de regresión reales comparando la metodología cooperativa multiobjetivo propuesta con diferentes métodos secuenciales multiobjetivo y diferentes métodos de un solo objetivo basados en la precisión. Los resultados han sido validados con test no paramétricos por pares, en los que se han tenido en cuenta los tres puntos más representativos del Pareto obtenido por cada uno de los métodos multiobjetivo comparados: el punto más interpretable, el mediano y el más preciso.

A.2. Propuestas de Nuevo Índice y Mecanismo para Mejorar la Interpretabilidad de los SBRDs con Defuzzificación Adaptativa Usando AGMOs

Es bien conocido que la defuzzificación adaptativa basada en el uso de un parámetro o peso en cada regla introduce una serie de valores asociados indirectamente a cada regla de la BR, que permiten mejorar la precisión de los SBRDs, pero lo hace a costa de aumentar la complejidad y por tanto decrementar la interpretabilidad del sistema.

Con objeto de cuantificar y reducir en lo posible dicho efecto se ha propuesto utilizar en estos sistemas una serie de medidas de interpretabilidad y un índice que junto con la precisión puede ser empleado en el proceso de

optimización multiobjetivo. La propuesta realizada lleva asociado un mecanismo que reduce el número de reglas, y el número de reglas con peso, de modo que sólo lo emplean aquellas que precisan el uso del parámetro con este factor de ponderación.

Se han planteado dos variantes de dicho mecanismo: uno más simple que utiliza dos parámetros prefijados y otro que ajusta él mismo sus propios parámetros durante el proceso evolutivo de aprendizaje de los pesos o parámetros de la defuzzificación.

La eficacia de la metodología propuesta se ha comprobado mediante la realización de un estudio experimental, validado con test estadísticos no paramétricos, en el que se han usado trece conjuntos de datos de regresión reales con dos BRs diferentes.

El estudio experimental se ha dividido en dos partes, cada una de ellas centrada en uno de los modelos: en la primera, la propuesta con valores fijos ha sido comparada con varios métodos de un solo objetivo. Además se ha analizado el comportamiento práctico del modelo estudiando la influencia que tienen los parámetros internos del mecanismo planteado en el equilibrio entre interpretabilidad y precisión. En la segunda se ha analizado el comportamiento y ventajas del modelo autoadaptativo con respecto al modelo con parámetros internos fijos.

A.3 Incremento de la Escalabilidad en los Modelos basados en AGMOs con Inferencia Adaptativa para Problemas de Alta Dimensionalidad

Cuando el problema que ha de resolverse presenta una alta dimensionalidad, es decir, un elevado número de variables o características de entrada, el diseño de modelos difusos lingüísticos para ellos se complica bastante por el crecimiento exponencial del número de reglas que se produce con el aumento lineal del número de variables. En este sentido, los operadores adaptativos que hay definidos actualmente no son aptos para ser aplicados en este tipo de problemas, ya que la adaptación de sus parámetros se vuelve problemática por el aumento crítico del espacio de búsqueda, que provoca que se necesiten tiempos de ejecución extremadamente grandes para la convergencia de los algoritmos de aprendizaje que lo utilizan.

Si bien en los problemas de regresión (en comparación con otros problemas del ámbito de la Minería de Datos) los conjuntos grandes considerados de alta dimensionalidad son en general relativamente más pequeños, es necesario trabajar en la línea de mejorar sus posibilidades de uso, es decir, en su escalabilidad.

Para solucionar este problema, se ha propuesto un eficaz sistema difuso evolutivo multiobjetivo de Inferencia adaptativa que permite aumentar la escalabilidad y rendimiento de los SBRDs que lo utilizan. El método propuesto está basado en el uso de un operador de conjunción adaptativo, que trabaja con valores discretos en lugar de continuos, más rápido, eficiente y con menor espacio de búsqueda que los habituales, que por sus características, permite por un lado precalcular parte del proceso, acortando el tiempo necesario para realizar la inferencia y por otro, reducir el tiempo necesario empleado en ajustar los parámetros del operador, al tener un espacio de búsqueda menor, lo que lo hace idóneo para ser usado en problemas de regresión de alta dimensionalidad.

Para analizar esta propuesta se ha llevado a cabo un extenso estudio experimental que ha sido validado con test estadísticos no paramétricos: se han utilizado diecisiete conjuntos de datos de complejidades diferentes con dos BRs iniciales distintas. En dicho estudio experimental, dividido en tres partes, se ha comparado el nuevo operador adaptativo con los operadores adaptativos más conocidos a través de los respectivos modelos multiobjetivos que hacen uso de dichos operadores. En la primera parte del estudio se ha comparado entre sí la solución más precisa alcanzada por cada modelo, en la segunda, los tiempos de ejecución, y en la tercera y última, los frentes de Pareto. Las conclusiones obtenidas han sido las siguientes:

- El modelo propuesto permite, en la mayoría de los casos, mejorar la precisión alcanzada por los modelos que usan operadores de conjunción adaptativos clásicos, obteniendo modelos con una complejidad similar a éstos.

- El nuevo modelo permite también reducir el tiempo de aprendizaje de forma importante y hace posible que pueda ser aplicado a conjuntos de datos que son imposibles de modelar con los operadores de inferencia adaptativos clásicos.

- En la mayoría de los problemas, las soluciones obtenidas por el modelo propuesto dominan a las obtenidas por las otras en casi todo el frente

del Pareto, lo que indica que para el mismo nivel de interpretabilidad obtiene soluciones más precisas.

En resumen, el nuevo modelo obtiene soluciones muy rápidamente y además sin apenas sobreaprendizaje. Esta propiedad hace al método escalable para problemas de alta dimensionalidad, para los cuales es capaz de obtener buenas soluciones.

B. Publicaciones Asociadas a la Tesis

Publicaciones en revistas internacionales:

Márquez A. A., Márquez F. A., Roldán Ana M. y Peregrín A. (2013) An efficient adaptive fuzzy inference system for complex and high dimensional regression problems in linguistic fuzzy modelling. *Knowledge-Based Systems* 54: 42–52.

Márquez A. A., Márquez F. A. y Peregrín A. (2012) A mechanism to improve the interpretability of linguistic fuzzy systems with adaptive defuzzification based on the use of a multi-objective evolutionary algorithm. *International Journal of Computational Intelligence Systems* 5(2): 297–321.

Márquez A. A., Márquez F. A. y Peregrín A. (2009) Rule base and adaptive fuzzy operators cooperative learning of Mamdani fuzzy systems with multi-objective genetic algorithms. *Evolutionary Intelligence* 2: 39–51.

Márquez A. A., Márquez F. A. y Peregrín A. (2008) Cooperation between the inference system and the rule base by using multiobjective genetic algorithms**.** *Lecture Notes in Computer Science* 5271: 739-746.

Publicaciones en congresos internacionales:

Márquez A. A., Márquez F. A. y Peregrín A. *(WCCI 2012 - FUZZ-IEEE 2012)* An efficient multi-objective evolutionary adaptive conjunction for high dimensional problems in linguistic fuzzy modelling. En *Proceedings of the IEEE International Conference on Fuzzy Systems,* páginas 1-8. Brisbane, Australia.

Márquez A. A., Márquez F. A. y Peregrín A. *(WCCI 2010 - FUZZ-IEEE 2010)* A multiobjective evolutionary algorithm with an interpretability improvement mechanism for linguistic fuzzy systems with adaptive defuzzification. En *Proceedings of the IEEE World Congress on Computational Intelligence - IEEE International Conference on Fuzzy Systems,* páginas 277-283. Barcelona, España.

Márquez A. A., Márquez F. A. y Peregrín A. *(IFSA-EUSFLAT 2009)* Rule base and inference system cooperative learning of Mamdani fuzzy systems with multiobjective genetic algorithms. En *Proceedings of the Joint 2009 International Fuzzy Systems Association World Congress and 2009 European Society of Fuzzy Logic and Technology Conference,* páginas 1045-1050. Lisboa, Portugal.

Publicaciones en congresos nacionales:

Serrano C., Márquez A. A., Márquez F. A. y Peregrín A. *(ESTYLF 2012)* Un operador de conjunción adaptativo para modelado difuso lingüístico de problemas de alta dimensionalidad. En *Proceedings of the XVI Congreso Español sobre Tecnologías y Lógica Fuzzy,* páginas 307-312. Valladolid, España.

Márquez A. A., Márquez F. A. y Peregrín A. *(ESTYLF 2010)* Introducción de un mecanismo para la mejora de la interpretabilidad en sistemas difusos lingüísticos con defuzzificación adaptativa. En *Proceedings of the XV Congreso Español sobre Tecnologías y Lógica Fuzzy,* páginas 85-90. Punta Umbría-Huelva, España.

Márquez A. A., Márquez F. A. y Peregrín A. *(ESTYLF 2008)* Aprendizaje evolutivo de sistemas difusos lingüísticos basado en la cooperación entre los operadores y las funciones de pertenencia. En *Proceedings of the XIV Congreso Español sobre Tecnologías y Lógica Fuzzy,* páginas 171-177. Cuencas Mineras, Lagreo, Asturias, España.

Márquez A. A., Márquez F. A. y Peregrín A. *(ESTYLF 2008)* Aprendizaje cooperativo de la base de reglas y el sistema de inferencia mediante algoritmos genéticos multiobjetivo. En *Proceedings of the XIV Congreso Español sobre Tecnologías y Lógica Fuzzy*, páginas 163-170. Cuencas Mineras, Lagreo, Asturias, España.

C. Líneas de Investigación Futuras

A continuación se señalan algunas líneas de trabajo futuras que quedan abiertas a partir de los resultados y conclusiones obtenidas en esta memoria.

C.1 Integrar el Aprendizaje de los Operadores Difusos Adaptativos del MI en el Aprendizaje de la BC Completa (BD y BR) con AGMOs

En la literatura especializada existen gran variedad de propuestas de diferentes modelos de aprendizaje de la BC según distintos criterios. La mayoría de ellas realizan esta tarea de dos formas: o bien aprendiendo la mejor BD a priori mediante un proceso evolutivo y posteriormente encontrando la BR mediante otro proceso de aprendizaje, o bien aprendiendo la BD y la BR mediante un proceso evolutivo de forma simultánea.

Una línea prometedora relacionada con lo expuesto consiste en estudiar la integración del aprendizaje de los ODAs del MI en los procesos multiobjetivo evolutivos de aprendizaje de la BD anteriormente descritos. En esta tesis doctoral, el modelo cooperativo evolutivo del Capítulo 2 aprende los operadores del MI al tiempo que la BR, por lo que esta línea de trabajo futuro que se propone explora la otra vía lógica consistente en combinar la adaptación del MI con cada uno de los componentes de la BC (BD y BR) de forma simultánea, para la obtención de la propia BC completa mediante AGMOs en el equilibrio entre precisión y la interpretabilidad.

C.2 Diseñar AGMOs con MI Adaptativo Eficientes para Problemas de Gran Escalabilidad

En esta memoria se ha presentado en el Capítulo 4 un modelo evolutivo multiobjetivo para abordar el aprendizaje de SBRDs compactos y precisos en problemas que presentan una alta dimensionalidad con respecto al número de variables de entrada. Otra posibilidad por la que un problema puede presentar una alta dimensionalidad es cuando contiene un número elevado de instancias o ejemplos. En este tipo de problemas el modelo evolutivo presenta algunas dificultades al tener que procesar tal cantidad de información, lo que puede afectar al equilibrio entre la interpretabilidad y la precisión del modelo

extraído. Para solucionar este problema, en la literatura especializada se ha propuesto el uso de técnicas de preprocesamiento que seleccionan un grupo reducido y significativo de instancias o ejemplos de entrenamiento, manteniendo así intactas las propiedades del problema a tratar, pero facilitando la tarea del modelo evolutivo. Por este motivo sería interesante realizar un estudio sobre el funcionamiento de los modelos evolutivos propuestos en esta memoria junto con el uso de diversas técnicas de preprocesamiento para selección de instancias en problemas que presentan este tipo de alta dimensionalidad.

C.3 Diseñar AGMOs Distribuidos con MI Adaptativo para Problemas de Alta Dimensionalidad y Gran Escalabilidad

Como complemento al modelo evolutivo presentado en el Capítulo 4 y a la línea de investigación apuntada en el apartado C.2 anterior, una línea prometedora y obvia es el uso de AGMOs que se ejecuten en paralelo que puedan ser usados en el diseño de SDEs para grandes bases de datos o para problemas con muchas variables o características poniendo especial énfasis en los aspectos de escalabilidad y eficiencia. Otra idea interesante podría ser el uso de metodologías distribuidas, dividiendo los conjuntos de datos de entrenamiento y de población en subpoblaciones, de forma que cada proceso que se ejecuta de forma distribuida pueda ajustar las reglas del submodelo utilizando su propio subconjunto de datos de entrenamiento y subconjunto de población.

Apéndices

Apéndice A

Algoritmos Evolutivos y Lógica Difusa

Este apéndice describe las líneas generales en las que se basa la Computación Evolutiva [Bäc96] puesto que todos los métodos propuestos en esta memoria están basados en el uso de AEs y más concretamente en AGMOs.

A.1. Introducción

La Computación Evolutiva se basa en el empleo de modelos que simulan algunos aspectos de la evolución de la naturaleza para el diseño e implementación de sistemas de resolución de problemas. Los distintos modelos computacionales que se han propuesto dentro de esta filosofía suelen recibir el nombre genérico de AEs [BFM97]. Existen cuatro tipos de AEs bien definidos:

- Los Algoritmos Genéticos (AGs)
- Las Estrategias de Evolución
- La Programación Evolutiva, y
- La Programación Genética.

Todos ellos tienen en común el hecho de modelar dentro de una población los procesos de reproducción, variación aleatoria, competición y selección de individuos rivales, de modo que la evolución ocurre al darse estos hechos como

en la naturaleza. De esta forma, un AE se basa en mantener una población de posibles soluciones del problema a resolver, llevar a cabo una seria de alteraciones sobre las mismas y efectuar una selección para determinar qué soluciones permanecen en generaciones futuras y cuáles son eliminadas.

Para el aprendizaje de los parámetros de las propuestas realizadas en los Capítulos 2, 3 y 4, emplearemos AGs monoobjetivo y multiobjetivo (los primeras para aquellas propuestas que utilicen un solo objetivo y los segundos para los que utilicen varios objetivos simultáneos a optimizar) cuyo funcionamiento va a ser descrito en las siguientes subsecciones.

A.2. Algoritmos Genéticos

Los AGs son algoritmos de búsqueda de propósito general que se basan en principios inspirados en la genética de las poblaciones naturales para llevar a cabo un proceso evolutivo sobre soluciones de problemas. Fueron inicialmente propuestos por Holland [Hol75] y han sido posteriormente estudiados en profundidad por otros autores [Gol89, Mic96]. Los AGs han demostrado ser, tanto desde un punto de vista teórico como práctico, una herramienta óptima para proporcionar búsqueda robusta en espacios complejos, ofreciendo un enfoque válido para solucionar problemas que requieran una búsqueda eficiente y eficaz.

Los AGs se han aplicado con mucho éxito en problemas de búsqueda y optimización. La razón de gran parte de este éxito se debe a su habilidad para explotar la información que van acumulando sobre el espacio de búsqueda que manejan, desconocido inicialmente, lo que les permite redirigir posteriormente la búsqueda hacia subespacios útiles. La *capacidad de adaptación* que presentan es su característica principal, especialmente en espacios de búsqueda grandes, complejos y con poca información disponible, en los que las técnicas clásicas de búsqueda (enumerativas, heurísticas, ...) no presentan buenos resultados.

La idea básica de estos algoritmos consiste en mantener una población de individuos que codifican soluciones del problema. Dichos individuos emplean una representación genética para codificar los valores de las características parciales que definen las distintas soluciones. Debido a ello, cada individuo recibe el nombre de *cromosoma* y cada una de sus componentes el de *gen*.

Los cromosomas se generan inicialmente a partir de la información disponible sobre el problema, o bien de un modo aleatorio cuando no se dispone de esta información, y la población se hace evolucionar a lo largo del tiempo mediante un proceso de competición y alteración controlada que emula los procesos genéticos que tienen lugar en la naturaleza hasta que se verifique una determinada condición de parada como puede ser, por ejemplo, que se haya realizado un determinado número máximo de evaluaciones de individuos o que la población haya evolucionado un determinado número de generaciones. A lo largo de sucesivas iteraciones, denominadas *generaciones*, los cromosomas se ordenan con respecto a su grado de adaptación al problema, es decir, con respecto a lo bien que resuelven dicho problema y, tomando como base estas evaluaciones, se construye una nueva población mediante un proceso de *selección* y una serie de operadores genéticos tales como el *cruce* y la *mutación*.

La mutación corresponde a una auto-replicación errónea de los individuos, mientras que el cruce intercambia material genético entre dos o más individuos ya existentes. Ambos operadores se aplican con una probabilidad definida por el usuario. Normalmente se establece una probabilidad de mutación inferior a la probabilidad de cruce, ya que una probabilidad de mutación muy elevada convertiría el proceso de búsqueda evolutiva en un proceso de búsqueda aleatoria. No obstante, la mutación es necesaria para incrementar la diversidad genética de individuos dentro de la población y para alcanzar valores de genes que no estén presentes en la población y que de otra forma serían inalcanzables, puesto que el operador de cruce solo intercambia genes (ya existentes) entre individuos.

Como en todos los AEs, es necesario diseñar una medida de calidad (*función de adaptación* o fitness) para cada problema que se desee resolver. Dado un cromosoma de la población, esta función devuelve un único valor numérico que se supone proporcional al grado de bondad de la solución que dicho cromosoma codifica. Esta función es la encargada de guiar al AG por el espacio de búsqueda. Por esta razón, debe estar bien diseñada para que sea capaz, no sólo de distinguir de un modo claro los individuos bien adaptados de los que no lo están, sino también de ordenar éstos en función de su capacidad para resolver el problema.

Los AGs realizan un proceso de búsqueda global. El hecho de trabajar con una población de soluciones candidatas en lugar de con una solución individual y utilizar operadores de cruce y mutación reduce la probabilidad de caer en un óptimo local e incremente la probabilidad de encontrar el óptimo global.

La Figura A.1, en la que P(t) denota la población en la generación t, muestra la estructura general de un AG básico.

Procedimiento AG básico

```
EMPEZAR
    t =0;
    inicializar P(t);
    evaluar P(t);
    MIENTRAS NO (condicion de parada) HACER
        EMPEZAR
            t = t + 1;
            seleccionar P'(t) a partir de P(t – 1);
            cruzar y mutar P'(t);
            evaluar P(t);
        FIN
FIN
```

Figura A.1: Estructura básica de un AG

Para aplicar un AG para resolver un problema se debe determinar:

- Una representación genética de las soluciones del problema
- Una forma de crear una población inicial de soluciones.
- Una función de evaluación que proporcione un valor de adaptación o fitness de cada cromosoma.
- Operadores que modifiquen la composición genética de la descendencia durante la reproducción.
- Valores de los parámetros utilizados (tamaño de la población, probabilidad de los operadores genéticos de cruce y mutación, etc.).

A continuación, comentaremos brevemente los aspectos básicos relacionados con los AGs, como son la representación de las soluciones, el mecanismo de selección y los operadores genéticos de cruce y mutación.

A.2.1. Representación de las Soluciones

El esquema de representación o codificación es un factor clave en la aplicación de los AGs, ya que éstos manipulan directamente una representación codificada del problema y, en consecuencia, el esquema escogido puede limitar de una forma muy severa la forma en la que el AG afronta el problema. Existen distintos esquemas generales de codificación entre los que destacan los siguientes:

1. La *codificación binaria*: Es la más antigua de todas las existentes [Gol89, Hol75]. Se basa en la representación de los cromosomas como cadenas de bits de modo que, dependiendo del problema, cada gen del cromosoma puede estar formado por una subcadena de varios bits.

2. La *codificación real*: Es más adecuada para problemas que incluyen variables definidas sobre dominios continuos [HLV98b] (con codificación binaria la longitud de los cromosomas sería excesiva), y se ha estudiado ampliamente en los últimos años. En este esquema de representación, cada variable del problema se asocia a un único gen que toma un valor real dentro del intervalo especificado, por lo que no existen diferencias entre el genotipo (la codificación empleada) y el fenotipo (la propia solución codificada). Gracias a esta propiedad se solucionan los problemas comentados.

3. La *codificación basada en orden*: Este esquema está diseñado específicamente para problemas de optimización combinatoria en los que las soluciones son permutaciones de un conjunto de elementos determinado [Gol89, Mic96]. Como ejemplos de este tipo de problemas podemos citar los conocidos problemas del viajante de comercio y del coloreado de grafos.

Por otro lado, la función de evaluación desempeña, al igual que el esquema de codificación, un papel determinante en el AG al guiar el proceso evolutivo dentro del espacio de búsqueda. Esta función debe estar bien diseñada para ser capaz no sólo de distinguir individuos bien adaptados, sino también de ordenarlos en función de su capacidad para resolver el problema.

A.2.2. El Mecanismo de Selección

El mecanismo de selección es el encargado de seleccionar la población intermedia de individuos la cual, una vez aplicados los operadores de cruce y mutación, formará la nueva población del AG en la siguiente generación. De este modo, si notamos por *P* la población actual formada por n cromosomas, $C_1,\ldots,C_n$, el mecanismo de selección se encarga de obtener una población intermedia *P'*, formada por copias de los cromosomas de *P* (véase la Figura A.2). El número de veces que se copia cada cromosoma depende de su adecuación, por lo que generalmente aquellos que presentan un valor mayor en la función de adaptación suelen tener más oportunidades para contribuir con copias a la formación de *P'*.

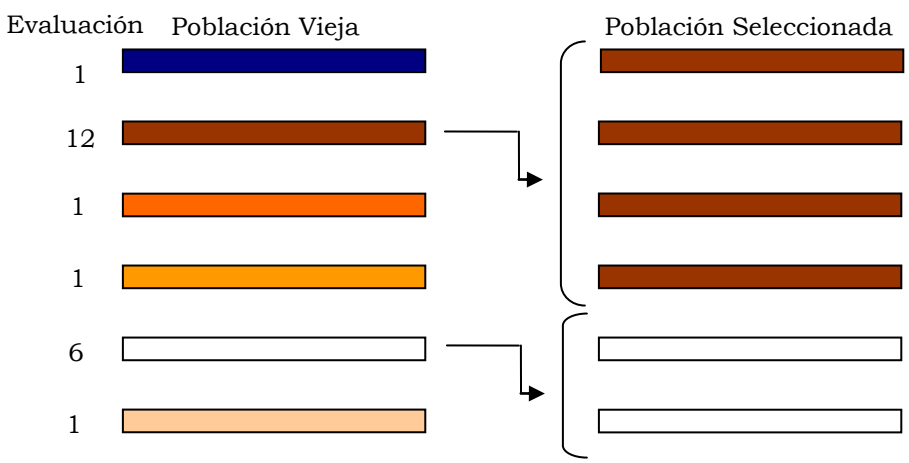


Figura A.2: Ejemplo de aplicación del mecanismo de selección

Existen diferentes formas de poner en práctica la selección [BS91], como la selección por torneo [MG95], o la selección proporcional [Hol75] en la que, estableciendo un paralelismo entre la población y una ruleta, cada cromosoma está representado por un sector de la misma cuyo tamaño es proporcional a la adaptación de dicho cromosoma. Los cromosomas se seleccionan girando la ruleta tantas veces como individuos tengamos que seleccionar para formar la población intermedia. Una de las más eficientes es la selección por ranking [Bak87], en el cual el número de copias de cada individuo en la población intermedia está acotado inferior y superiormente por un número de copias esperado calculado en función de su adaptación.

El mecanismo de selección puede ser complementado por el *modelo de selección elitista*, basado en mantener un número determinado de los individuos mejor adaptados de la población anterior en la nueva población (la obtenida después de llevar a cabo el proceso de selección y de aplicar los operadores de cruce y mutación) [Gol89, Mic96].

A.2.3. El Operador de Cruce

Este operador constituye un mecanismo para compartir información entre cromosomas. Combina las características de dos cromosomas padre para obtener dos descendientes, con la posibilidad de que los cromosomas hijo, obtenidos mediante la recombinación de sus padres, estén mejor adaptados que éstos. No suele ser aplicado a todas las parejas de cromosomas de la población intermedia sino que se lleva a cabo una selección aleatoria en función de una determinada probabilidad de aplicación, la *probabilidad de cruce, Pc*.

El operador de cruce juega un papel fundamental en el AG. Su tarea es la de *explotar el espacio de búsqueda* refinando las soluciones obtenidas hasta el momento mediante la combinación de las buenas características que presenten. Como ya hemos comentado, tanto la definición del operador de cruce como la del operador de mutación dependen directamente del tipo de representación empleada. Por ejemplo, trabajando con el esquema de codificación binario, se suele emplear el clásico *cruce simple* en un punto, basado en seleccionar aleatoriamente un punto de cruce e intercambiar el código genético de los dos cromosomas padre a partir de dicho punto para formar los dos hijos (véase Figura A.3), o el *cruce multipunto*, que procede como el anterior pero trabajando sobre dos o más puntos de cruce.

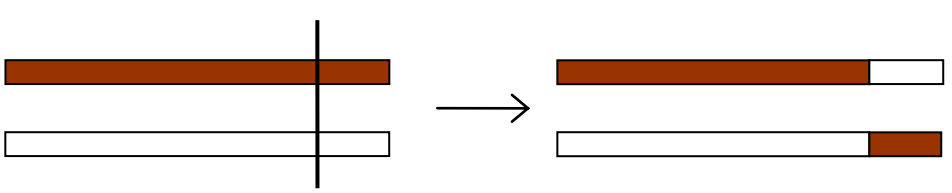


Figura A.3: Ejemplo de aplicación del operador de cruce simple en un punto

También se pueden emplear ambos operadores cuando se trabaja con el esquema de codificación real, aunque existe una serie de operadores diseñados para su uso específico con esta representación [HLV98b]. Entre éstos, destacaremos una familia de operadores que manejan técnicas basadas en Lógica Difusa para mejorar el comportamiento del operador de cruce [HLV97]. Como ejemplo de estos operadores, introduciremos el operador de cruce max-min-aritmético. Dados dos cromosomas de la población $P(t)$, $C_v^t = (c_1,\ldots,c_k,\ldots,c_H)$, y $C_W^t = (c'_1,\ldots,c'_k,\ldots,c'_H)$, que van a ser cruzados, este operador genera los cuatro descendientes siguientes:

$$C_1^{t+1} = a\, C_w^t + (1-a)\, C_v^t$$

$$C_2^{t+1} = a\, C_w^t + (1-a)\, C_w^t$$

$$C_3^{t+1} \text{ con } C_{3k}^{t+1} = \min \{c_k, c'_k\}$$

$$C_4^{t+1} \text{ con } C_{4k}^{t+1} = \max \{c_k, c'_k\}$$

y escoge los dos mejor adaptados para formar parte de la nueva población. El parámetro a empleado en los dos primeros puede definirse como constante a lo largo de toda la ejecución del AG o variable dependiendo de la edad de la población.

A.2.4. El Operador de Mutación

Este segundo operador altera arbitrariamente uno o más genes del cromosoma seleccionado con el propósito de aumentar la diversidad de la población. Los genes son alterados de acuerdo a una probabilidad de mutación P_m establecida para todos ellos. La propiedad de búsqueda asociada al operador de mutación es la *exploración*, ya que la alteración aleatoria de una de las componentes del código genético de un individuo suele conllevar el salto a otra zona del espacio de búsqueda que puede resultar más prometedora.

El operador de mutación clásicamente empleado en los AGs con codificación binaria se basa en cambiar el valor del bit seleccionado para mutar por su complementario en el alfabeto binario, tal y como recoge la Figura A.4.

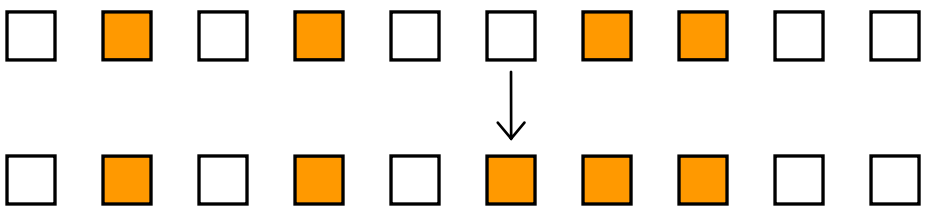


Figura A.4: Ejemplo de aplicación del operador de mutación.

Este operador puede trasladarse al campo de los AGs con codificación real, de forma que el nuevo valor del gen mutado se escoja aleatoriamente dentro del intervalo de definición asociado. Al igual que en el caso del operador de cruce, existen distintos operadores de mutación específicos para trabajar con esta codificación [HLV98b, Mic96].

A.3. Algoritmos Genéticos Multiobjetivo

La mayor parte de los problemas de optimización del mundo real son de naturaleza multiobjetivo. Esto es, suelen tener dos o más funciones objetivo que deben satisfacerse simultáneamente y que posiblemente están en conflicto entre sí.

La presencia de múltiples objetivos en un problema provoca que no exista una solución óptima única, sino que exista un conjunto de soluciones óptimas (conocidas como soluciones Pareto óptimas o soluciones en la frontera del Pareto), mayores cuantos más objetivos haya, debido a que ninguna de ellas domina a las demás soluciones en todas las funciones objetivo a optimizar. Por ello, a falta de cualquier otra información adicional, ninguna de estas soluciones Pareto óptimas puede ser considerada mejor que otra por lo que en principio interesa encontrar tantas soluciones Pareto óptimas como sea posible.

Los AGMOs [Deb01] surgen como una extensión de los AEs para problemas que obtienen más de una solución y, al igual que éstos, basan su funcionamiento en la simulación del proceso de evolución natural donde cada

individuo de la población representa una solución potencial del problema, codificada de acuerdo a un esquema de representación basado en números enteros o reales. Inicialmente la población se genera de forma aleatoria y luego evoluciona mediante la aplicación iterativa de operadores de reproducción (cruce y mutación). Esta evolución es guiada por una estrategia de selección de los individuos más adaptados a la resolución del problema, de acuerdo a sus valores de fitness.

Un AGMO debe diseñarse para lograr dos propósitos en forma simultánea: lograr buenas aproximaciones al frente del Pareto y mantener la diversidad de las soluciones, muestreando adecuadamente el espacio de soluciones. El mecanismo evolutivo de los AEs permite lograr el primer propósito, mientras que para preservar la diversidad los AGMOs utilizan las técnicas de nichos, sharing, crowding o similares, utilizadas tradicionalmente por los AE en la optimización de funciones multimodales.

Aunque las primeras sugerencias que tuvieron en cuenta la posibilidad de usar AEs para resolver problemas multiobjetivo fueron hechas por Rosenberg en 1967 [Ros67], realmente fue David Schaffer [Sch85] el primero en diseñar un AGMO, el llamado Vector Evaluated Genetic Algorithm (VEGA). Después de VEGA, los investigadores diseñaron una primera generación de AGMOs que se caracterizaban por su simplicidad. Estos AGMOs de primera generación combinaban un buen mecanismo para seleccionar los individuos no dominados junto con un buen mecanismo para mantener y preservar la diversidad, pero presentaban un problema fundamental: no incorporaban elitismo explícitamente. Por ello, una segunda generación de AGMOs comenzó cuando el elitismo llegó a ser un mecanismo estándar tras ser propuesto en SPEA [ZT99]. De hecho, el uso de elitismo es un requerimiento teórico para garantizar la convergencia de un AGMO. La Tabla A.1 muestra un resumen de los más representativos AGMOs de ambas generaciones.

Aunque pocos de estos AGMOs de segunda generación han sido adoptados como referencia o usados por otros, la mayoría de los investigadores coinciden en afirmar que, de todos ellos, los AGMOs SPEA2 y NSGA-II pueden ser considerados como los más representativos, ya que permite obtener un conjunto de soluciones del Pareto más amplio en una gran variedad de problemas, propiedad muy apreciada en este marco de trabajo.

Tabla A.1: Clasificación de los AGMOs

Referencia	**AGMO**	**1ª Generación**	**2ª Generación**
[FF93]	MOGA	✓	
[HNG94]	NPGA	✓	
[SD94]	NSGA	✓	
[CT01]	micro-GA		✓
[EMH01]	NPGA 2		✓
[DAPM02]	NSGA-II		✓
[KC00]	PAES		✓
[CKO00, CJKO01]	PESA & PESA-II		✓
[ZT99, ZLT01]	SPEA & SPEA2		✓

A.3.1. SPEA2

El algoritmo SPEA2 [ZLT01] (Strength Pareto Evolutionary Algorithm for multiobjective optimization) es una de las técnicas más utilizadas en la resolución de problemas multiobjetivo. Este algoritmo fue diseñado para superar los problemas que presentaba su predecesor, el algoritmo SPEA [ZT99]. Éste se diferencia de otros AGMO en varios aspectos, entre los cuales hay tres de gran importancia:

- Incorpora una estrategia de asignación del fitness que tienen en cuenta tanto las soluciones dominantes como las soluciones dominadas para cada individuo, es decir, considera para cada individuo, el número de individuos que domina y el número de individuos por los cuales es dominado.

- Utiliza una función de densidad que se calcula mediante el empleo de la técnica del vecino más cercano, dirigiendo la búsqueda de forma más eficiente.

- Utiliza un método de truncamiento de archivo mejorado que garantiza la preservación y la diversidad de las soluciones del Pareto.

Según la descripción de los autores en [ZLT01] el bucle principal de SPEA2 consta de los siguientes pasos:

Entrada: N *(tamaño de la población),*
$\overline{N}$ *(tamaño de la población externa),*
T *(máximo número de generaciones).*
Salida: A *(conjunto de no dominados).*

1. ***Inicialización:*** *Crear la población inicial* P_0 *y una población externa vacía* $\overline{P_0} = \emptyset$;. *Establecer* $t = 0$.
2. ***Asignación de fitness:*** *Calcular el fitness de los individuos de* P_t *y* $\overline{P_t}$.
3. ***Selección:*** *Copiar los individuos no dominados de* $P_t \cup \overline{P_t}$ *en* $\overline{P_{t+1}}$.
   Si $|\overline{P_{t+1}}| > \overline{N}$ *se aplica el operador de truncamiento.*
   Si $|\overline{P_{t+1}}| < \overline{N}$ *se rellena con dominados de* $P_t \cup \overline{P_t}$.
4. ***Terminación:*** *Si* $t \geq T$, *se devuelve A y acaba la ejecución.*
5. ***Selección Padres:*** *Aplicar torneo binario con reemplazamiento en* $\overline{P_{t+1}}$ *hasta completar la población de padres.*
6. ***Cruce y Mutación:*** *Aplicar cruce y mutación para construir* P_{t+1}.
   Volver al Paso 2 con $t = t + 1$.

El algoritmo SPEA2 utiliza una población y un tamaño de archivo fijo. La población forma la base de las posibles nuevas soluciones, mientras que el archivo contiene las actuales soluciones. El archivo se construye y actualiza en cada iteración del algoritmo, copiando en él todos (si caben) los individuos no dominados existentes tanto en el archivo como en la población. Si el número de no dominados existente difiere del tamaño del archivo, algunos individuos serán añadidos o eliminados según sea necesario. Así, si el número de no dominados es menor que el tamaño del archivo, los mejores individuos dominados serán añadidos, mientras que si el número de no dominados es mayor se eliminan, mediante la aplicación de una heurística de clustering en el espacio de los objetivos, aquellos no dominados que estén más cercanos unos de otros, con idea de asegurar que en el archivo exista una representación de las distintas partes del espacio objetivo.

Una vez tenemos formado la nueva población externa, de ésta se selecciona mediante torneo binario la nueva población de padres que formará la nueva población mediante su cruce y mutación.

Este proceso se repite de forma reiterada hasta que se alcance el número de generaciones establecidas.

En la siguiente figura podemos ver una representación de la técnica de truncamiento usada en SPEA2. La figura de la izquierda representa un conjunto de no dominados, mientras que la figura de la derecha muestra qué no dominados y en qué orden son eliminados mediante el operador de truncamiento suponiendo que el tamaño de la población externa $\overline{N}$ = 5.

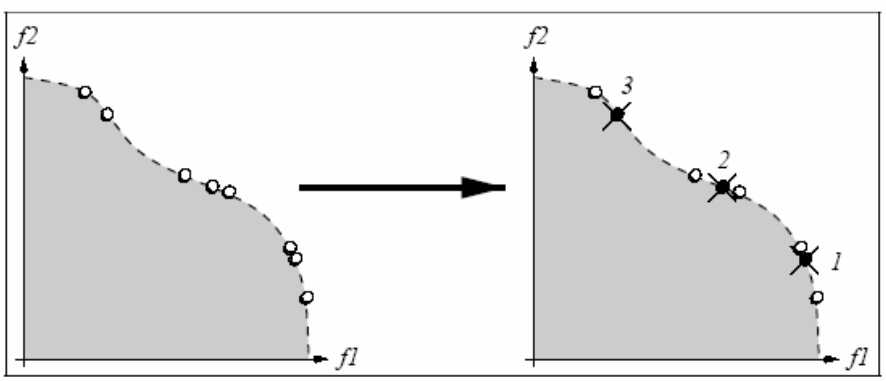


Figura A.5: Ilustración del método de truncamiento de archivo usado en SPEA2

A la hora de asignar un valor fitness a cada individuo de la población y determinar que individuos pertenecen a la frontera del Pareto (aquellos que no son dominados en todos los objetivos por el resto de individuos de la población) y cuáles no, el método de asignación de fitness utilizado en SPEA2 tiene en cuenta para cada individuo el número de soluciones que domina y el número de soluciones por las que es dominado. Además incorpora información sobre la densidad para discriminar entre individuos a los que una vez calculado el número de soluciones que domina y el número de soluciones por las que es dominado obtienen un mismo valor.

En concreto, suponiendo que *Pt* y $\overline{Pt}$ denotan los individuos de la población y del archivo respectivamente, a cada individuo i en $Pt \cup \overline{Pt}$ se le asigna un valor *S(i),* que representa el número de soluciones que domina

$$S(i) = \left| \left\{ j \mid j \in P_t \cup \overline{P_t} \wedge i \succ j \right\} \right| \qquad \text{(A.3)}$$

donde $|\cdot|$ representa la cardinalidad del conjunto, *U* representa la unión y el símbolo $\succ$ corresponde a la relación de dominancia del Pareto, que para el caso concreto de problemas de minimización de objetivos se define como:

$$a \succ b \leftrightarrow \forall i = 1,2,...,M \;\; f_i(a) \leq f_i(b) \wedge \exists\, i \mid f_i(a) < f_i(b) \qquad \text{(A.4)}$$

donde *a, b* representan dos individuos y *M* es el número de objetivos.

En este sentido, decimos que un individuo a domina a otro individuo b si es mejor o igual en todos los objetivos y al menos estrictamente mejor en uno de ellos.

Basado en el valor *S(i)*, a cada individuo *i* se le asigna un valor previo de fitness *R(i)* igual a:

$$R(i) = \sum_{j \in P_t \cup \overline{P_t},\, j \succ i} S(j) \qquad \text{(A.5)}$$

Es importante resaltar que el fitness es minimizado aquí, por lo que *R(i)*=0 corresponde a los individuos no dominados, mientras que un alto valor de *R(i)* denota que *i* es dominado por muchos individuos (los cuales a su vez dominan a muchos otros individuos).

Para poder discriminar entre aquellos individuos que tiene un idéntico valor inicial de fitness *R(i)*, el valor final del fitness es recalculado añadiendo un valor de densidad *D(i)*, estimado en el espacio de cada objetivo de acuerdo a la siguiente fórmula:

$$D(i) = \frac{1}{\delta_i^k + 2} \qquad \text{(A.6)}$$

donde δ_i^k representa la k-ésima distancia más cercana para el individuo $i \in P_t \cup \overline{P_t}$ en cada espacio objetivo. El 2 es añadido para garantizar que el valor del denominador nunca sea 0 y para que el valor devuelto por *D(i)* < 1.

Para realizar esto, para cada individuo *i* se calcula las distancias (en cada espacio objetivo) a todos los individuos *j* que hay en la población y en el archivo conjuntamente. Estas distancias se ordenan en orden ascendente y se escoge la que ocupa el k-ésimo lugar.

El valor normalmente usado para k es la parte entera de la siguiente expresión:

$$k = \sqrt{N + \overline{N}} \tag{A.7}$$

donde N y $\bar{N}$ representan el tamaño de la población y del archivo respectivamente. Finalmente, el valor de fitness asignado a cada individuo es calculado como:

$$F(i) = R(i) + D(i) \tag{A.8}$$

Con ello, las mejores soluciones serán aquellas que tengan asignado un menor valor de fitness *F(i)*, de forma que los individuos no dominados serán todos aquellos cuyo valor *F(i)* < 1.

A.3.2. NSGA-II

El algoritmo NSGA-II [DAPM02] es otro de los AGMOs más conocidos y frecuentemente utilizados en la literatura para la resolución de problemas multiobjetivo. Como en otros AEs, primeramente el NSGA-II genera una población inicial. La población descendiente se genera desde la población actual a través de la selección, el cruce y la mutación. La siguiente generación se construye desde la población actual y la descendiente. Este proceso se realiza de forma reiterada hasta que se cumpla la condición de parada.

El NSGA-II posee varias características que lo hacen uno de los principales y más importantes AGMO:

- Una selección de torneo binario basado en una rápida ordenación de las soluciones no dominadas.

- Un método de distancia de aglomeración para estimar la diversidad de una solución. Para ello incorpora una estrategia de asignación de la aptitud (fitness) basada en el Ranking del Pareto y en el operador de hacinamiento (crowding), el cual ordena los individuos de la población ascendentemente según su ranking del Pareto y, en caso de empate, en orden descendente según la distancia determinada por el operador de crowding.

- Utiliza elitismo en el procedimiento de actualización de cada generación, lo que permite que no se pierdan las mejores soluciones que se van generando a lo largo del proceso de convergencia del algoritmo hacia el conjunto óptimo del Pareto.

Según la descripción de los autores en [DAPM02] el bucle principal de NSGA-II consta de los siguientes pasos:

Entrada: N *(tamaño de la población),*
T *(máximo número de generaciones).*

Salida: A *(conjunto de no dominados).*

$R_t = P_t \cup Q_t$	*1. Una población combinada R_t se forma con la población inicial de la Población padre P_t y la población descendiente Q_t (inicialmente vacía).*
$F = ordenar\_fronteras(R_t)$	*2. Generar todas las fronteras de soluciones no dominadas $F = (F_1, F_2, ...)$ de R_t.*
$P_{t+1} = 0$ e $i = 1$	*3. Inicializar $P_{t+1} = 0$ e $i = 1$.*
mientras $(\|P_{t+1}\| + \|F_i\| \le N)$	*4. Repetir hasta que la solución padre sea completada.*
$distancia\_crowding(F_i)$	*5. Calcular la distancia de crowding en Fi.*
$P_{t+1} = P_{t+1} \cup F_i$	*6. Incluir i-th individuos no-dominados en la población padre.*
$i = i + 1$	*7. Comprobar la siguiente frontera para su inclusión.*
$ordenar\ (F_i, \prec_i)$	*8. Ordenar en orden descendiente usando el operador de crowding para comparar.*
$P_{t+1} = P_{t+1} \cup F_i[1:(N-\|P_{t+1}\|)]$	*9. Elegir los primeros $(N - \| P_{t+1} \|)$ elementos de F_i.*
$Q_{t+1} = nueva\_poblacion(P_{t+1})$	*10. Aplicar cruce y mutación para crear la nueva población Q_{t+1}.*
$t = t + 1$	*11. Incrementar el contador de generación.*

Inicialmente se crea de forma aleatoria una población padre P_0 y una población descendente Q_0 inicialmente vacía. En cada iteración del algoritmo, una población combinada actual $R_t = P_t \cup Q_t$ se forma con la población padre y la población descendiente. De esta forma se asegura el elitismo, al incluir en la población combinada actual tanto los individuos previos (población padre) como los nuevos individuos (población descendiente). Cada solución de la población combinada actual es evaluada, asignándole un valor de Ranking del Pareto de la siguiente manera:

1. En primer lugar se le asigna a todos los individuos no dominados de la población actual un valor de Ranking igual a 1. Todos los individuos con Ranking 1 son provisionalmente eliminados de la población actual.

2. A continuación se le asigna un valor de Ranking igual a 2 a los individuos no dominados de esta población actual reducida (a la que provisionalmente se les ha eliminado los individuos con Ranking 1). Al igual que antes, todos los individuos con Ranking 2 son provisionalmente eliminados de esta población actual reducida.

3. Este proceso se repite de forma reiterada hasta que todos los individuos son provisionalmente eliminados de la población actual.

Como resultado de este proceso, cada individuo de la población combinada actual tendrá asignado un Ranking del Pareto, de forma que aquellos individuos con un valor de Ranking menor serán mejores que aquellos con un valor de Ranking mayor, escogiéndose por tanto, para formar la próxima generación, aquellos individuos con un menor valor de Ranking.

Para poder discriminar entre aquellos individuos con igual valor de Ranking y para preservar la diversidad en la población con idea de asegurar la existencia de una representación de las distintas partes del espacio objetivo, los individuos con un mismo valor de Ranking son ordenados utilizando un operador de crowding, el cual calcula para cada individuo la distancia entre sus soluciones adyacentes con el mismo ranking en el espacio objetivo. De esta manera se calcula una estimación de la densidad de las soluciones, de forma que los individuos que presentan mayor diversidad son aquellos con un valor de operador de crowding mayor y los que presentan menor diversidad son aquellos con un valor pequeño. Los individuos mejores serán por tanto aquellos con menor densidad (mayor diversidad), aquellos con un valor de operador de crowding mayor.

De esta forma el valor final de fitness asignado a cada individuo (basado en el ranking del Pareto y en el operador de crowding) ordena los individuos de la población ascendentemente según su ranking del Pareto y, caso de empate, en orden descendente según la distancia determinada por el operador de crowding:

$$i \prec_n j \text{ si } (i_{rank} < j_{rank}) \text{ or } ((i_{rank} = j_{rank}) \text{ and } (i_{distancia} > j_{distancia})) \quad \text{(A.9)}$$

En resumen, entre dos soluciones con diferente ranking, escogemos aquella con un menor valor de ranking, y si ambas soluciones pertenecen a la misma frontera escogemos aquella solución que presenta mayor diversidad, aquella que está en una región menos densa.

Una vez tenemos los individuos de la población combinada actual ordenados según su ranking de Pareto y su operador de crowding, seleccionamos los N mejores para formar la nueva población de padres. Para ello en primer lugar se escogen los individuos con Ranking 1. Si el número de individuos con Ranking 1 es menor que N, se añaden todos ellos a la nueva población de padres. Los restantes miembros que forman la nueva población de padres son elegidos de entre las siguientes fronteras según su ranking. Así todos los individuos con Ranking 2 son elegidos, seguidos de los de Ranking 3, y así sucesivamente. Este proceso se repite de forma reiterada hasta que la población de padres sea completada. En caso que todos los individuos de una determinada frontera no puedan ser añadidos, éstos se ordenan mediante el operador de crowding y se añaden aquellos con un valor mayor.

En la siguiente figura podemos ver una representación de la técnica de ordenación usada en NSGA-II para seleccionar la nueva población. La figura de la izquierda representa la población combinada actual R_t, formada por la actual población padre P_t y la población descendente Q_t. Los individuos de la población combinada son ordenados por Ranking de Frontera, como se ilustra en la figura central. Finalmente la nueva población padre se forma con los N mejores individuos de esta población combinada, como se ilustra en la figura de la derecha. Este proceso como podemos comprobar garantiza el elitismo al quedarnos siempre con las mejores soluciones de la población combinada.

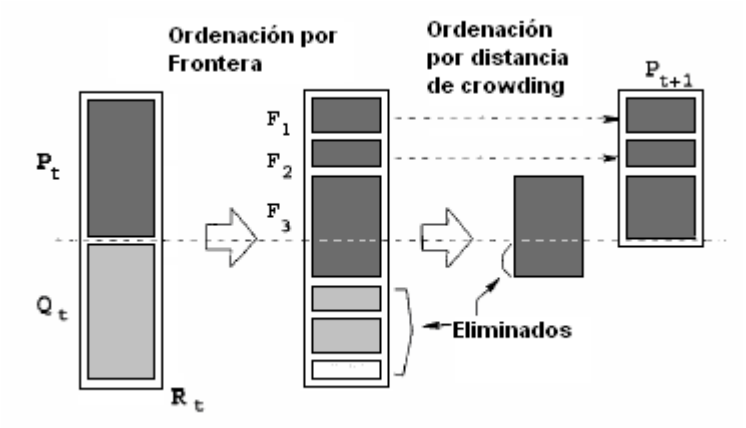


Figura A.6: Ilustración del método de ordenación basado en el Ranking de Pareto y operador de crowding usado en NSGA-II para seleccionar la nueva población

Como podemos observar en la figura, todos los individuos pertenecientes a la Frontera 1 y 2 han sido añadidos a la nueva población de padres P_{t+1}. En cambio, al no caber la totalidad de los individuos de la Frontera 3, éstos son ordenados por la distancia de crowding, añadiéndose a P_{t+1} los mejores, aquellos con mayor distancia de crowding.

Una vez tenemos la nueva población de padres P_{t+1} formada, se selecciona mediante torneo binario, basado en el Ranking de Pareto y operador de crowding, los N individuos que serán combinados y mutados para formar la nueva población descendente Q_{t+1}. La nueva población combinada será la resultante de la unión de esta nueva población de padres y de descendientes. Cada solución de esta nueva población combinada es evaluada y ordenada de la misma manera que antes, usando el Ranking de Pareto y el operador de crowding, repitiéndose todo el proceso anterior hasta que se alcance el número de generaciones deseado.

A.4. Lógica Difusa

La Lógica Difusa [Zad65] permite modelar conocimiento impreciso y cuantitativo, así como transmitir y manejar incertidumbre y soportar el razonamiento humano de una forma natural. La lógica difusa se ha aplicado

en múltiples área de investigación, fundamentalmente por su cercanía al razonamiento humano y por proporcionar una forma efectiva para capturar la naturaleza aproximada e inexacta del mundo real.

La forma de representación del conocimiento ha sido una de las áreas de mayor interés investigadas en IA. Uno de los principales aspectos es la representación de conocimiento lingüísticamente impreciso, para lo que se han demostrado ineficaces las técnicas convencionales, motivo por el cual la lógica difusa ha sido desarrollada.

La lógica difusa se puede considerar una extensión de la lógica clásica, en la que se incorporan nuevos conceptos para trabajar con valores imprecisos. La diferencia fundamental entre la lógica difusa y la clásica está en el rango de valores de verdad. Mientras que en las proposiciones clásicas sólo existen dos posibles valores de verdad (verdadero o falso), el grado de verdad o falsedad de las proposiciones difusas puede tomar distintos valores numéricos.

La lógica difusa se basa en el concepto de conjunto difuso, que puede definirse como una generalización de los conjuntos clásicos. Mientras que en los clásicos, la función de pertenencia sólo puede tomar dos valores (0 o 1, pertenencia o no pertenencia), en los conjuntos difusos, la función de pertenencia asigna a cada elemento un grado de pertenencia dentro del intervalo [0, 1]. Esto permite representar conceptos con límites borrosos, pero cuyo significado sí está definido de forma completa e imprecisa. De esta forma, los conjuntos difusos permiten una representación con mayor precisión para conceptos reales. Esta expresividad permite simplificar las reglas y los sistemas basado en ellos.

Una partición difusa de una variable determina una distinción de niveles en los valores de la misma mediante conjuntos difusos y permite solapamiento en las fronteras, al igual que ocurre en el razonamiento humano cuando trabajamos con valores lingüísticos [Zad75]. Así una variable se describe mediante distintos términos lingüísticos de los que se utilizan habitualmente en el razonamiento humano con variables numéricas (por ejemplo bajo, medio y alto). Cuando utilizamos estos conceptos, mentalmente pensamos en ellos con un cierto solapamiento entre algunos términos. De ahí que, cuando decidimos trabajar con una variable considerándola una variable lingüística que toma como valores términos lingüísticos, definimos una partición difusa que considere siempre una división del dominio en términos lingüísticos con cierto nivel de solapamiento. El significado de cada término viene especificado por un conjunto difuso y por tanto por una función de pertenencia que

determinará, de forma precisa, el grado de pertenencia al conjunto difuso correspondiente para cualquier valor de la variable.

Los sistemas basado en reglas que utilizan conjuntos difusos para describir los valores de sus variables se denominan SBRDs. Ante una situación dada, la regla se podrá aplicar si el antecedente de la misma describe una zona del espacio a la que pertenece el ejemplo, es decir, si el grado de compatibilidad del antecedente de la regla y el ejemplo es mayor que cero. Cuando el antecedente de una regla difusa está formado por una conjunción de condiciones para las distintas variables, dicho grado de compatibilidad se calcula aplicando un operador de intersección difusa (una t-norma) a los grados de pertenencia de cada variable a su conjunto difuso correspondiente. Si utilizamos la t-norma del mínimo o el producto, cuando una de las variables no pertenece al conjunto difuso implicado, el grado de compatibilidad del ejemplo con la regla es cero.

Las reglas difusas se pueden considerar modelos locales simples, lingüísticamente interpretables y con un rango de aplicación muy amplio. Permiten la incorporación de toda la información disponible en el modelado de sistemas, tanto de la que proviene de expertos humanos, como de las que tiene su origen en medidas empíricas y modelos matemáticos.

Apéndice B

Métodos de Aprendizaje de la Base de Reglas

Este apéndice describe de forma detallada los métodos de aprendizaje de BR basado en ejemplos utilizados en esta memoria: WM y COR y que han sido utilizados en los distintos estudios experimentales.

B.1. Método WM

El algoritmo WM es una técnica de aprendizaje de reglas lingüísticas con envoltura de ejemplos. En este algoritmo previamente se considera la definición del conjunto de términos lingüísticos que componen la entrada y la salida primaria de las particiones difusas. Esta información se puede obtener bien de un experto (si fuera posible) o bien mediante un proceso de normalización.

Las reglas lingüísticas se generan mediante la comprobación de algún criterio de cubrimiento sobre los datos que tengamos como ejemplos (de ahí el nombre de "*con envoltura de ejemplos*"). El mecanismo de aprendizaje consiste en dar un grado de importancia a cada regla de la BR sobre su cubrimiento y al final seleccionar la regla con más alto grado de importancia para cada grupo, es decir, para cada combinación de antecedentes.

La generación de la BR se efectúa mediante los cuatro pasos siguientes:

1. Considerar una partición difusa del espacio de variables del sistema.

2. Generar un conjunto preliminar de reglas lingüísticas – Este conjunto estará formado por la regla que mejor cubra a cada ejemplo contenidos en el conjunto de datos de entrada-salida. La estructura de cada regla, CR^l, se obtiene tomando un ejemplo concreto, e_l, y asignando a cada variable lingüística la etiqueta existente del conjunto de términos de la misma que tenga asociado el conjunto difuso con el componente del vector que tenga un mayor grado de pertenencia., $(A^l_1,\ldots,A^l_n,B^l)$, con $A^l_i \in A_i$ y $B^l \in B$.

3. Asignar un grado de importancia a cada regla: Sea la regla CR^l = *SI* X_1 *es* A^l_1 *y … y* X_n *es* A^l_n *ENTONCES Y es* B^l , la regla lingüística generada por el ejemplo e_l, el grado de importancia asociada a la misma se obtendrá computando el valor de cubrimiento de la regla sobre el ejemplo correspondiente de la siguiente forma:

$$CV_{IT}(CR^l, e_l) = \mu A^l_1(x^l_1)\cdot\ldots\cdot \mu A^l_n(x^l_n)\cdot \mu B^l_1(y^l).$$

4. Construir una BR final a partir de las reglas candidatas – Para construir dicha BR final se procede de la siguiente forma de acuerdo a sus antecedentes:

   - En primer lugar, todas las reglas existentes en el conjunto preliminar que presenten la misma combinación de antecedentes y tengan asociado el mismo consecuente, serán automáticamente insertadas (una sola vez) en la BR final.

   - En cambio, en el caso en que existan reglas conflictivas, es decir, reglas con el mismo antecedente y distintos valores en el consecuente, la regla insertada será aquella que presente mayor grado de importancia.

B.2. Método COR

Un método guiado mediante ejemplos para la generación de reglas lingüísticas difusas, generalmente busca las reglas lingüísticas con mejor

ejecución individual (ejemplo, el mencionado método WM [WM92]) y la interacción global entre las reglas de la BR no es considerada, obteniéndose así BC las cuales pueden ser mejoradas.

Partiendo de esta premisa y guardando las ventajas de los métodos de generación de BR con envoltura de ejemplos, la metodología COR fue propuesta en [CCH02]. Esta metodología está basada en una búsqueda combinatoria de reglas cooperativas sobre un conjunto de reglas candidatas para encontrar el mejor conjunto de reglas que cooperen. Concretamente, a diferencia del método WM, en vez de seleccionar el consecuente con mayor importancia en cada subespacio como es habitual, la metodología COR considera la posibilidad de usar otros consecuentes, diferentes del mejor, permitiendo obtener SBRD con más precisión gracias a tomar la BR con mejor cooperación. Para este propósito, COR hace una búsqueda combinatorial entre las reglas candidatas buscando el conjunto de consecuentes, los cuales globalmente permiten mayor precisión.

La metodología COR consta de dos fases:

a) Construcción del espacio de Búsqueda– Esta obtiene un conjunto de consecuentes candidatos para cada regla. Obtener los antecedentes R^i_{ant} de las reglas y un conjunto de consecuentes candidatos CR^i_{ant} para cada antecedente.

   Para la obtención de antecedentes y consecuentes candidatos, la metodología se basa en el uso de mecanismos de envoltura de ejemplos.

b) Selección del conjunto de reglas que mejor cooperen.- En esta fase se hace una búsqueda combinatorial entre estos conjuntos CR^i_{ant}, buscando la combinación de consecuentes con mejor precisión global.

Con el fin de realizar esta búsqueda combinatoria se pueden considerar una búsqueda exhaustiva o una técnica de búsqueda heurística. La búsqueda exhaustiva asegura que se encuentre la solución óptima, pero esta puede ser costosa o irrealizable en tiempo de ejecución cuando haya un gran número de combinaciones. Por este motivo, se utilizará una técnica de búsqueda heurística, por ser eficaz y rápida como es el caso de los AEs.

Una descripción del proceso de generación de reglas basado en COR lo podemos ver en los siguientes pasos:

Entradas:

- *Un conjunto de datos de entrada-salida* -- $E = \{ e_1, e_2, e_3, \ldots\ldots, e_M \}$, con $e_l = (x^l_1,\ldots\ldots, x^l_m, y^l_1,\ldots\ldots, y^l_p)$, $l \in \{1,\ldots\ldots,N\}$, siendo N el tamaño del conjunto de datos, y n (m) el número de variables de entrada (salida) – representando el comportamiento del problema a solucionar.

- *Una partición difusa del espacio de variables.* En nuestro caso, tenemos conjuntos difusos uniformemente distribuidos. Sea A_i el conjunto de términos lingüísticos de la *i-ésima* variable de entrada, con $i \in \{1,\ldots\ldots,m\}$, y B_j el conjunto de términos lingüísticos de la *j-ésima* variable de salida , con $j \in \{1,\ldots\ldots,p\}$, con $|A_i|$ $(|B_j|)$ siendo el número de etiquetas de la i-ésima (j-ésima) variable de entrada (salida).

Algoritmo:

1. *Construcción del espacio de búsqueda:*

   1.1. *Definir el subespacio de entrada difuso que contiene los ejemplos positivos:* Para hacer esto, deberíamos definir el conjunto de ejemplos positivos $(E^+ (S_s))$ para cada subespacio de entrada $S_s = (A^s_1, \ldots\ldots,A^s_i,\ldots\ldots,A^s_n)$, con $A^s_i \in A_i$ siendo $s \in \{1,\ldots\ldots,N_s\}$ una etiqueta, y $N_s = \prod^{n}_{i=1} |A_i|$ el número de subespacios difusos de entrada. En este estudio, se usa el siguiente:

   $$E^+ (S_s) = \{ e_l \in E \mid \forall i \in \{1,\ldots\ldots,m\}, \quad \forall A'_i \in A_i , \ \mu A^s_i(x^l_i) \geq \mu A'_i(x^l_i) \ \}$$

   siendo $\mu A^s_i(.)$ la función de pertenencia asociada con la etiqueta A^s_i.

   Para todos los N_s posibles subespacios de entrada difusos, considerar solo aquellos que contengan al menos un ejemplo positivo. Para ello, el conjunto de subespacios de entrada con ejemplos positivos se define como $S^+ = \{S_h \mid E^+ (S_h) \neq \varnothing\}$.

   1.2. *Generar el conjunto de reglas candidatas en cada subespacio con ejemplos positivos:* En primer lugar, el conjunto de consecuentes candidatos con cada subespacio conteniendo al menos un ejemplo, $S_h \in S^+$, es definido. En este estudio, usamos el siguiente:

   $$C (S_h) = \{ (B_1{}^k h, \ldots\ldots, B_p{}^k h \ ) \in B_1 \times \ldots \times B_p \mid \exists e_l \in E^+ (S_h) \text{ donde } \forall j \in \{1,\ldots\ldots,p\}, \quad \forall B'_j \in B_j , \ \mu B_j{}^k h \ (y^l_j) \geq \mu B'_j(y^l_j) \ \}$$

Entonces, el conjunto de reglas candidatas para cada subespacio es definida como $CR(S_h) = \{Rk_h = [$ SI X_1 es A^h_1 y ... y X_m es A^h_m *ENTONCES* Y_1 es $B_1{}^k h$ y ... y Y_p es $B_p{}^k h]$ tal que $B^k h = (B_1{}^k h, ..., B_p{}^k h) \in C(S_h)\}$.

Para permitir que COR reduzca el número inicial de reglas difusas, el elemento especial $R_\varnothing$ (el cual significa "ningún consecuente") se añade a cada regla candidata, ejemplo., $CR(S_h) = CR(S_h) \cup R_\varnothing$. Si esta es seleccionada, no se selecciona ninguna regla en el correspondiente subespacio de entrada difusa.

2. *Selección del conjunto de reglas difusas que mejor cooperación obtengan.*– Esta fase es ejecutada por un algoritmo de búsqueda combinatorial para buscar la combinación $BR = \{R_1 \in CR(S_1), ..., R_h \in CR(S_h), ..., R_{|S+1|} \in CR(S_{|S+1|})\}$ con la mejor precisión. Debido a que dicho espacio de búsqueda es generalmente extenso, las técnicas de búsqueda heurísticas serían las más adecuadas.

   Para evaluar la calidad global del conjunto de reglas codificadas se considera un índice de medida *f(BR)* que mide la calidad de cada solución. Para obtener soluciones con una mayor interpretabilidad, esta función original se modifica para penalizar el excesivo número de reglas:

$$f'(BR) = f(BR) + \gamma \cdot f(BR_0) - \#BR / |S^+|$$

   siendo $\gamma \in [0, 1]$ un parámetro definido por el diseñador para regular la importancia del número de reglas, *#BR* el número de reglas usadas en la solución evaluada (i.e., $|S^+| - |\{R_h \in BR \text{ tal que } R_h = R_0\}|$), y siendo BR_0 la BR inicial considerada por el algoritmo de búsqueda.

Apéndice C

Tests no Paramétricos para el Análisis Estadístico de los Resultados

En este apéndice se explican con mayor detalle los distintos test estadísticos no paramétricos que se han usado en esta memoria para el análisis de los resultados obtenidos por los distintos métodos en estudio.

C.1. Test de Contraste de Hipótesis no Paramétricos

Un contraste o test de hipótesis es una técnica de inferencia estadística que permite, a partir de los datos obtenidos de una (o varias) muestra observada, decidir si se acepta o no una hipótesis formulada sobre una (o varias) población(es) o algoritmo(s).

Una hipótesis estadística es una asunción relativa a una o varias poblaciones, que puede ser cierta o no. Las hipótesis estadísticas se pueden contrastar con la información extraída de las muestras y se puede cometer un error, tanto si se aceptan como si se rechazan.

La hipótesis que se formula se denomina hipótesis de trabajo o nula y se denota H_0. La hipótesis contraria se denomina hipótesis alternativa, H_1. El test

de hipótesis decidirá, basándose en la muestra observada, si se acepta o no la hipótesis nula formulada frente a la hipótesis alternativa.

En un test de hipótesis se pueden cometer dos tipos de errores:

Error tipo I, se rechaza la hipótesis H_0 cuando es cierta.
Error tipo II, se acepta la hipótesis H_0 cuando es falsa.

Sólo se puede cometer uno de los dos tipos de error y, en la mayoría de las situaciones, se desea controlar la probabilidad de cometer un error de tipo I. Se denomina *nivel de significación o confianza* α de un test a la probabilidad de cometer un error tipo I, es decir, la probabilidad de que el estadístico de contraste caiga en la región de rechazo:

α = P(rechazar H_0 | H_0 es cierta)

El nivel de confianza α se ha de decidir de antemano, de forma que se establece la probabilidad máxima que se está dispuesto a asumir de rechazar la hipótesis nula cuando es cierta. Dicho nivel lo elige el usuario. La selección de dicho nivel conduce a dividir en dos regiones el conjunto de posibles valores del estadístico de contraste: la región de rechazo, con probabilidad α, bajo H_0 y la región de aceptación, con probabilidad 1-α, bajo H_0.

Si el estadístico de contraste calculado por el test (también conocido como p-valor o p-value), toma un valor perteneciente a la región de aceptación (p-valor $\geq \alpha$) entonces no existen evidencias suficientes para rechazar la hipótesis nula con un nivel de significación α y se dice que el contraste estadísticamente no es significativo. Si, por el contrario, el estadístico cae en la región de rechazo (p-valor $< \alpha$) entonces se asume que los datos no son compatibles con la hipótesis nula y se rechaza a un nivel de significación α. En este supuesto se dice que el contraste es *estadísticamente significativo*.

Si p-valor $\geq \alpha$ ==> Se acepta la hipótesis nula H0
Si p-valor $< \alpha$ ==> Se rechaza la hipótesis nula H0

En nuestro caso los test de contraste los vamos a utilizar para comparar los resultados obtenidos por los diferentes algoritmos y ver si existen o no diferencias significativas entre ellos. Es decir:

La hipótesis nula H_0 es que los resultados de los diferentes métodos son similares, mientras que la hipótesis alternativa H_1 significa que los resultados de los diferentes métodos son distintos.

Al aplicar el test, si la hipótesis nula H_0 se rechaza significa que estadísticamente existen diferencias significativas entre los resultados obtenidos por los métodos y por el contrario si se acepta H_0, no existen diferencias significativas entre los resultados, es decir, los métodos obtienen resultados equivalentes y se puede afirmar que son estadísticamente iguales.

Los test estadísticos pueden ser paramétricos y no paramétricos. Para aplicar test paramétricos es necesario, entre otras cosas, conocer la forma de la distribución de donde proceden los datos, ya que si no siguen una distribución normal no se pueden utilizar. Como en la práctica esto es difícil de conocer, una alternativa es el uso de test no paramétricos o de libre distribución, que no se basan en la hipótesis de que los datos siguen una determinada distribución de probabilidad.

Que las condiciones de aplicación de los test no paramétricos sean menos restrictivas que la de los test paramétricos, unido al hecho de que la mayoría de las veces sean más fáciles de aplicar, han motivado que nos decantemos por este tipo de test para validar los resultados obtenidos en los experimentos.

A continuación se describe los distintos métodos estadísticos no paramétricos utilizados en esta memoria para el análisis de los resultados obtenidos por los distintos métodos en estudio.

C.2. Test de Friedman

El test de Friedman es un test no paramétrico utilizado para realizar comparaciones múltiples y es equivalente al test paramétrico ANOVA. El test de Friedman se utiliza para determinar si en un conjunto de k muestras, al menos dos de las muestras son suficientemente dispares para ser consideradas significativamente diferentes (se ejecuta por tanto para realizar comparaciones múltiples y determinar si hay diferencias entre las muestras).

El test de Friedman trabaja de la siguiente forma: Calcula un *ranking* de los resultados obtenidos por cada algoritmo (r_j para cada algoritmo j, habiendo k algoritmos) sobre cada uno de los conjunto de datos, y le asigna al mejor el ranking 1 y al peor el ranking k. Si se produce un empate entre dos algoritmos, el test asigna el ranking medio a ambos. Bajo la hipótesis nula, este test indica que todos los algoritmos son equivalentes (y por tanto sus

rankings son similares). Por lo tanto, el rechazo de dicha hipótesis implica la existencia de diferencias en el rendimiento de todos los algoritmos estudiados.

Supongamos que r_j^i es el ranking del j-ésimo algoritmo (siendo k el número total de algoritmos) sobre el i-ésimo conjunto de datos (con un total de N). El test de Friedman lleva a cabo una comparación del ranking medio de los algoritmos, $R_j = 1/N \sum_i r_j^i$. Bajo la hipótesis nula, que como hemos dicho anteriormente afirma que todos los algoritmos son equivalentes y por lo tanto sus rankings R_j también deberían de ser igual, el estadístico de Friedman

$$X_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right] \qquad \text{(C.1)}$$

se distribuye de acuerdo con una distribución X^2_F con $k-1$ grados de libertad, siempre que N y k sean lo suficientemente grandes (como regla general, $N > 10$ y $K > 5$).

C.3. Test de Finner

El rechazo de la hipótesis nula en el test de Friedman no implica que existan diferencias entre todos los algoritmos. Sólo indica que existen diferencias entre todas las muestras obtenidas por los algoritmos, pero no indica si dichas diferencias son estadísticamente significativas. Para determinar si los algoritmos son estadísticamente diferentes es necesario compararlos por pares mediante un procedimiento a posteriori (post-hoc). En este caso, se elige uno de los algoritmos (algoritmo de control) y el procedimiento a posterior) comparan el elegido con el resto de los algoritmos.

El propósito de los test post hoc (a posteriori), por tanto, es determinar exactamente cuál de las muestras difieren con las otras en términos de diferencias de medias. Estos test se realiza generalmente después del test de Friedman, si éste indica que las muestras no son idénticas (si el test de Friedman acepta la hipótesis nula, no tiene sentido aplicar los test a posteriori, al no indicar Friedman que hay evidencias de diferencias entre las muestras).

El método de Finner es un test a posteriori para el estadístico de Friedman. Se utiliza cuando el test de Friedman rechaza la hipótesis nula (indica que hay evidencias de diferencias entre las muestras) y queremos corroborar que dichas diferencias son estadísticamente significativas. Este método lleva a

cabo un proceso de comparación por pares múltiple que trabaja con un algoritmo de control (normalmente, se elige el mejor de los algoritmos), el cual es comparado con el resto de algoritmos. El estadístico utilizado para comparar el i-ésimo y j-ésimo método es el siguiente:

$$z = (R_i - R_j) \Big/ \sqrt{\frac{k(k+1)}{6N}} \qquad \text{(C.2)}$$

El valor z se utiliza para encontrar la probabilidad correspondiente (*p-valor*) dentro de la tabla que represente la distribución normal, el cual se compara con un nivel de confianza α apropiado.

El test de Finner es un procedimiento incremental, que testea secuencialmente las hipótesis ordenadas por su valor de significancia. De esta forma, los procedimientos se ordenan secuencialmente por su importancia. Sean $p_1, p_2, \ldots, p_{k-1}$ los *p-valores* ordenados, de forma que $p_1 \leq p_2 \leq \ldots \leq p_{k-1}$ y sean $H_1, H_2, \ldots, H_{k-1}$ las hipótesis correspondientes. El test de Finner compara cada p_i con $1-(1-\alpha)^{(k-i)/k}$, comenzando por el que tiene el valor p más significativo. Si p_1 es menor que $1-(1-\alpha)^{(k-1)/k}$, la correspondiente hipótesis se rechaza y se pasa a comparar p_2 con $1-(1-\alpha)^{(k-2)/k}$. Si se rechaza la segunda hipótesis, el test continúa con la tercera y así sucesivamente. Cuando una cierta hipótesis nula no se puede rechazar, el test acaba las comprobaciones aceptando la hipótesis nula para el resto de hipótesis que aun quedaban por comprobar.

C.4. Test de Wilcoxon

El test de ranking de signos de Wilcoxon [Wil45] es un test no paramétrico análogo al *t-test*. Por lo tanto, se trata de un test por parejas que tiene por objetivo detectar la existencia de diferencias significativas entre el comportamiento de dos medias muestrales o dos algoritmos (si las medias se refieren a las salidas de dos algoritmos, entonces el test prácticamente evalúa el comportamiento recíproco de los dos algoritmos).

Su funcionamiento se basa en calcular las diferencias entre los resultados de dos algoritmos y calcular un ranking utilizando dicho valor, ignorando signos. En este caso el ranking va desde 1 a N, en vez de hasta k.

Sea d_i la diferencia entre las medidas de rendimiento de los dos algoritmos sobre el i-ésimo conjunto de datos (de un total de N). Se ordenan las

diferencias obtenidas en un ranking, según su valor absoluto (en caso de empates se asignan los ranking medios). Se calcula R^+ como la suma de los rankings de los conjuntos de datos en los que el primer algoritmo supera al segundo, y R^- la suma de los rankings en el caso contrario. Los rankings con valor $d_i = 0$ se reparten de forma equitativa entre las dos sumas anteriores (si hay un número impar de rankings con $d_i = 0$ se ignora uno de ellos).

$$R^+ = \sum_{d_i>0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i)$$
$$R^- = \sum_{d_i<0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i) \qquad \text{(C.3)}$$

Sea T el valor menor de ambas sumas, $T = min(R^+, R^-)$. Si T es menor o igual que el valor de la distribución de T de Wilcoxon para N grados de libertad (Tabla B.12 en [Wil45]), se rechaza la hipótesis nula de igualdad de las medias y el algoritmo asociado al mayor de los valores es el mejor.

El test de ranking de signos de Wilcoxon es más sensible que el t-test. Desde el punto de vista estadístico, el test de Wilcoxon es más seguro, ya que no se asume una distribución normal. Además, los valores atípicos (actuaciones excepcionalmente buenas o malas en unos pocos conjuntos de datos) tienen un efecto menor en el test de Wilcoxon que en el t-test. El test de Wilcoxon asume diferencias continuas d_i, por lo tanto, no se debería redondear a uno o dos decimales, ya que esto disminuiría el poder del test debido a un alto número de empates.

Apéndice D

Sistemas Basados en Reglas Difusas

Este apéndice describe los distintos tipos de SBRDs conocidos. En concreto, se describirán los SBRDs lingüísticos, empleados habitualmente para modelar sistemas que sean más interpretables, y los SBRDs aproximativos y TSK, que se utilizan para obtener sistemas más precisos.

D.1. Introducción a los SBRDs

Los Sistemas Difusos constituyen una de las áreas de aplicación más importantes de la Teoría de Conjuntos Difusos y de la Lógica Difusa, enunciadas por Zadeh en 1965 [Zad65]. Por regla general, este tipo de sistemas consideran la estructura de un modelo en la forma de reglas difusas y se denominan SBRDs. Los SBRDs son una extensión de los Sistemas Basados en Reglas de la lógica clásica, puesto que emplean reglas de tipo SI-ENTONCES en las que los antecedentes y consecuentes están compuestos de sentencias de lógica difusa, en lugar de proposiciones booleanas clásicas.

Podemos definir un sistema difuso como una caja gris [Bab98] que es capaz de inferir una salida dadas unas variables de entrada. El modelo de representación de las variables se realiza de forma que cada variable toma un conjunto de valores cualitativos que poseen una interpretación lingüística.

Cada valor que toma la variable se denomina etiqueta o término lingüístico. El significado de los términos lingüísticos por cada variable de entrada y salida constituyen un conjunto difuso, concretamente la función de pertenencia, que establece una correspondencia entre la variable numérica de entrada y salida y la variables difusa de la regla. Para modelizar el conocimiento dentro de un sistema difuso, se establece una relación entre las variables de entrada y salida al sistema mediante las reglas difusas.

Este conjunto de reglas difusas de tipo SI-ENTONCES constituye la parte esencial de los SBRDs y es la que efectúa el razonamiento difuso, teniendo en cuenta la información contenida en una BC, cuyos antecedentes y consecuentes se componen de estados difusos. La estructura lógica de las reglas difusas SI-ENTONCES facilita la comprensión y el análisis de los modelos en forma muy parecida a como razonan los seres humanos sobre el mundo real. La novedad que presentan los SBRDs es la adición de dos nuevos componentes que dotan al sistema con la capacidad de manejar entradas y salidas reales en lugar de difusas: las *interfaces de fuzzificación y defuzzificación.*

La Figura D.1 muestra la estructura general de los SBRDs.

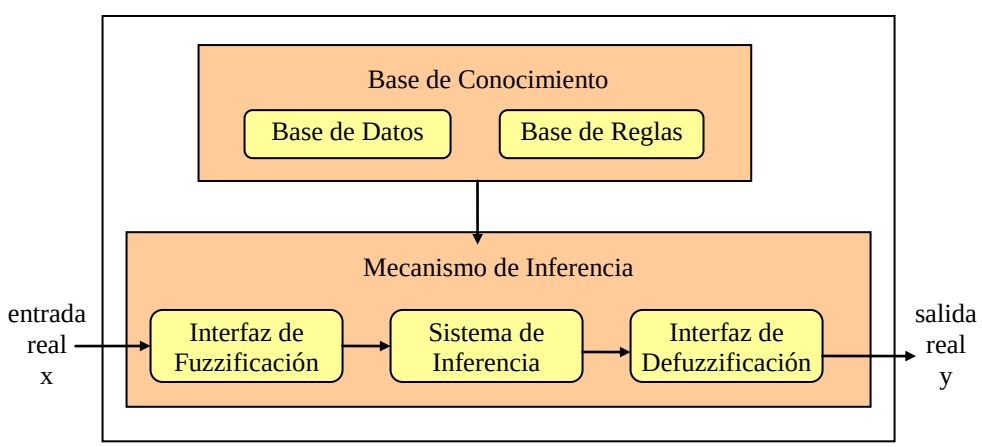


Figura D.1: Estructura básica de un SBRD

Como se puede apreciar en dicha figura, un SBRD está formado por los siguientes componentes:

- Una BC que contiene la información en forma de reglas difusas de tipo *SI-ENTONCES*:

  SI *un conjunto de condiciones se satisfacen*

  ENTONCES *un conjunto de consecuentes pueden ser inferidos*

  que guían el comportamiento del mismo.

- Un MI que se encarga de inferir la información de salida a partir de la información de entrada, que incluye:

  - Una interfaz de fuzzificación, que permite al SBRD lingüístico trabajar con entradas y salidas reales, ya que se encarga de transformar los datos de entrada precisos en conjuntos difusos.

  - Un SI, que emplea estos valores y la información contenida en la BC para hacer la inferencia por medio de un método de razonamiento.

  - Una ID o de salida, que transforma, mediante un método defuzzificación, los conjuntos difusos obtenidos del proceso de inferencia en una acción precisa que constituye la salida global del SBRD, en el caso de los problemas de regresión, o proporciona la clase final asociada al patrón de entrada de acuerdo con el modelo de inferencia, en el caso de los problemas de clasificación.

Tal como veremos en esta sección, es posible llevar a cabo distintos tipos de modelado empleando distintos tipos de SBRDs, dependiendo del grado de descripción y precisión que deseemos que tenga el futuro modelo. De esta forma, los SBRDs pueden ser clasificados de forma general en diferentes familias o categorías, cada una de las cuales están enfocadas a un tipo de modelado (más interpretable o más preciso) determinado.

La primera de ellas la compone los SBRDs lingüísticos o de tipo Mamdani [MA75, Mam74], que permiten llevar a cabo un modelado claramente interpretables para el ser humano, al estar las reglas difusas formadas por variables lingüísticas de entrada y salida que toman valores dentro de un conjunto de términos con un significado en el mundo real.

En esta primera categoría se incluye por tanto, los modelos lingüísticos basados en colecciones de reglas SI-ENTONCES unidas por el operador *ADEMÁS* cuyos antecedentes son valores lingüísticos, que permiten que el

comportamiento del sistema se puede describir en términos naturales y cuyo consecuente es una acción de salida (en los problemas de regresión) o clase final asociada (en los problemas de clasificación).

De esta forma, dependiendo de si el modelo se aplica a problemas de regresión o se aplica a problemas de clasificación el formato de las reglas es el siguiente:

$$R_i : SI\ X_{i1}\ es\ A_{i1}\ y\ ...\ y\ X_{im}\ es\ A_{im}\ ENTONCES\ Y\ es\ B_i \quad \text{(regresión)}$$

$$R_i : SI\ X_{i1}\ es\ A_{i1}\ y\ ...\ y\ X_{im}\ es\ A_{im}\ ENTONCES\ C_k\ con\ w_{ik} \quad \text{(clasificación)}$$

con i = 1 hasta N, siendo N el número de reglas, X_{i1} a X_{im} las variables lingüísticas de entrada y A_{i1} a A_{im} las etiquetas lingüísticas de los antecedentes, Y la variable lingüística de salida y B_i la etiqueta de los consecuentes implicados (en el caso de la regresión) y C_k la clase de salida asociado a la regla de clasificación y w_{ik} el peso de la regla o factor de certeza asociado a la clase (en el caso de la clasificación).

En los SBRDs lingüísticos, la BC está compuesta por dos componentes, la BR y la BD, los cuales almacenan respectivamente, las reglas que describen el comportamiento del sistema y los conjuntos difusos que describen a las variables lingüísticas contenidas en dichas reglas.

La segunda categoría es la formada por los SBRDs aproximativos [ACCH01, Koc96], que se caracteriza por el uso directo de variables difusas, de forma que cada regla difusa presenta su propia semántica (las variables toman diferentes conjuntos difusos como valores y no términos lingüísticos procedentes de un conjunto de términos globales como en el lingüístico).

Dado que en los SBRDs aproximativos no se emplea una semántica global, los conjuntos difusos no pueden interpretarse con facilidad. Estos modelos son más precisos que los anteriores a costa de la consiguiente pérdida de interpretabilidad, por lo que están enfocados para realizar un modelado preciso.

La estructura de las reglas de un SBRD aproximativo es la misma que la de un lingüístico, la única diferencia radica en que las reglas no manejan variables lingüísticas sino, directamente, variable difusas:

$$R_i : SI\ X_{i1}\ es\ \hat{A}_{i1}\ y\ ...\ y\ X_{im}\ es\ \hat{A}_{im}\ ENTONCES\ Y\ es\ \hat{G}_i,$$

donde $\hat{A}_{i1}$ a $\hat{A}_{im}$ y $\hat{G}_i$ son conjuntos difusos sin una interpretación lingüística directa, en lugar de etiquetas lingüísticas.

Al tener las reglas asociados valores difusos en lugar de variables lingüísticas, los SBRDs aproximativos no necesitan una BD para almacenar los términos lingüísticos y los conjuntos difusos que determinan la semántica asociada a los mismos. De esta forma, la BC empleada en los SBRDs lingüísticos, que estaba compuesta por la BD comentada y por una BR, queda reducida en los SBRDs aproximativos a una BR difusa compuesta por un conjunto de reglas en la que cada regla individual contiene la semántica que la describe.

La Figura D.2 muestra gráficamente la diferencia entre la BC de los SBRDs lingüísticos y la BR difusa de los SBRDs aproximativos.

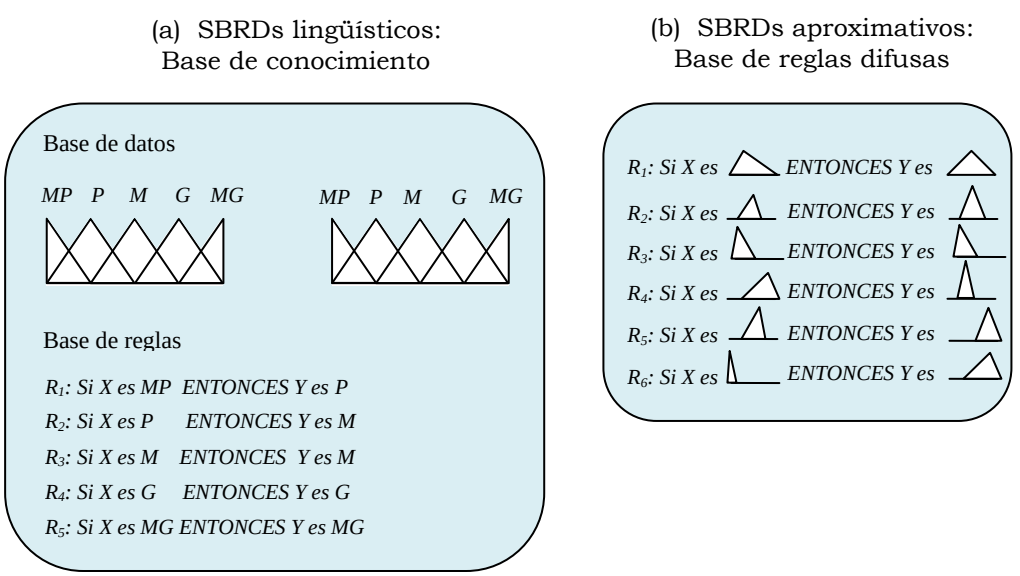


Figura D.2: Comparación gráfica entre una BC lingüística y una BR difusa aproximativa

La última categoría la compone los SBRDs de tipo TaKagi-Sugeno-Kang (TSK), en alusión a sus creadores [SK88, TS85] e incluye a los modelos

lingüísticos basados en una estructura de reglas en las que el antecedente está constituido por variables lingüísticas o difusas y el consecuente representa una función de las variables de entrada. La forma general de este tipo de reglas es la siguiente:

$$R_i : SI\ X_{i1}\ es\ A_{i1}\ y\ \ldots\ y\ X_{im}\ es\ A_{im}\ entonces\ Y = p(X_{i1}; \ldots; X_{im}),$$

siendo $p\ (.)$ una función polinómica de las variables contenidas en el antecedente, por lo general una expresión lineal de tipo $Y = p_0 + p_1 \cdot X_{i1} + \ldots + p_m \cdot X_{im}$, donde X_{im} son las variables de entrada del sistema, Y es la variable de salida y los p_i son parámetros reales. En lo que respecta a los A_{im}, pueden ser bien etiquetas lingüísticas asociadas con conjuntos difusos en el caso en que las X_{im} sean variables lingüísticas, o bien conjuntos difusos en el caso en que éstas sean directamente variables difusas.

En lo que respecta al tipo de modelado que es posible llevar a cabo, los SBRDs TSK constituyen un punto intermedio entre los dos anteriores, estando más cercano a uno u otro dependiendo de si los antecedentes de las reglas estén compuestos por variables lingüísticas, o por conjuntos difusos.

En esta tesis nos hemos centrado en los SBRD lingüísticos, que por su propia naturaleza son más interpretables que los otros. Por esa razón vamos a describir con más profundidad este tipo de sistemas.

D.1.1. Sistemas Basados en Reglas Difusas Lingüísticos

Los SBRDs lingüísticos fueron inicialmente propuestos por Mamdani y Assilian [MA75, Mam74], que plasmaron las ideas preliminares de Zadeh [Zad73] en el primer SBRD concreto en una aplicación de control. Este tipo de sistemas difusos es uno de los más usados desde entonces y se conoce también por el nombre de SBRD de tipo Mamdani o, sencillamente, *controlador difuso*, ya que su aplicación principal ha sido históricamente el control de sistemas.

Los SBRDs lingüísticos presentan una serie de características muy interesantes:

- Por un lado, puede emplearse en aplicaciones reales de ingeniería, puesto que maneja entradas y salidas reales.

- Por otro, proporciona un marco natural para incluir reglas lingüísticas proporcionadas por un experto y/o reglas obtenidas a partir de conjuntos de datos que reflejen el comportamiento del sistema.

- Por último, presenta una mayor libertad a la hora de elegir los interfaces de fuzzificación y defuzzificación, así como el SI, de modo que permite diseñar el SBRD más adecuado para un problema concreto.

La Figura D.3 muestra la estructura general de los SBRDs lingüísticos.

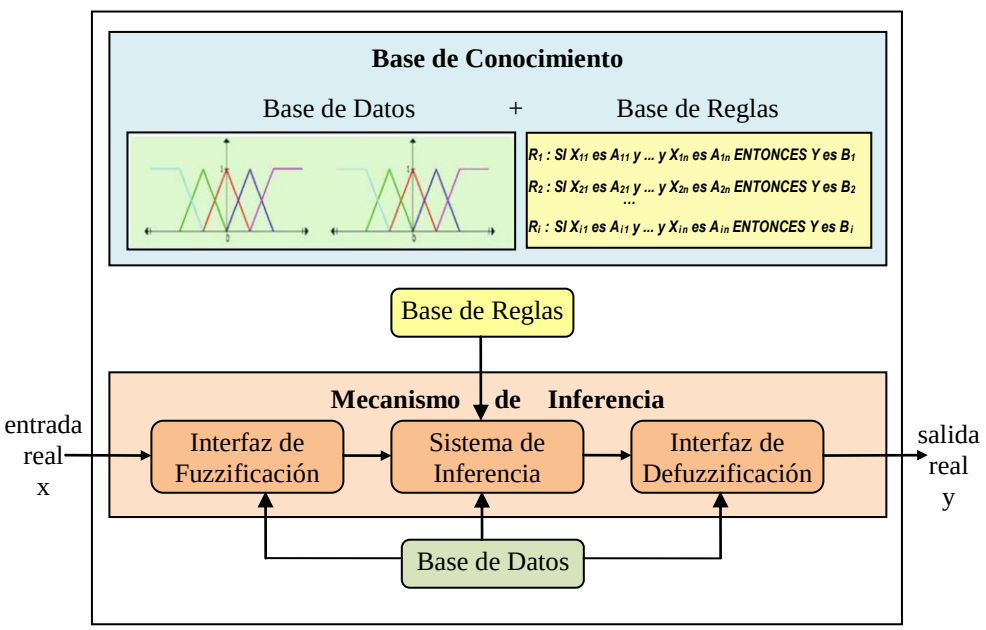


Figura D.3: Estructura general de un SBRD lingüístico.

Los SBRDs lingüísticos están compuestos por dos componentes fundamentales que son la BC y el módulo con el motor de inferencia. La BC es el elemento que contiene la información particular para cada problema en forma de reglas, la cual es interpretada por el MI. La BC de un SBRD lingüístico consta de dos componentes, la BD y la BR, que almacenan respectivamente la estructura de los conjuntos difusos que describen a las variables lingüísticas contenidas en dichas reglas (particiones lingüísticas) y las propias reglas:

- La BR está compuesta por un conjunto de reglas lingüísticas de tipo *SI-ENTONCES* unidas por una conectiva de reglas (el operador *ADEMÁS*), que indica que más de una regla puede dispararse simultáneamente para la misma entrada.

- La BD contiene los términos lingüísticos (más concretamente la definición de los conjuntos difusos asociados a ellos) empleados en las reglas de la BR y las funciones de pertenencia que definen la semántica de las etiquetas difusas lingüísticas. De este modo, cada variable lingüística involucrada en el problema tendrá asociada una partición difusa de su dominio que representa el conjunto difuso asociado a cada uno de los términos lingüísticos que dicha variable puede asumir. La Figura D.4 muestra un ejemplo de partición difusa con cinco etiquetas.

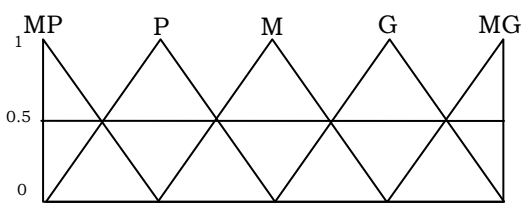


Figura D.4: Ejemplo de partición difusa

El número de conjuntos difusos en una partición difusa puede ser variable y se denomina granularidad. La determinación de las particiones difusas es fundamental en el modelado difuso [ACW06], especialmente en los problemas de control y regresión, ya que la granularidad de las particiones difusas juega un papel importante en el comportamiento y rendimiento de los SBRDs [CHV00].

El módulo con el motor de inferencia incluye:

- Una interfaz de fuzzificación, que es el componente que permite al sistema trabajar con entradas reales, ya que se encarga de transformar los datos de entrada precisos en valores utilizables en el proceso de razonamiento difuso, es decir, en algún tipo de conjunto difuso.

- Un SI, que a través de los datos recibidos por la interfaz de fuzzificación, utiliza la información contenida en la BC para llevar a cabo el proceso de inferencia difuso.

- Una ID, que es el componente que transforma la acción difusa resultante del proceso de inferencia en una acción precisa que constituye la salida global del SBRD.

A continuación se describen con mayor detalle los dos últimos elementos, por ser en estos donde realizamos nuestra investigación.

D.1.2. El Sistema de Inferencia

El SI es la componente esencial encargada de llevar a cabo el proceso de inferencia difuso, el cual se aplica a nivel de reglas individuales. Para ello, se hace uso de los principios de la Lógica Difusa para establecer una relación entre los conjuntos difusos definidos en la entrada y los conjuntos difusos definidos en la salida del modelo. Para llevar a cabo dicha inferencia, el SI emplea, en el caso de los problemas de regresión (que es el que nos interesa), los siguientes operadores:

- *un operador de conjunción C*, que permite realizar la intersección de conjuntos difusos propios de las premisas de las reglas difusas y calcular el grado de emparejamiento.

- *un operador de implicación difusa I*, correspondiente al condicional de las reglas *ENTONCES*, para calcular el consecuente inferido.

- la *agregación ADEMÁS*, que se emplea para combinar las aportaciones individuales de la inferencia con cada regla junto con un método de defuzzificación o de salida.

Por tanto, para el diseño del SI, se deben tomar las siguientes decisiones [KKS85]:

1. *Definir el operador de conjunción C*, a usar en el caso de que las reglas de la base presenten más de una variable de entrada. Para esta elección se dispone de distintos operadores de diversas familias. Entre los más utilizados están las t-normas.

2. *Definir el operador de implicación I*, en las reglas lingüísticas de tipo "*SI-ENTONCES*" contenidos en la BC, es decir, elegir el operador I que se empleará para modelar la implicación.

Existen distintas posibilidades para la elección de este operador. De las diferentes familias en las que pueden ser clasificados los *operadores de implicación*, las *funciones de implicación* [TV85] y las *t-normas* [GQ91] son las más conocidas.

3. *Definir el operador de agregación TAMBIÉN o ADEMÁS*, según el modo de defuzzificación que emplee el SBRD (véase siguiente apartado D.1.3).

En la práctica, para la selección del *operador de conjunción* y de *implicación* se utilizan generalmente *t-normas* [CHP00, GQ91, RPF13], por ser una familia de operadores generalmente utilizados en el diseño de modelos difusos [GQ91], dada su sencillez y eficacia computacional, y su buen comportamiento [CHP00] con gran independencia del método de defuzzificación con el que se combinen y la aplicación de modelado que se pretenda resolver. Generalmente como *operador de implicación* se suele utilizar la *t-norma del mínimo* y como *operador de conjunción* se suele utilizar alguna de las *t-normas* clásicas indicadas a continuación:

Tabla D.1: T-normas clásicas más utilizadas

T-norma	**Expresión**	
Producto Lógico (Mínimo)	$T_{Min}(x,y) = \text{Min}(x, y)$	(D.1)
Producto de Hamacher	$T_{Ham}(x,y) = \frac{x \cdot y}{x + y - x \cdot y}$	(D.2)
Producto Algebraico	$T_{Alg}(x,y) = x \cdot y$	(D.3)
Producto de Einstein	$T_{Eins}(x,y) = \frac{x \cdot y}{1+(1-x)\cdot(1-y)}$	(D.4)
Producto Acotado	$T_{Aco}(x,y) = \text{Max}(0, x+y-1)$	(D.5)
Producto Drástico	$T_{Dras}(x,y) = \begin{cases} x, \text{ si } y=1 \\ y, \text{ si } x=1 \\ 0, \text{ en otro caso} \end{cases}$	(D.6)

Un estudio detallado de la influencia de los distintos operadores que pueden intervenir en el diseño del MI lo podemos encontrar en [CHP95, CHP97, CHP99, CHP00]. En ella los autores hacen un estudio de la influencia del *operador de conjunción*, el *operador de implicación* y de los *métodos de defuzzificación*, así como las interacciones entre la decisión sobre estos elementos y la precisión del sistema obtenido finalmente.

D.1.3. La Interfaz de Defuzzificación

Como se ha descrito en la sección anterior, el proceso de inferencia difusa llevada a cabo por el SI se aplica a nivel de reglas individuales, de modo que, una vez aplicada la inferencia sobre las N reglas que componen la BR, se obtienen N conjuntos difusos B'_i que representan las acciones difusas que ha inferido el SI.

Dado que el SBRD debe devolver una salida precisa, la ID es quien asume la tarea de agregar la información aportada por cada uno de los conjuntos difusos individuales B'_i, resultantes del proceso de inferencia, y transformarla en un único valor concreto. Existen dos formas diferentes de implementar esta agregación [BD95, CHP97, Wan94]:

1. *Modo A (FATI: First Agregate Then Infer): agregar primero, defuzzificar después*. En este primer caso, la ID agrega los conjuntos difusos individuales inferidos B'_i, obteniendo como resultado un conjunto difuso final B' que posteriormente se transforma, mediante un *método de defuzzificación*, en un valor preciso que representa la salida global del sistema.

2. *Modo B: (FITA: First Infer Then Agregate): defuzzificar primero, agregar después*. En este caso se defuzzifica primero cada conjunto difuso inferido por cada regla a un valor característico concreto, y posteriormente, mediante una operación de agregación (una media o suma ponderada) aplicada sobre cada uno de los valores defuzzificados anteriores, se obtiene el valor preciso final.

siendo la forma más común de hacer esto el modo FITA [BH94], debido a su buen rendimiento, eficiencia y fácil implementación y porque evita el cálculo del conjunto difuso final B', hecho que ahorra una gran cantidad de tiempo computacional y complejidad.

Los métodos de defuzzificación más habituales son: el *Centro de Gravedad* y la *Media de los Máximos* [DHR93], cuando se trabaja en Modo A-FATI, y el *Centro de Gravedad*, el *Centro de Sumas* (aproximación al centro de gravedad computacionalmente más rápida de obtener) y el *Punto de Máximo Valor* al hacerlo en Modo B-FITA.

En caso de trabajar en Modo A-FATI, la función del *operador de agregación ADEMÁS* del SI sería la de agregar todos los conjuntos difusos individuales resultantes de la inferencia en un único conjunto difuso global. Para esta tarea se emplean habitualmente las *t-normas* y las *t-conormas*, principalmente el *mínimo* y el *máximo*, respectivamente por su sencillez.

Por otro lado, en caso de trabajar en Modo B-FITA, los *operadores de agregación* habitualmente empleados son la *media*, la *media ponderada* o la selección de algún *valor característico* de los conjuntos difusos, en función de algún grado de importancia de la regla que los ha generado en el proceso de inferencia [CHP97]. Como *valores característicos* se suelen emplear el *Centro de Gravedad* y el *Punto de Máximo Valor*, y como grados de importancia de la regla, el *área* y la *altura del conjunto difuso* inferido, o el *grado de emparejamiento de los antecedentes* [CHP97, HT93, SY93]. El operador más empleado dentro de este grupo es la *media ponderada por el grado de emparejamiento*, que se suele combinar con el *centro de gravedad* como *valor característico* del conjunto difuso [CHP97, HT93].

D.1.4. Operadores Adaptativos Difusos en el Sistema de Inferencia y la Interfaz de Defuzzificación

En esta sección se describe el Sistema de Inferencia adaptativo así como los métodos de defuzzificación adaptativos usados para nuestro propósito de aprendizaje.

Para mejorar el funcionamiento de los SBRDs, se puede recurrir a realizar un proceso de aprendizaje o ajuste evolutivo de alguno de los componentes del sistema, entre los cuales se encuentra los parámetros del SI y de la ID. A continuación se analizarán dichos operadores y se verá como introducir parámetros en ellos para dar lugar al SI adaptativo y la ID adaptativa, explicando la influencia y alcance que dichas parámetros tienen en el MI y por tanto, en la respuesta del sistema.

D.1.4.1. Sistema de Inferencia Adaptativo

Como hemos visto anteriormente, los SBRDs lingüísticos para Modelado de Sistemas utilizan reglas *SI - ENTONCES* de la siguiente forma:

$$R_i : Si\ X_{i1}\ es\ A_{i1}\ y\ ...\ y\ X_{im}\ es\ A_{im}\ ENTONCES\ Y\ es\ B_i \qquad (D.7)$$

con i = 1 hasta N, el número de reglas de la BR, siendo X_{i1} hasta X_{im} las variables de entradas e Y la de salida, y A_{i1} hasta A_{im} y B_i las etiquetas de los antecedentes y consecuentes de las reglas involucradas, respectivamente.

A partir de dichas reglas, el SI lleva a cabo el proceso de inferencia difuso obteniendo, a partir de los datos de entrada proporcionados por la interfaz de fuzzificación, unas salidas difusas que serán transformadas por la ID en una salida precisa que constituye la salida global del sistema. El proceso de inferencia se aplica a nivel de reglas individuales, por lo que, una vez aplicada la inferencia sobre las N reglas que componen la BR, el SI obtiene N conjuntos difusos B'_i que representan las acciones difusas que ha inferido.

La expresión de la Regla Composicional de Inferencia en modelado difuso para el caso más habitualmente empleado en el que la fuzzificación es puntual (cada entrada se transforma en un conjunto difuso que toma sólo un valor discreto), y la variable de salida es única, es la siguiente:

$$\mu_{B'}(y) = I\,(C\,(\mu_{A1}(x_1)\ ,\ ...\ ,\ \mu_{Am}(x_m)),\ \mu_B(y)) \qquad (D.8)$$

donde $\mu_{B'}(\cdot)$ es la función de pertenencia del consecuente inferido, $I(\cdot)$ es el operador de implicación, $C(\cdot)$ el operador de conjunción, $\mu_{Ai}(x_i)$ al tratarse de fuzzificación puntual, son los puntos de corte (los valores del grado de emparejamiento) de las entradas discretas del sistema $(x_{1,...,}x_m)$ con las funciones de pertenencia de los antecedentes de las reglas, *y* $\mu_B(\cdot)$ el consecuente de la regla.

Para realizar dicha inferencia, el SI primero utiliza el operador de conjunción $C(\cdot)$ para calcular el *grado de emparejamiento* de cada regla (h_i), y a continuación calcula el *consecuente inferido* de cada regla aplicando el operador de implicación $I(\cdot)$ al grado de emparejamiento calculado (h_i) y a la parte consecuente de cada regla $\mu_B(\cdot)$. Para la implementación de ambos operadores se suelen utilizar t-normas (véase Sección D.1.2) por las ventajas en sencillez, rendimiento y eficacia computacional que presentan.

Aunque estos dos operadores, conjunción e implicación, pueden utilizarse de forma parametrizada para permitir que el SI pueda adaptarse, [AFHMP07] y [MPH07] demostraron que los operadores de implicación parametrizados tienen poca influencia frente a los operadores de conjunción adaptativos por lo que, en la práctica en el SI adaptativo sólo se insertan parámetros en los operadores de conjunción, utilizándose habitualmente como operador de implicación la t-norma del mínimo.

Con el fin de aumentar el número de grados de libertad se pueden considerar a su vez dos modelos de SI adaptativo:

- Un modelo con un solo parámetro, α, para adaptar globalmente el comportamiento del SI adaptativo en su conjunto.

  $C(\alpha, \cdot)$ y $I(\cdot)$ SI adaptativo con Inferencia Global

- Un modelo con un parámetro individual para cada regla de la BR, α_i, teniendo en este caso un mecanismo de adaptación local del comportamiento para cada regla del SI.

  $C(\alpha_i, \cdot)$ y $I(\cdot)$ SI adaptativo con Inferencia Local

Como operador de conjunción adaptativo se suele utilizar algunas de las t-normas adaptativas clásicas descritas en [Miz89], las cuales se muestran en la tabla siguiente, junto con los rangos o dominios de sus parámetros:

Tabla D.2: T-normas adaptativas

T-norma	**Expresión**	**Dominio**	
Dubois	$T_{Dubois}(x,y,\alpha) = \frac{x \cdot y}{Max(x,y,\alpha)}$	$(0 \le \alpha \le 1)$	(D.9)
Dombi	$T_{Dombi}(x,y,\alpha) = \frac{1}{1+\sqrt[\alpha]{\left(\frac{1-x}{x}\right)^{\alpha}+\left(\frac{1-y}{y}\right)^{\alpha}}}$	$(\alpha > 0)$	(D.10)
Frank	$T_{Frank}(x,y,\alpha) = \log_{\alpha}\left[1+\frac{(\alpha^{x}-1)(\alpha^{y}-1)}{\alpha-1}\right]$	$(\alpha > 0)$, $(\alpha \neq 1)$	(D.11)

Para facilitar la comprensión de su funcionamiento y la comparación con las t-normas clásicas no adaptativas (véase Tabla D.1), la Tabla D.3 muestra la relación entre las seis t-normas clásicas no adaptativas y los valores de los parámetros de las t-normas adaptativas, es decir, en qué casos son equivalentes a ellas y dónde se encuentran sus valores.

Tabla D.3: Relación entre las t-normas clásicas y adaptativas dependiendo del valor del parámetro α

	$T_{Mínimo}$	$T_{Hamacher}$	$T_{P.\ Algebraico}$	$T_{Einstein}$	$T_{P.\ Acotado}$	$T_{P.\ Drástico}$
T_{Dubois}	0		1			
T_{Dombi}	∞	1				$\to 0$
T_{Frank}	$\to 0$		$\to 1$		∞	

Aunque la elección del operador de conjunción tiene escasa influencia en la precisión alcanzada por el sistema [CHP97], el modelo con un sólo parámetro, como es de esperar, ofrece peores resultados que el modelo que utiliza parámetros individuales para cada regla, ya que este último, al tener un mecanismo para alterar el comportamiento del SI para cada regla, consigue una precisión más alta [AFHMP07], al permitir que el operador de conjunción se adapte individualmente a cada regla, lo cual equivale a la utilización de una t-norma más apropiada para cada regla. Por todo ello, en la práctica se suele utilizar el modelo con un parámetro para cada regla, que dota de mayor flexibilidad al sistema y por tanto, consigue mejores resultados.

El operador de conjunción adaptativo utilizado modula la influencia del grado de emparejamiento, produciendo un efecto similar al uso de modificadores lingüísticos [CdJH98, LCT01], que alteran el significado lingüístico de la estructura de la regla modificando el significado de la etiqueta. En este sentido, el parámetro en la conjunción juega un papel similar, cambiando la forma de la función de pertenencia asociada con las etiquetas lingüísticas de los antecedentes de la regla.

La siguiente figura muestra los efectos que producen los modificadores lingüísticos de concentración 'muy' y el de dilatación 'más-o-menos' al aplicarlos sobre una función de pertenencia triangular:

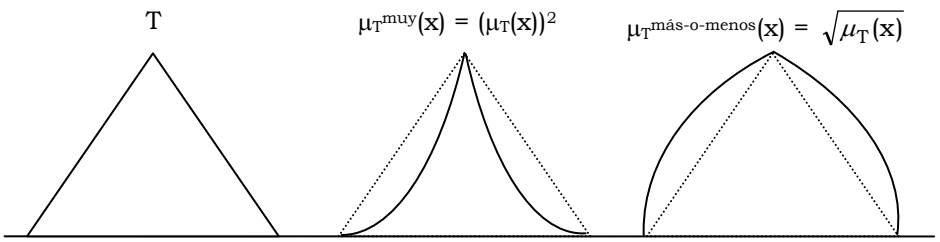


Figura D.5: Efectos de los modificadores lingüísticos 'muy' y 'más-o-menos'

D.1.4.2. Interfaz de Defuzzificación Adaptativa

Históricamente la ID, que es la encargada de la tarea de agregar la información aportada por cada uno de los conjuntos difusos individuales B'_i inferidos por el SI y transformarla en una salida precisa, fue el primer elemento del MI que trató de adaptarse [ESH00, JL96, YF93].

Aunque en la literatura científica podemos encontrar varios enfoques en el desarrollo de métodos de defuzzificación adaptativos [Hon95, KCL02, SL96], [CHMP04] demostró que la mayoría de las propuestas de parametrización de la ID, más eficientes y que mejores resultados prácticos muestran, pueden expresarse de forma generalizada mediante una expresión que incluye un término funcional que puede adoptar distintas formas. Esta expresión es la siguiente:

$$y_0 = \frac{\sum_{i=1}^{N} f(h_i) \cdot V_i}{\sum_{i=1}^{N} f(h_i)}, \qquad \text{(D.12)}$$

donde h_i es el grado de emparejamiento, $f(h_i)$ es una función sobre el grado de emparejamiento entre las variables de entrada y los conjuntos difusos de los antecedentes de las reglas y V_i representa un valor característico del conjunto difuso inferido desde la regla R_i, el máximo valor (MV_i) o el centro de gravedad (CG_i).

Como puede apreciarse, se trata de un método de defuzzificación que opera en el Modo B – FITA, consistente en una suma ponderada por el grado de emparejamiento. Si recordamos, la metodología FITA aplica la función de defuzzificación de forma individual a cada conjunto difuso inferido por cada regla (obteniendo para cada uno de ellos un valor característico concreto) y el valor preciso final se obtiene mediante un operador de promedio ponderado (una media, una suma ponderada o la selección de una de ellas, entre otras) aplicada sobre estos valores característicos obtenidos.

El término funcional puede utilizar un solo parámetro β, o bien un parámetro por cada regla de la BC, β_i. y puede ser definido como un producto o una potencia, dando lugar a las siguientes expresiones:

$$f(h_i) = h_i \cdot \beta, \quad f(h_i) = h_i \cdot \beta_i$$
$$f(h_i) = h_i^{\beta}, \quad f(h_i) = h_i^{\beta_i}. \tag{D.13}$$

Sin embargo, cabe constatar que no tiene sentido considerar la forma $f(h_i) = h_i \cdot \beta$ debido a que el efecto de β se cancela en la expresión. La combinación de los tres funcionales restantes con el MV o el CG genera seis métodos de defuzzificación (las expresiones se pueden consultar en [CHMP04]).

El papel del uso de un único parámetro β se interpreta como una modulación de la influencia del grado de emparejamiento, potenciándolo o penalizándolo. Sin embargo, cuando se usa un parámetro β_i por cada regla la interpretación es sensiblemente diferente ya que en lugar de una modulación global de la influencia del grado de emparejamiento, se tiene una acción local para cada regla defuzzificada.

El empleo del funcional de tipo producto con un parámetro diferente para cada regla tiene un efecto equivalente al uso de reglas con pesos [CP00]. Los valores de los parámetros de defuzzificación asociados con las reglas indican la importancia de estas reglas en el proceso de inferencia, lo mismo que el empleo de pesos que acompañan a cada regla de la BR (consistente en añadir un valor adicional β_i asociado a cada regla R_i) tiene el significado de indicar qué grado de importancia debe tener esa regla en el proceso de inferencia.

Para el caso del término funcional potencia, el efecto es equivalente al uso de modificadores lingüísticos que alteran el significado de la estructura de las reglas [LCT01], al igual que ocurre con la conjunción adaptativa.

Teniendo en cuenta los estudios realizados en [CHMP04], en el que se concluye que el uso de un parámetro por cada regla obtiene mejores

resultados que el uso de un único parámetro y que el funcional tipo producto, obtiene resultados similares al funcional tipo potencia y se calcula de forma más eficiente, en la práctica se suele utilizar el funcional de tipo producto con un parámetro para cada regla. De esta forma, la fórmula de defuzzificación adaptativa que se suele utilizar y que mejores resultados ofrece es la siguiente:

$$y_0 = \frac{\sum_{i=1}^{N} h_i \cdot \beta_i \cdot V_i}{\sum_{i=1}^{N} h_i \cdot \beta_i}, \qquad \text{(D.14)}$$

donde V_i representa un valor característico del conjunto difuso inferido por la regla R_i, generalmente el *MV* o el *CG* (éste último es el que usamos en nuestra memoria), h_i representa el grado de emparejamiento que hay entre las variables de entrada y los conjuntos difusos de los antecedentes de las reglas, y β_i representa un parámetro por cada regla R_i, i = 1 hasta N, de la BR, que indica la importancia de la regla para el proceso de inferencia.

Este método de defuzzificación es equivalente al conocido WCOA cuando los parámetros β_i se igualan a la unidad. Como se puede apreciar, la influencia del parámetro individual β_i se puede interpretar como una modulación de la influencia del grado de emparejamiento, potenciándolo o penalizándolo. Esta modulación es solo lineal en el caso de producto. Particularmente, el estudio zonal del efecto de β_i es el siguiente:

$$\begin{array}{l} \beta_i \cdot h_i,\ \beta_i \in [1, \infty)\text{: potenciación de } h_i, \\ \beta_i \cdot h_i,\ \beta_i \in [0, 1]\text{: penalización de } h_i. \end{array} \qquad \text{(D.15)}$$

En la práctica, el intervalo para β_i utilizado generalmente se limita a [0, 1] con el fin de actuar sólo como una penalización en diferentes niveles, obteniendo resultados similares.

D.2. Tareas en el Aprendizaje de SBRDs

El aprendizaje de un SBRD mediante un proceso de aprendizaje inductivo supervisado comienza con un conjunto de ejemplos y tiene como objetivo final la obtención de un SBRD que determine, con el mínimo error posible la salida

del sistema a partir de unos datos de entrada. Una vez obtenido el SBRD, se determina su rendimiento sobre datos de prueba para tener una estimación del error real del SBRD.

El análisis de los componentes y del funcionamiento de un SBRD nos permite determinar que el proceso de aprendizaje implica principalmente la realización de dos tareas complementarias entre sí:

1. El establecimiento de las particiones difusas para los dominios de las variables del problema y

2. La generación de un conjunto de reglas difusas.

Ambas tareas se engloban dentro del proceso de generación de la BC, el cual debe obtener un conjunto de reglas difusas que representen, de la forma más precisa posible, el comportamiento del sistema.

La primera de las tareas consiste en la obtención del conjunto de etiquetas lingüísticas (y conjuntos difusos asociados) óptimas para cada variable. Este es un problema difícil que se ha afrontado de diferentes formas en la literatura especializada. Mientras que algunos proponen utilizar distintas particiones difusas para cada variable, que se diferencian en el número de términos utilizados [INT94, IYN05], otros proponen establecer una partición difusa fija para cada variable en base a conocimiento experto o a una división equidistante del dominio con un tipo de función de pertenencia (triangular, trapezoidal, …) preestablecida o determinada tras una experimentación.

Con respecto a la segunda de las tareas, la literatura especializada también muestra que es posible utilizar diferentes métodos para la generación de un conjunto de reglas difusas: métodos basados en redes neuronales, métodos basados en técnicas de agrupamiento, métodos basados en la derivación de reglas difusas a partir de árboles de decisión o métodos basados en AEs.

La razón de incluir los AEs como método para la generación de un conjunto de reglas difusas estriba en que dicho proceso se puede tratar como un problema de optimización o búsqueda y es bien sabido que los AEs y más concretamente los AGs son técnicas de búsqueda que han mostrado de forma teórica y práctica capacidad robusta de búsqueda en espacios complejos. De hecho, en los últimos años ha aumentado el interés sobre los AGs debido a su alta potencialidad debido a que aportan una forma adecuada para codificar y evolucionar operadores de agregación de BRs y métodos de defuzzificación. Por esta razón, y aunque los AEs no se pueden considerar algoritmos de

aprendizaje, constituyen una herramienta de optimización robusta, independiente del dominio y aplicable a ésta y otras tareas que componen el aprendizaje de SBRDs.

Bibliografía

[AAFGH07] Alcalá R., Alcalá-Fdez J., Gacto M. J. y Herrera F. (2007) Rule base reduction and genetic tuning of fuzzy systems based on the linguistic 3-tuples representation. *Sof Computing* 11(5): 401-419.

[AAFHO07] Alcalá R., Alcalá-Fdez J., Herrera F. y Otero J. (2007) Genetic learning of accurate and compact fuzzy rule based systems based on the 2-tuples linguistic representation. *International Journal of Approximate Reasoning* 44(1): 45-64.

[ABS+15] Alshomrani S., Bawakid A., Shim S.O., Fernandez A. y Herrera F. (2015) A proposal for evolutionary fuzzy systems using feature weighting: dealing with overlapping in imbalanced datasets. *Knowledge-Based Systems* 73: 1–17.

[ACCH01] Alcalá R., Casillas J., Cordón O. y Herrera F. (2001) Building fuzzy graphs: features and taxonomy of learning for non-grid-oriented fuzzy rule-based systems. *Journal of Intelligent & Fuzzy Systems* 11: 99-119.

[ACW06] Au W. H., Chan K. C. C. y Wong A. K. C. (2006) A fuzzy approach to partitioning continuous attributes for classification. *IEEE Transactions on Knowledge Data Engineering* 18(5): 715–719.

[ADH+09] Alcalá R., Ducange P., Herrera F., Lazzerini B. y Marcelloni F. (2009) A multi-objective evolutionary approach to concurrently learn rule and data bases of linguistic fuzzy rule-based systems. *IEEE Transactions on Fuzzy Systems* 17(5): 1106-1122.

[ADLM09] Antonelli M., Ducange P., Lazzerini B. y Marcelloni F. (2009) Learning concurrently partition granularities and rule bases of mamdani fuzzy systems in a multi-objective evolutionary framework. *International Journal of Approximate Reasoning* 50(7): 1066–1080.

[ADLM11] Antonelli M., Ducange P., Lazzerini B. y Marcelloni F. (2011) Learning concurrently data and rule bases of Mamdani fuzzy rule-based systems by exploiting a novel interpretability index. *Soft Computing* 15: 1981–1998.

[ADM12] Antonelli M., Ducange P. y Marcelloni F. (2012) Genetic training instance selection in multi-objective evolutionary fuzzy system: A co-evolutionary approach, *IEEE Transactions on Fuzzy Systems* 20(2): 276–290.

[AFHMP07] Alcalá-Fdez J., Herrera F., Márquez F. y Peregrín A. (2007) Increasing fuzzy rules cooperation based on evolutionary adaptive inference system. *International Journal of Intelligent Systems* 22(9): 1035-1064.

[AFSG+09] Alcalá-Fdez J., Sánchez L., García S., del Jesus M. J., Ventura S., Garrell J., Otero J., Romero C., Bacardit J., Rivas V., Fernández J. y Herrera F. (2009) KEEL: A software tool to assess evolutionary algorithms to data mining problems, *Soft Computing* 13(3): 307–318.

[AGH11] Alcalá R., Gacto M. J. y Herrera F. (2011) A fast and scalable multiobjective genetic fuzzy system for linguistic fuzzy modeling in high-dimensional regression problems. *IEEE Transactions on Fuzzy Systems* 19(4): 666-681.

[AGHAF07] Alcalá R., Gacto M. J., Herrera F. y Alcalá-Fdez J. (2007) A multi-objective genetic algorithm for tuning and rule selection to obtain accurate and compact linguistic fuzzy rule-based systems. *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems* 15(5): 539-557.

[AM11] Alonso J.M. y Magdalena L. (2011) Special issue on interpretable fuzzy systems. *Information Sciences*, 181:4331–4339.

[AMGR09] Alonso J. M., Magdalena L. y Gonzalez-Rodríguez G. (2009) Looking for a good fuzzy system interpretability index: An experimental approach. *International Journal of Approximate Reasoning* 51(1): 115–134.

[Amo13] Amores J. (2013) Multiple instance classification: review, taxonomy and comparative study. *Artificial Intelligence* 201: 81–105.

[ANHI11] Alcalá R., Nojima Y., Herrera F. y Ishibuchi H. (2011) Multiobjective genetic fuzzy rule selection of single granularity-based fuzzy classification rules and its interaction with the lateral tuning of membership functions. *Soft Computing, Special Issue on Evolutionary fuzzy Systems* 15(12): 2303–2318.

[APLM09] Antonelli M., Pietro D., Lazzerini B. y Marcelloni F. (2009) Multi-objective evolutionary learning of granularity, membership function parameters and rules of Mamdani fuzzy systems, *Evolutionary Intelligence, Special Issue on GFS: New Advances*, 2: 21-37.

[Ata15] Ata R. (2015) Artificial neural networks applications in wind energy systems: a review. *Renewable and Sustainable Energy Reviews* 49:534–562.

[AT02] Alba E. y Troya J.M. (2002) Improving flexibility and efficiency by adding parallelism to genetic algorithms, *Statistics and Computing* 12(2): 91–114.

[Bab98] Babuška R. (1998) *Fuzzy modeling for control*. Kluwer Academic, Norwell, Massachusetts, EE.UU.

[BAB01] Baron L., Achiche S. y Balazinski M. (2001) Fuzzy decision support system knowledge base generation using a genetic algorithm. *International Journal of Approximate Reasoning* 28(1): 125-148.

[Bäc96] Bäck T. (1996) *Evolutionary algorithms in theory and practice*. Oxford University Press.

[Bak87] Baker J. E. (1987) Reducing bias and inefficiency in the selection algorithm. En Grefenstette J. J. (Ed.) *Proceedings of the 2nd International Conference on Genetic Algorithms* (ICGA'87), páginas 14-21. Lawrence Erlbaum Associates, Hillsdale, NJ, EE.UU.

[BBK09] Bacardit J., Burke E. K. y Krasnogor N. (2009) Improving the scalability of rule-based evolutionary learning, *Memetic Computing* 1(1): 55–67.

[BC13] Benitez A. D. y Casillas J. (2013) Multi-objective genetic learning of serial hierarchical fuzzy systems for large-scale problems, *Soft Computing* 17(1): 165–194.

[BD95] Bardossy A. y Duckstein L. (1995) *Fuzzy rule-based modeling with application to geophysical, biological and engineering systems*. CRC Press, Boca Raton, Florida, EE. UU.

[BdVMP09] Bardallo J. M., de Vega M. A., Márquez F. A. y Peregrín A. (2009) Parallel evolutionary multiobjective methodology for granularity and rule base learning in linguistic fuzzy systems. E*n Proceedings of the IEEE International Conference on Fuzzy System (FUZZ-IEEE'09),* páginas 1700–1705. Jeju Island, Korea.

[BFK13] Bäck T., Foussette C. y Krause P. (2013) *Taxonomy of Evolution Strategies*. Springer Berlin Heidelberg. ISBN 978-3-642-40136-7.

[BFM97] Bäck T., Fogel D. y Michalewicz Z. (1997) *Handbook of evolutionary computation*. Oxford University Press.

[BH94] Buckley J. J. y Hayashi Y. (1994) Can approximate reasoning be consistent? *Fuzzy Sets and Systems* 65(1): 13–18.

[BJ11] Beliakov G. y James S. (2011) Citation-based journal ranks: the use of fuzzy measures. *Fuzzy Sets and Systems* 167(1): 101–119.

[BLMS09] Botta A., Lazzerini B., Marcelloni F. y Stefanescu D. (2009) Context adaptation of fuzzy systems through a multiobjective evolutionary approach based on a novel interpretability index. *Soft Computing* 13(5): 437–449.

[Bon96] Bonarini A. (1996) Evolutionary learning of fuzzy rules: competition and cooperation. En Pedrycz W. (Ed.) *Fuzzy Modelling: Paradigms and Practice,* páginas 265-284. Kluwer Academic Press, Norwell, MA.

[BS91] Bäck T. y Schwefel H. P. (1991) Extended selection mechanisms in genetic algorithms. En *Proceedings Fourth International Conference on Genetic Algorithms (ICGA'91),* páginas 2-9. San Diego, EE.UU.

[BS02] Beyer H.G. y Schwefel H.P. (2002) Evolution strategies. a comprehensive introduction. *Natural Computing* 1(1):3–52.

[CA98] Combs W. E. y Andrews J. E. (1998) Combinatorial rule explosion eliminated by a fuzzy rule configuration. *IEEE Transactions on Fuzzy Systems* 6(1): 1-11.

[CBFO06] Crockett K. A., Bandar Z., Fowdar J., y O'Shea J. (2006) Genetic tuning of fuzzy inference within fuzzy classifier systems. *Expert Systems* 23(2): 63–82.

[CBM07] Crockett K., Bandar Z., y Mclean D. (2007) On the optimization of T-norm parameters within fuzzy decision trees. En *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'07)*, páginas 103–108. London, UK.

[CCdJH05] Casillas J., Cordón O., del Jesus M. J. y Herrera F. (2005) Genetic tuning of fuzzy rule deep structures preserving interpretability for linguistic modeling. *IEEE Transactions on Fuzzy Systems* 13(1): 13-29.

[CCH02] Casillas J., Cordón O. y Herrera F. (2002) COR: A methodology to improve ad hoc data-driven linguistic rule learning methods by inducing cooperation among rules. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics* 32(4): 526-537.

[CCHM03a] Casillas J., Cordón O., Herrera F. y Magdalena L. (Eds.) (2003) *Accuracy improvements to find the balance interpretability-accuracy in linguistic fuzzy modeling: an overview*. Springer Berlin Heidelberg, Alemania. ISBN 978-3-642-05703-8.

[CCHM03b] Casillas J., Cordón O., Herrera F. y Magdalena L. (Eds.) (2003) *Interpretability improvements to find the balance interpretability-accuracy in fuzzy modeling: an overview*. Springer Berlin Heidelberg, Alemania. ISBN 978-3-642-05702-1.

[CdJH98] Cordón O., del Jesús M. J. y Herrera F., (1998) Genetic learning of fuzzy rule-based classification systems cooperating with fuzzy reasoning methods. *International Journal of Intelligent Sy*stems, 13: 1025-1053.

[CdJHL99] Cordón O., del Jesus M. J., Herrera F. y Lozano M. (1999) MOGUL: A methodology to obtain genetic fuzzy rule-based systems under the iterative rule learning approach. *International Journal of Intelligent Systems* 14(9): 1123-1153.

[CDLM07] Cococcioni M., Ducange P., Lazzerini B. y Marcelloni F. (2007) A pareto-based multi-objective evolutionary approach to the identification of mamdani fuzzy systems. *Soft Computing* 11: 1013-1031.

[CFM96] Carse B., Fogarty T. C. y Munro A. (1996) Evolving fuzzy rule based controllers using genetic algorithms. *Fuzzy Sets and Systems* 80: 273-294.

[CFT13] Cordeiro R. L. F., Faloutsos C. y Traina Jr. C. (2013) *Data Mining in Large Sets of Complex Data*. Springer Briefs in Computer Science, Springer.

[CGH+04] Cordón O., Gomide F., Herrera F., Hoffmann F. y Magdalena L. (2004) Ten years of genetic fuzzy systems: Current framework and new trends. *Fuzzy Sets and Systems* 41(1): 5-31.

[CH97] Cordón O. y Herrera F. (1997) A three-stage evolutionary process for learning descriptive and approximate fuzzy logic controller knowledge bases from examples. *International Journal of Approximate Reasoning* 17(4): 369-407.

[CH01] Cordón O. y Herrera F. (2001) Hybridizing genetic algorithms with sharing scheme and evolution strategies for designing approximate fuzzy rule-based systems. *Fuzzy Sets and Systems* 118(2): 235-255.

[CHHM01] Cordón O., Herrera F., Hoffmann F. y Magdalena L. (2001) GENETIC FUZZY SYSTEMS. *Evolutionary tuning and learning of fuzzy knowledge bases*, volumen 19 *of Advances in Fuzzy Systems – Applications and Theory*. World Scientific, Singapore.

[CHMP04] Cordón O., Herrera F., Márquez F. y Peregrín A. (2004) A study on the evolutionary adaptive defuzzification methods in fuzzy modeling. *International Journal of Hybrid Intelligent Systems* 1: 36-48.

[CHP95] Cordón O., Herrera F y Peregrín A. (1995) T-norms vs implication functions as implications operators in fuzzy logic controllers. *Fuzzy Sets and Systems* 86: 15-41.

[CHP97] Cordón O., Herrera F. y Peregrín A. (1997) Applicability of the fuzzy operators in the design of fuzzy logic controllers. *Fuzzy Sets and Systems* 86: 15-41.

[CHP99] Cordón O., Herrera F. y Peregrín A. (1999) A practical study on the implementation of fuzzy logia controllers. *International Journal of Intelligent Control and Systems* 3: 49-91.

[CHP00] Cordón O., Herrera F. y Peregrín A. (2000) Searching for basic properties obtaining robust implications operators in fuzzy control. *Fuzzy Sets and Systems* 111: 237-251.

[CHP+07] Casillas J., Herrera F., Pérez R., del Jesus M. J. y Villar P. (2007) Special issue on genetic fuzzy systems and the interpretability accuracy trade-off. *International Journal of Approximate Reasoning* 44: 1–90.

[CHV00] Cordón O., Herrera F. y Villar P. (2000) Analysis and guidelines to obtain a good fuzzy partition granularity for fuzzy rule-based systems using simulated annealing. *International Journal of Approximate Reasoning* 25(3): 187-215.

[CHV01] Cordón O., Herrera F. y Villar P. (2001) Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base. *IEEE Transactions on Fuzzy Systems* 9(4): 667-674.

[CJKO01] Corne D., Jerram N., Knowles J. y Oates M. (2001) PESA-II: Region based selection in evolutionary multiobjective optimization. En *Proceedings of genetic and evolutionary computation conference*, páginas 283–290. San Francisco.

[CKO00] Corne D., Knowles J. y Oates M. (2000) The pareto envelope-based selection algorithm for multiobjective optimization. En *Proceedings Parallel Problem Solving from Nature Conference - PPSN VI, LNCS* 1917, páginas 839–848. Paris, France.

[CL00] Cheong F. y Lai R. (2000) Constraining the optimization of a fuzzy logic controller using an enhanced genetic algorithm. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics* 30(1): 31-46.

[CLM10] Cococcioni M., Lazzerini B. y Marcelloni F. (2010) Toward efficient multi-objective genetic Takagi-Sugeno fuzzy systems for high dimensional problems. *Computational Intelligence in Expensive Optimization Problems, Springer Series Studies in Evolutionary Learning and Optimization (ELO),* páginas 397-422. Springer-Verlag.

[CLM11] Cococcioni M., Lazzerini B. y Marcelloni F. (2011) On reducing computational overead in multi-objective genetic Takagi-Sugeno fuzzy systems. *Applied Soft Computing* 11(1): 675-688.

[CLV07] Coello C. A, Lamont G. B y Veldhuizen D. A. V. (2007) *Evolutionary Algorithms for Solving Multi-objective Problems, second ed.*. Genetic and Evolutionary Computation, Springer, Berlin, Heidelberg.

[CM12] Castillo O. y Melin P. (2012) Optimization of type-2 fuzzy systems based on bioinspired methods: a concise review. *Information Sciences* 205: 1–19.

[CMG+11] Castillo O., Melin P., Garza A. A., Montiel O. y Sepulveda R. (2011) Optimization of interval type-2 fuzzy logic controllers using evolutionary algorithms. *Soft Computing* 15(6): 1145–1160.

[Cor11] Cordon O. (2011) A historical review of evolutionary learning methods for Mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems. *International Journal of Approximate Reasoning* 52(6): 894–913.

[CP00] Cho J. S. y Park D. J. (2000) Novel fuzzy logia control based on weighting of partially inconsistent rules using neural network. *Journal of Intelligent & Fuzzy Systems* 8: 99-100.

[CRHSG14] Cruz-Ramirez M., Hervas-Martinez C., Sanchez-Monedero J. y Gutierrez P. A. (2014) Metrics to guide a multi-objective evolutionary algorithm for ordinal classification. *Neurocomputing* 135: 21–31.

[CS11] Cardoso J. S. y Sousa R. (2011) Measuring the performance of ordinal classification. *International Journal of Pattern Recognition and Artificial Intelligence* 25(8): 1173–1195.

[CSSL09] Candela J. Q., Sugiyama M., Schwaighofer A. y Lawrence N.D. (2009) *Dataset Shift in Machine Learning*. The MIT Press.

[CT01] Coello C. A. y Toscano G. (2001) A micro-genetic algorithm for multiobjective optimization. En *First international conference on evolutionary multi-criterion optimization. LNCS* 1993, páginas 126–140. London, UK.

[CVL02] Coello C. A., Veldhuizen D. A. V. y Lamont G. B. (2002) *Evolutionary algorithms for solving muti-objective problems*. Kluwer Academic Publishers.

[DAPM02] Deb K., Agrawal S., Pratab A. y Meyarivan T. (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2): 182–197.

[Deb01] Deb K. (2001) *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons. New York, USA.

[Dem06] Demšar J. (2006) Statistical comparisons of classifiers over multiple data sets, *Journal of Machine Learning Research* 7: 1–30.

[dHGGP09] de Haro-Garcia A. y Garcia-Pedrajas N. (2009) A divide-and-conquer recursive approach for scaling up instance selection algorithms. *Data Mining and Knowledge Discovery* 18(3): 392–418.

[DHR93] Driankov D., Hellendoorn H. y Reinfrank M. (1993) *An introduction to fuzzy control*. Springer- Verlag.

[dJGHM07] del Jesus M. J. , González P., Herrera F. y Mesonero M. (2007) Evolutionary fuzzy rule induction process for subgroup discovery: A case study in marketing. *IEEE Transactions on Fuzzy Systems* 15(4): 578-592.

[DLLP97] Dietterich T., Lathrop R. y Lozano-Perez T. (1997) Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence* 89(1–2): 31–71.

[dVBMP09] de Vega M. A., Bardallo J. M., Márquez F. A. y Peregrín A. (2009) Parallel distributed two-level evolutionary multiobjective methodology for granularity learning and membership functions tuning in linguistic fuzzy systems. En *Proceedings of ISDA'09 International Conference on Intelligent System Design and Applications*, páginas 134–139. Pisa, Italia.

[EMH01] Erickson M., Mayer A. y Horn J. (2001) The niched pareto genetic algorithm 2 applied to the design of groundwater remediation systems. En *Proceedings of first international conference on evolutionary multi-criterion optimization. LNCS* 1993, páginas 681–695. London, UK.

[Esh91] Eshelman L. J. (1991) The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. En Rawlin G. (Ed.) *Foundations of genetic Algorithms,* volumen 1, páginas 265-283. Morgan Kaufman.

[ESH00] Esogbue A. O., Song Q. y Hearnes II W. E. (2000) Defuzzification filters and applications to power system stabilization problems. *Journal of Mathematical Análisis and Applications* 251: 406-432.

[FAN+13] Fazzolari M., Alcalá R., Nojima Y., Ishibuchi H. y Herrera F. (2013) A review of the application of multi-objective evolutionary systems: Current status and further directions. *IEEE Transactions on Fuzzy Systems* 21(1): 45-65.

[FF93] Fonseca C. M. y Fleming P. J. (1993) Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. En *Proceedings of 5th international conference on genetic algorithms*, páginas 416–423. San Mateo.

[FF96] Fogel D.B. y Fogel L.J. (1996) *An introduction to evolutionary programming*. Springer Berlin Heidelberg. ISBN 978-3-540-61108-0.

[FGA+13] Fazzolari M., Giglio B., Alcala R., Marcelloni F. y Herrera F. (2013) A study on the application of instance selection techniques in genetic fuzzy rule-based classification systems: accuracy-complexity trade-off. *Knowledge-Based Systems* 54: 32–41.

[Fin93] Finner H. (1993) On a monotonicity problem in step-down multiple test procedures. *Journal of the American Statistical Association* 88: 920–923.

[FLdJH15] Fernández A., López V., del Jesús M. J. y Herrera F. (2015) Revisiting evolutionary fuzzy systems: Taxonomy, applications, new trends and challenges. *Knowledge-Based Systems* 80:109–121.

[Fog12] Fogel G.B. (2012) *Evolutionary Programming*. Springer Berlin Heidelberg. ISBN 978-3-540-92909-3.

[Fri37] Friedman M. (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association* 32: 674–701.

[FdRL+14] Fernandez A., del Rio S., Lopez V., Bawakid A., del Jesus M. J., Benitez J.M. y Herrera F. (2014) Big data with cloud computing: an insight on the computing environment, MapReduce and programming frameworks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 4(5): 380–409.

[FV14] Frenay B. y Verleysen M. (2014) Classification in the presence of label noise: a survey. *IEEE Transactions on Neural Networks and Learning Systems* 25(5): 845–869.

[GAH09] Gacto M. J., Alcalá R. y Herrera F. (2009) Adaptation and application of multi-objective evolutionary algorithms for rule reduction and parameter tuning of fuzzy rule-based systems. *Soft Computing* 13(5): 419-436.

[GAH10] Gacto M. J., Alcalá R. y Herrera F. (2010) Integration of an index to preserve the semantic interpretability in the multi-objective evolutionary rule selection and tuning of linguistic fuzzy systems. *IEEE Transactions on Fuzzy Systems* 18(3): 515-531.

[GAH11] Gacto M. J., Alcalá R. y Herrera F. (2011) Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures. *Information Sciences* 181(20): 4340–4360.

[GFLH09] García S., Fernández A., Luengo J. y Herrera F. (2009) A study of statistical techniques and performance measures for genetics-based machine learning: Accuracy and interpretability. *Soft Computing* 13(10): 959–977.

[GFLH10] García S., Fernández A., Luengo J. y Herrera F. (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences* 180(10): 2044–2064.

[GGAH14] Gacto M. J., Galende M., Alcala R. y Herrera F. (2014) METSK-HDe: a multiobjective evolutionary algorithm to learn accurate TSK-fuzzy systems in highdimensional and large-scale regression problems. *Information Sciences* 276: 63–79.

[GH03] Guillaume S. y Charnomordic B. (2003) A new method for inducing a set of interpretable fuzzy partitions and fuzzy inference systems from data. En J. Casillas, O. Cordón, F. Herrera, L. Magdalena (Eds.) *Interpretability issues in fuzzy modeling*, páginas 148–175. Springer-Verlag.

[GH04] Guillaume S. y Charnomordic B. (2004) Generationg an interpretable family of fuzzy partitions from data. *IEEE Transaction on Fuzzy Systems* 12(3): 324–335.

[GH08] García S. y Herrera F. (2008) An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. *Journal of Machine Learning Research* 9: 2579–2596.

[GLH14] Garcia S., Luengo J. y Herrera F. (2014) *Data Preprocessing in Data Mining, Intelligent Systems Reference Library, first ed., vol. 72, Springer*.

[Gol89] Golberg D. E. (1989) *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley.

[GP99] González A. y Pérez R. (1999) SLAVE: a genetic learning system based on an iterative approach. *IEEE Transactions on Fuzzy Systems* 7(2): 176-191.

[GPdHG12] Garcia-Pedrajas N. y de Haro-Garcia A. (2012) Scaling up data mining algorithms: review and taxonomy. *Progress in Artificial Intelligence* 1(1): 71–87.

[GPdHP13] Garcia-Pedrajas N., de Haro A. y Perez J. (2013) A scalable approach to simultaneous evolutionary instance and feature selection. *Information Sciences* 228 (2013) 150–174.

[GQ91] Gupta M. y Qi J. (1991) Theory of t-norms and fuzzy inference methods. *Fuzzy Sets and Systems* 40: 431-451.

[GRP+07] González J., Rojas I., Pomares H., Herrera L. J., Guillén A., Palomares J. M. y Rojas F. (2007) Improving the accuracy while preserving the interpretability of fuzzy function approximators by means of multi-objective evolutionary algorithms. *International Journal of Approximate Reasoning* 44: 32–44.

[GT12] Garg A. y Tai K. (2012) Review of genetic programming in modeling of machining processes. In *Proceedings of International Conference on Modelling, Identification and Control (ICMIC),* páginas 653–658. IEEE.

[Her08] Herrera F. (2008) Genetic fuzzy systems: Taxonomy, current research trends and prospects. *Evolutionary Intelligence* 1: 27–46.

[Hir93] Hirota K. (Ed.) (1993) *Industrial applications of fuzzy technology*. Springer-Verlag.

[HLS03] Herrera F., Lozano M. y Sánchez A. (2003) A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study. *International Journal of Intelligent Systems* 18: 309–338.

[HLV97] Herrera F., Lozano M. y Verdegay J. L. (1997) Fuzzy connectives based crossover operators to model genetic algorithms population diversity. *Fuzzy Sets and Systems* 92(1): 21-30.

[HLV98a] Herrera F., Lozano M. y Verdegay J. L. (1998) A learning process for fuzzy control rules using genetics algorithms. *Fuzzy Sets and Systems* 100: 143-158.

[HLV98b] Herrera F., Lozano M. y Verdegay J. L. (1998) Tackling real-coded genetic algorithms: operators and tools for behaviorural analysis. *Artificial Intelligence Review* 12: 265-319.

[HM95] Homaifar A. y McCormick E. (1995) Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms. *IEEE Transactions on Fuzzy Systems* 3(2): 129-139.

[HMP03] Herrera F., Márquez F. y Peregrín A. (2003) Genetic adaptation of rule connectives and conjunction operators in fuzzy rule based system: An experimental comparative. *Third International Conference of the European Society for Fuzzy Logic and Technology Study*, páginas 100-104. Zittau, Alemania.

[HNG94] Horn J., Nafpliotis N. y Goldberg D. E. (1994) A niched pareto genetic algorithm for multiobjective optimization. En *Proceedings of first IEEE conference on evolutionary computation, IEEE World Congress on Computational Intelligence,* volumen 1, páginas 82-87. Piscataway, N. J.

[Hol75] Holland J. H. (1975) *Adaptation in natural and artificial systems*. Ann arbor: The University of Michigan Press.

[Hon95] Hong C. Z. (1995) Handwritten numeral recognition using a small number of fuzzy rules with optimized defuzzification parameters. *Neural Networks* 8(5): 821-827.

[HPZ+12] Hu Q., Pan W., Zhang L., Zhang D., Song Y., Guo M. y Yu D. (2012) Feature selection for monotonic classification. *IEEE Transactions on Fuzzy Systems* 20(1): 69–81.

[HT93] Hellendoorn H. y Thomas C. (1993) Defuzzification in fuzzy controllers. *Journal of Intelligent Fuzzy Systems* 1: 109-123.

[Hül08] Hüllermeier E. (2008) Fuzzy methods for data mining and machine learning: State of the art and prospects. *Fuzzy Sets and Their Extensions: Representation, Aggregation and Models*, páginas 357–375.

[IM96] Ishibuchi H. y Murata T. (1996) Multi-objective genetic local search algorithm. En *Proceedings Third IEEE International Conference on Evolutionary Computation,* páginas 119–124. Japan.

[IMN13] Ishibuchi H., Mihara S. y Nojima Y. (2013) Parallel distributed hybrid fuzzy GBML models with rule set migration and training data rotation. *IEEE Transactions on Fuzzy Systems* 21(2): 355-368.

[IMT95] Ishibuchi H., Murata T. y Türksen I. B. (1995) Selecting linguistic classification rules by two-objective genetic algorithms. En *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, páginas 1410–1415. Vancouver, Canada.

[IMT97] Ishibuchi H., Murata T. y Türksen I. B. (1997) Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems. *Fuzzy Sets and Systems* 89(2): 135-150.

[IMTN14] Ishibuchi H., Masuda H., Tanigaki Y. y Nojima Y. (2014) Review of coevolutionary developments of evolutionary multi-objective and many-objective algorithms and test problems. In *IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*, páginas 178–184, Orlando, Florida, EE.UU. IEEE.

[IN07] Ishibuchi H. y Nojima Y. (2007) Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning. *International Journal of Approximate Reasoning* 44(1): 4-31.

[INM99] Ishibuchi H., Nakashima T. y Murata T. (1999) Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems. *IEEE Transactions on Systems, Man and Cybernetics –Part B: Cybernetics* 29: 601-618.

[INM01] Ishibuchi H., Nakashima T. y Murata T., (2001) Three-objective genetics-based machine learning for linguistic rule extraction. *Information Sciences* 136(1-4): 109-133.

[INN04] Ishibuchi H., Nakashima T. y Nii M. (2004) *Classification and modeling with linguistic information granules.* (Eds.) *Advanced Approaches to Linguistic Data Mining. Advanced Information Processing.* Springer, Berlin.

[INN11] Ishibuchi H., Nakashima Y. y Nojima Y. (2011) Performance evaluation of evolutionary multiobjective optimization algorithms for multiobjective fuzzy genetics-based machine learning. *Soft Computing, Special Issue on Evolutionary Systems* 15(12): 2415–2434.

[INT94] Ishibuchi H., Nozaki K. y Tanaka H. (1994) Construction of fuzzy classification systems with rectangular fuzzy rules using genetic algorithms. *Fuzzy Sets and Systems* 65: 237–253.

[INYT95] Ishibuchi H., Nozaki K., Yamamoto N. y Tanaka H. (1995) Selecting fuzzy if-then rules for classification problems using genetic algorithms. *IEEE Transactions on Fuzzy Systems* 3(3): 260-270.

[IY03] Ishibuchi H. y Yamamoto T. (2003) Trade-off between the number of fuzzy rules and their classification performance. En Casillas J., Cordón O., Herrera F. y Magdalena L. (Eds.) *Accuracy improvements in linguistic fuzzy modeling*, Studies in Fuzziness and Soft Computing, páginas 72-99. Springer-Verlag, Heidelberg, Alemania.

[IY04] Ishibuchi H. y Yamamoto T. (2004) Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining. *Fuzzy Sets and Systems* 141(1): 59-88.

[IYN05] Ishibuchi H., Yamamoto T. y Nakashima T. (2005) Hybridization of fuzzy gbml approaches for pattern classification problems. *IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics* 35(2): 359–365.

[JGSRB01] Jiménez F., Gómez-Skarmeta A., Roubus H. y Babuška R. (2001) A multi-objective evolutionary algorithm for fuzzy modeling. En *Proceedings of the 9th IFSA World Congress and the 20th NAFIPS International Conference*, páginas 1222-1228. Vancouver, Canadá.

[Jin00] Jin Y. (2000) Fuzzy modeling of high-dimensional systems: complexity reduction and interpretability improvement. *IEEE Transactions on Fuzzy Systems* 8(2): 212-221.

[Jin05] Jin Y. (2005) A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing* 9: 3-12.

[JL96] Jiang T. y Li Y. (1996) Generalized defuzzifications strategies and their parameter learning procedures. *IEEE Transactions on Fuzzy Systems* 4(1): 64-71.

[JLM14] Jarraya A., Leray P. y Masmoudi A. (2014) Discrete exponential bayesian networks: Definition, learning and application for density estimation. *Neurocomputing* 137: 142–149.

[JTL12] Jiang J. Y., Tsai S.C. y Lee S. J. (2012) FSKNN: multi-label text categorization based on fuzzy similarity and k nearest neighbors. *Expert Systems with Applications* 39(3): 2813–2821.

[KA05] Kaya M. y Alhajj R. (2005) Genetic algorithm based framework for mining fuzzy association rules. *Fuzzy Sets and Systems* 152(3): 587-601.

[Kar91] Karr C. (1991) Genetic algorithms for fuzzy controllers. *AI Expert* 6(2): 26-33.

[KC00] Knowles J. D. y Corne D. W. (2000) Approximating the non dominated front using the Pareto archived evolution strategy. *Evolutionary Computation* 8(2): 149-172.

[KCL02] Kim D., Choi Y. y Lee S. (2002) An accurate COG defuzzifier design using Lamarckian coadaptation of learning and evolution. *Fuzzy Sets and Systems* 130(2): 207-225.

[KKS85] Kiszka J., Kochanska M. y Sliwinska D. (1985) The influence of some fuzzy implication operators on the accuracy of a fuzzy model. Partes I y II *Fuzzy Sets and Systems* 15,15: 111-128, 223-240.

[KKWZ14] Karau H., Konwinski A., Wendell P. y Zaharia M. (2014) *Learning Spark: Lightning-Fast Big Data Analytics,* first ed. O'Reilly Media.

[KML99] Karnik N.N., Mendel J.M. y Liang Q. (1999) Type-2 fuzzy logic systems, *IEEE Transactions on Fuzzy Systems* 7(6): 643–658.

[Koc96] Koczy L. (1996) Fuzzy if… then rule models and their transformation into one another. *IEEE Transactions on Systems, Man, and Cybernetics* 26: 621-637.

[Kot13] Kotsiantis S. B. (2013) Decision trees: a recent overview. *Artificial Intelligence Review* 39:261–283.

[KS13] Kotlowski W. y Slowinski R. (2013) On nonparametric ordinal classification with monotonicity constraints. *IEEE Transactions on Knowledge and Data Engineering* 25(11): 2576–2589.

[Kun00] Kuncheva L. (2000) *Fuzzy classifier design*. Springer, Berlin.

[Lam11] Lam C. (2011) *Hadoop in Action*, first ed. Manning.

[LCT01] Liu B. D., Chen C. Y. y Tsao J. Y. (2001) Design of adaptative fuzzy logia controller based on linguistic hedge concepts and genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics* 31: 32-53.

[LdRBH14] Lopez V., del Rio S., Benitez J. M. y Herrera F. (2014) On the use of MapReduce to build linguistic fuzzy rule based classification systems for big data. En *IEEE World Congress on Computational Intelligence (WCCI 2014). IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2014),* páginas 1905–1912. Beijing (China).

[LdRBH15] Lopez V., del Rio S., Benitez J. M. y Herrera F. (2015) Cost-sensitive linguistic fuzzy rule based classification systems under the MapReduce framework for imbalanced big data. *Fuzzy Sets and Systems* 258: 5–38.

[Lee90] Lee C. C. (1990) Fuzzy logic in control systems: Fuzzy logic controller. Partes I y II *IEEE Transactions on Systems, Man and Cybernetics* 20 (2): 404-418, 419-435.

[LFG+13] Lopez V., Fernandez A., Garcia S., Palade V. y Herrera F. (2013) An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics. *Information Sciences* 250(20): 113–141.

[LFGH11] Luengo J., Fernandez A., Garcia S. y Herrera F. (2011) Addressing data complexity for imbalanced data sets: analysis of SMOTE–based oversampling and evolutionary undersampling. *Soft Computing* 15(10): 1909–1936.

[LFH13] Lopez V., Fernandez A. y Herrera F. (2013) Addressing covariate shift for genetic fuzzy systems classifiers: a case of study with FARC-HD for imbalanced datasets. En *Proceedings of the 2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2013)*, páginas 1–8.

[LFH14] Lopez V., Fernandez A. y Herrera F. (2014) On the importance of the validation technique for classification with imbalanced datasets: addressing covariate shift when data is skewed. *Information Sciences* 257: 1–13.

[LHH14] Lee W.-P., Hsiao Y.-T. y Hwang W.-C. (2014) Designing a parallel evolutionary algorithm for inferring gene networks on the cloud computing environment. *BMC Systems Biology* 8(1): 1–5.

[MA75] Mamdani E. H. y Assilian S. (1975) An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies* 7: 1-13.

[Mam74] Mamdani E. H. (1974) Applications of fuzzy algorithm for control a simple dynamic plant. *Proceedings of the IEE* 121(12): 1585-1588.

[MAS+15] Mohd Adnan M. R. H., Sarkheyli A., Mohd Zain A. y Haron H. (2015) Fuzzy logic for modeling machining process: a review. *Artificial Intelligence Review* 43(3):345–379.

[MF08] Mencar C. y Fanelli A. (2008) Interpretability constraints for fuzzy information granulation, *Information Sciences* 178(24): 4585–4618.

[MG95] Miller B. L. y Goldberg D. E. (1995) Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems* (9): 193-212.

[Mic96] Michalewicz Z. (1996) *Genetic algorithms + data structures = evolution programs*. Springer-Verlag.

[Miz89] Mizumoto M. (1989) Pictorial representations of fuzzy connectives, Part I: Cases of t-norms, t-conorms and averaging operators. *Fuzzy Sets and Systems* 31: 217-242.

[MMH97] Magdalena L. y Monasterio-Huelin F. (1997) A fuzzy logic controller with learning through the evolution of its knowledge base. *International Journal of Approximate Reasoning* 16(3): 335-358.

[MP12] Mahnot A. y Popescu M. (2012) FUMIL-fuzzy multiple instance learning for early illness recognition in older adults. En 2012 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), páginas. 1–5. Brisbane, Australia.

[MPH07] Márquez F. A., Peregrín A. y Herrera F. (2007) Cooperative evolutionary learning of linguistic fuzzy rules and parametric aggregation connectors for Mamdani fuzzy systems. *IEEE Transactions on Fuzzy Systems* 15(6): 1162-1178.

[MTSH12] Moreno-Torres J. G., Saez J. A. y Herrera F. (2012) Study on the impact of partitioninduced dataset shift on k-fold cross-validation, *IEEE Transactions on Neural Networks and Learning Systems* 23(8): 1304–1313.

[MTR+12] Moreno-Torres J. G., Raeder T., Alaiz-Rodriguez R., Chawla N. V. y Herrera F. (2012) A unifying view on dataset shift in classification. *Pattern Recognition* 45(1): 521–530.

[MV97] Magdalena L. y Velasco J. R. (1997) Evoutionary based learning of fuzzy controllers. En Pedrycz W. (Ed.) *Fuzzy Evolutionary Computation*, páginas 249-268. Kluwer Academic.

[Nau03] Nauck D. (2003) Measuring interpretability in rule-based classification systems. En *Proceedings of the 12th IEEE International Conference on Fuzzy Systems,* volumen 1(2), páginas 196–201.

[NK98] Nauck D. y Kruse R. (1998) How the learning of rule weights affects the interpretability of fuzzy systems. En *Proceedings of the 7th IEEE International Conference on Fuzzy Systems*, volume 2, páginas 1235-1240.

[Ped96] Pedrycz W. (ed.) (1996) *Fuzzy modeling: Paradigms and practice*. Kluwer Academic.

[PK99] Provost F. J. y Kolluri V. (1999) A survey of methods for scaling up inductive algorithms. *Data Mining and Knowledge Discovery* 3(2): 131–169.

[PK10] Pulkkinen P. y Koivisto H. (2010) A dynamically constrained multiobjective genetic fuzzy system for regression problems. *IEEE Transactions on Fuzzy Systems* 18(1): 161-177.

[PKL94] Park D., Kandel A. y Langholz G. (1994) Genetic-based new fuzzy reasoning models with application to fuzzy control. *IEEE Transactions on Systems, Man, and Cybernetics* 24(1): 39-47.

[PR10] Patel R. y Raghuwanshi M. M. (2010) Review on real coded genetic algorithms used in multiobjective optimization. In *3rd International Conference on Emerging Trends in Engineering and Technology*, páginas 610–613, Goa, India. IEEE.

[Qui86] Quinlan J. (1986) Introduction to decision tree. *Machine Learning* 1:81–106.

[RABH09] Robles I., Alcalá R., Benítez J. M. y Herrera F. (2009) Evolutionary parallel and gradually distributed lateral tuning of fuzzy rule-based systems, *Evolutionary Intelligence, Special Issue on GFS: New Advances* 2: 5-19.

[Roj96] Rojas R. (1996) *Neural networks: A systematic introduction*. Springer, Berlin

[Ros67] Rosenberg R. S. (1967) *Simulation of genetic populations with biochemical properties*. MA Thesis, Univ. Michigan. Ann Harbor, Michigan.

[RPF13] Rudas I. J., Pap E. y Fodor J. (2013) Information aggregation in intelligent systems: An application oriented approach. *Knowledge-Based Systems, Special Issue on Advances in Fuzzy Knowledge Systems: Theory and Application* 38: 3–13.

[Sam75] Sambuc R. (1975), Function Φ-flous, application a l'aide au diagnostic en pathologie thyroidienne, Ph.D. thesis, University of Marseille.

[Sch85] Schaffer J. D. (1985) Multiple objective optimization with vector evaluated genetic algorithms. En *Proceedings of first international conference on genetic algorithms*, páginas 93–100. Pittsburgh.

[SD94] Srinivas N., Deb K. (1994): Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation* 2: 221–248.

[SD08] Sivanandam S.N. y Deepa S.N. (2008) *Introduction to Genetic Algorithms*. Springer Berlin Heidelberg. ISBN 978-3-540-73189-4.

[SFBH10] Sanz J. A., Fernandez A., Bustince H. y Herrera F. (2010) Improving the performance of fuzzy rule-based classification systems with interval-valued fuzzy sets and genetic amplitude tuning, Inf. Sci. 180(19): 3674–3685.

[SFBH11] Sanz J., Fernandez A., Bustince H. y Herrera F. (2011) A genetic tuning to improve the performance of fuzzy rule-based classification systems with interval-valued fuzzy sets: degree of ignorance and lateral position, Int. J. Approx. Reason. 52(6): 751–766.

[SFBH13] Sanz J. A., Fernandez A., Bustince H. y Herrera F. (2013) IVTURS: a linguistic fuzzy rulebased classification system based on a new interval-valued fuzzy reasoning method with tuning and rule selection, IEEE Trans. Fuzzy Syst. 21(3): 399–411.

[SGGT12] Stavrakoudis D. G., Galidaki G. N., Gitas I. Z. y Theocharis J. B. (2012) Reducing the complexity of genetic fuzzy classifiers in highly-dimensional classification problems. *International Journal of Computational Intelligence Systems, Special Issue on Evolutionary Fuzzy Systems* 5(2): 254–275.

[She03] Sheskin D. J. (2003) *Handbook of Parametric and Nonparametric Statistical Procedures*. CRC Press, Boca Raton, FL.

[Shi14] Shi G. (2014) *Decision Trees*. Elsevier, San Diego, California, EE.UU. ISBN 978-0-12-410437-2.

[SK88] Sugeno M. y Kang G. T. (1988) Structure identification of fuzzy model. *Fuzzy Sets and Systems* 28: 15-33.

[SKK14] Sumathi R., Kirubakaran E. y Krishnamoorthy R. (2014) Multi class multi label based fuzzy associative classifier with genetic rule selection for coronary heart disease risk level prediction. *International Review on Computers and Software* 9(3): 533–540.

[SL96] Song Q. y Leland R. P. (1996) Adaptive learning defuzzification techniques and applications. *Fuzzy Sets and Systems* 8(31): 321-329.

[SLH11] Saez J. A., Luengo J. y Herrera F. (2011) Fuzzy rule based classification systems versus crisp robust learners trained in presence of class noise's effects: a case of study. En *11th International Conference on Intelligent Systems Design and Applications (ISDA)*, páginas 1229–1234.

[SLSH15] Saez J. A., Luengo J., Stefanowski J. y Herrera F. (2015) SMOTE-IPF: addressing the noisy and borderline examples problem in imbalanced classification by a resampling method with filtering. *Information Sciences* 291: 184–203.

[SMS07] Sanchez J. S., Mollineda R. A. y Sotoca J. M. (2007) An analysis of how training data complexity affects the nearest neighbor classifiers. *Pattern Analysis and Applications* 10(3): 189–201.

[SS15] Stulp F. y Sigaud O. (2015) Many regression algorithms, one unified model: A review. *Neural Networks* 69:60–79.

[Sun13] Sun S. (2013) A review of deterministic approximate inference techniques for Bayesian machine learning. *Neural Computing and Applications* 23:2039–2050.

[SY93] Sugeno M. y Yasukawa T. (1993) A fuzzy logic-based approach to qualitative modeling. *IEEE Transactions on Fuzzy Systems* 1(1): 7-31.

[Thr91] Thrift P. R. (1991) Fuzzy logic synthesis with genetic algorithms. En *Proceedings of the 4th International Conference on Genetic Algorithms* (ICGA'91), páginas 509-513. Morgan Kaufmann, San Mateo, CA, EE.UU.

[TS85] Takagi T. y Sugeno M. (1985) Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics* 15(1): 116-132.

[TV85] Trillas E. y Valverde L. (1985) On implication and indistinguishability in the setting of fuzzy logic. En Kacpryck J. y Yager R. (Eds.) *Management Decision Support Systems Using Fuzzy Sets and Possibility Theory*, páginas 198-212. Verlag TÜV Rheinland.

[UAG99] Uysal I. y Altay-Güvenir H. (1999). An overview of regression techniques for knowledge discovery. *The Knowledge Engineering Review* 14(4):319–340.

[Wan94] Wang L. X. (1994) *Adaptive fuzzy systems and control, design and stability analysis*. Prentice-Hall, Engleweed Cliffs, NJ, EE.UU.

[WCWW15] Wang S., Chung F. L., Wang J. y Wu J. (2015) A fast learning method for feedforward neural networks. *Neurocomputing* 149:295–307.

[WFP14] Wang R., Fleming P.J. y Purshouse R.C. (2014) General framework for localised multiobjective evolutionary algorithms. *Information Sciences* 258:29–53.

[WH07] Wagner C. y Hagras H. (2007) A genetic algorithm based architecture for evolving type-2 fuzzy logic controllers for real world autonomous mobile robots. En *Fuzzy Systems Conference, FUZZ-IEEE 2007. IEEE International,* páginas 1–6.

[Wil45] Wilcoxon F. (1945) Individual comparisons by ranking methods. *Biometrics* 1: 80–83.

[WKJ+05a] Wang H. L., Kwong S., Jin Y. C., Wei W. y Man K. F. (2005): Multiobjective hierarchical genetic algorithm for interpretable fuzzy rule-based knowledge extraction. *Fuzzy Sets and Systems* 149(1): 149–186.

[WKJ+05b] Wang H. L., Kwong S., Jin Y. C., Wei W. y Man K. F. (2005): Agentbased evolutionary approach for interpretable rule-based knowledge extraction. *IEEE Transactions on Systems, Man, and Cybernetics Part C: Application and Reviews* 35(2): 143–155.

[WM92] Wang L. X. y Mendel J. M. (1992) Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man, and Cybernetics* 22(6): 1414-1427.

[YF93] Yager R. y Filev D. P. (1993) SLIDE: a simple adaptive defuzzification method. *IEEE Transactions on Fuzzy Systems* 1: 69-78.

[YF94] Yager R. y Filev D. P. (1994) *Essentials of fuzzy modeling and control*. Wiley, New York.

[YL14] Yao X. y Liu Y. (2014) *Machine Learning*. Springer US. ISBN 978-1-4614-6939-1.

[Zad65] Zadeh L. A. (1965) Fuzzy sets. *Information and Control* 8: 338-353.

[Zad73] Zadeh L. A. (1973) Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics* 3: 28-44.

[Zad75] Zadeh L. A. (1975) The concept of a linguistic variable and its applications to approximate reasoning. Partes I, II y III *Information Science* 8, 8 y 9: 199-249, 301-357 y 43-80.

[ZEdR+11] Zikopoulos P. C., Eaton C., deRoos D., Deutsch T. y Lapis G. (2011) *Understanding Big Data – Analytics for Enterprise Class Hadoop and Streaming Data*. first ed., McGraw-Hill Osborne Media.

[ZDT00] Zitzler E., Deb K. y Thiele L. (2000) Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. Evolutionary Computation 8(2):173–195.

[ZG08] Zhou S. M. y Gan J. Q. (2008) Low-level interpretability and high-level interpretability: A unified view of data-driven interpretable fuzzy system modelling. *Fuzzy Sets and Systems* 159(23): 3091–3131.

[ZLSJ15] Zhang Y., Li Y., Sun J. y Ji J. (2015). Estimates on compressed neural networks regression. *Neural Networks* 63:10–17.

[ZLT01] Zitzler E., Laumanns M. y Thiele L. (2001) SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. En *Evolutionary methods for design, optimization and control with applications to industrial problems (EUROGEN´01)*, páginas 95–100. Barcelona, España.

[ZT99] Zitzler E. y Thiele L. (1999) Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions Evolutionary Computation* 3(4): 257–271.

[ZW04] Zhu X. y Wu X. (2004) Class noise vs. attribute noise: a quantitative study. *Artificial Intelligence Review* 22(3): 177–210.

[ZZHL12] Zhou Z.-H., Zhang M.-L., Huang S.-J. y Li Y.-F. (2012) Multi-instance multi-label learning. *Artificial Intelligence* 176(1): 2291–2320.