

UNIVERSIDAD DE GRANADA



Departamento de Ciencias de la Computación
e Inteligencia Artificial

*Modelado Lingüístico Flexible
Mediante el Uso de la
Representación de 2-Tuplas Lingüísticas*

Tesis Doctoral

Jesús Alcalá Fernández

Granada, Julio de 2006

Editor: Editorial de la Universidad de Granada
Autor: Jesús Alcalá Fernández
D.L.: Gr. 1275 - 2006
ISBN: 978-84-338-4014-1

UNIVERSIDAD DE GRANADA



**Modelado Lingüístico Flexible
Mediante el Uso de la
Representación de 2-Tuplas Lingüísticas**

MEMORIA QUE PRESENTA

Jesús Alcalá Fernández

PARA OPTAR AL GRADO DE DOCTOR EN INFORMÁTICA

Julio de 2006

DIRECTORES

Rafael Alcalá Fernández
Francisco Herrera Triguero

Departamento de Ciencias de la Computación
e Inteligencia Artificial

La memoria titulada “*Modelado Lingüístico Flexible Mediante el Uso de la Representación de 2-Tuplas Lingüísticas*”, que presenta D. Jesús Alcalá Fernández para optar al grado de doctor, ha sido realizada dentro del programa de doctorado “*Diseño, Análisis y Aplicaciones de Sistemas Inteligentes*” del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada bajo la dirección de los doctores D. Rafael Alcalá Fernández y D. Francisco Herrera Triguero.

Granada, Julio de 2006

El Doctorando

El Director

El Director

Fdo: D. Jesús Alcalá Fernández

Fdo: D. Rafael Alcalá Fernández

Fdo: D. Francisco Herrera Triguero

Agradecimientos

Dedico esta memoria a mi familia y amigos. A mis padres, de los que estoy orgulloso, por dedicarme toda una vida de esfuerzo y cariño. Como siempre he dicho, los incondicionales. A mi hermano y su mujer, por su ayuda en muchas y diferentes ocasiones, sus consejos y su forma de ver las cosas. A mi novia Zori, por su ayuda, por su comprensión y por ser la principal motivación y el sustento de todo este esfuerzo. Prometo devolverte el tiempo perdido. Y a mis amigos, en especial a Alberto, Imanol, Javi, Jorge, José y Roberto, por escucharme y animarme en los momentos más duros. A todos ellos, gracias por su gran PACIENCIA. Sin duda, deben quererme mucho.

Agradezco a mis directores, Rafael Alcalá y Francisco Herrera, el valioso tiempo dedicado y el cariño con el que siempre me han tratado. Sé que están muy ocupados, pero siempre han tenido tiempo para mí.

Quiero dar las gracias a los excompañeros y amigos de Mecenas, un abrazo y gracias por los buenos momentos vividos y los que nos quedan por vivir.

Y finalmente, mi agradecimiento a los compañeros y amigos de Jaén, en especial a Antonio Gabriel, por hacer del trabajo algo agradable, el justo contrapunto a los momentos más duros.

Según *Albert Einstein* (1879 - 1955), “*Hay dos formas de vivir la vida. Una es pensar que nada es un milagro. La otra es pensar que todo es un milagro*”. En mi caso, el principal milagro ha sido el poder contar con todas las personas mencionadas y algunas más que seguro se me quedan en el tintero. Sin todas y cada una de ellas, esta memoria no hubiera sido posible. Por ello,

GRACIAS A TODOS

Índice

Introducción	1
A Planteamiento	1
B Objetivos	4
C Resumen	6
1. Modelado Difuso Lingüístico Extendido: Mejoras a la Precisión	9
1.1. Modelado de Sistemas Mediante Lógica Difusa: Modelado Difuso Lingüístico versus Modelado Difuso Preciso	10
1.1.1. Modelado de Sistemas Mediante Sistemas Basados en Re- glas Difusas	11
1.1.2. Modelado Difuso Lingüístico y Modelado Difuso Preciso . .	13
1.2. Modelado Difuso Lingüístico Extendido: Mejoras a la Precisión . .	17
1.2.1. Mejoras en el Proceso de Derivación del Modelo Difuso Lingüístico	19
1.2.1.1. Métodos Basados en la Definición de la Base de Datos	19
1.2.1.1.1. Integración del Diseño de la Base de Datos en el Proceso de Derivación Glo- bal del Modelo Difuso Lingüístico	20
1.2.1.1.2. Aprendizaje/Ajuste de las Funciones de Pertenencia	21

1.2.1.1.3.	¿Cómo Mantener la Interpretabilidad del Modelo Difuso Lingüístico al Diseñar la Base de Datos?	25
1.2.1.2.	Métodos de Generación de Reglas Orientados a Mejorar la Cooperación	27
1.2.2.	Extensiones en la Estructura del Modelo Difuso Lingüístico	29
1.2.2.1.	Uso de Modificadores Lingüísticos	29
1.2.2.2.	Uso de más de un Consecuente para cada Regla	30
1.2.2.3.	Uso de Reglas Ponderadas	31
1.2.2.4.	Uso de las 2-Tuplas Lingüísticas	31
1.2.3.	Interpretabilidad del Modelado Difuso Lingüístico Extendido	32
1.3.	Selección de Reglas Lingüísticas con Algoritmos Genéticos: Estudio, Análisis y Ejemplos	36
1.3.1.	Estudio de Diferentes Técnicas y Enfoques Evolutivos para la Selección de Reglas Lingüísticas	38
1.3.2.	Selección Genética a Partir de un Conjunto de Reglas Candidatas	39
1.3.3.	Selección Genética de Bases de Reglas Integrando Modificadores Lingüísticos para Modificar las Funciones de Pertenencia	42
1.3.3.1.	Un Proceso de Multiselección Considerando Modificadores Lingüísticos	43
1.3.3.2.	Adaptación del Método Básico de Selección Genética	44
1.3.3.3.	Algoritmo para el Proceso de Multiselección Genética	48
1.3.3.4.	Extensión Paramétrica	49
1.3.4.	Un Ejemplo Simple de Selección de Reglas	49
1.4.	Ajuste Lateral de Funciones de Pertenencia: Estudio, Análisis y Ejemplos	51

1.4.1. Traducción Simbólica de una Etiqueta 52

1.4.2. Representación de Regla Basada en 2-Tuplas Lingüísticas . 52

1.4.3. Sistema de Inferencia Difusa 55

1.4.4. Principales Diferencias entre el Ajuste Clásico y el Ajuste Lateral 55

1.4.5. Algoritmo Genético para el Ajuste Lateral 56

1.4.6. Un Ejemplo Simple de Ajuste Lateral de Funciones de Pertenencia 58

1.5. Descripción de las Propuestas Consideradas en los Sigüientes Capítulos 60

2. Aprendizaje Automático de Sistemas Basados en Reglas Difusas Precisos y Compactos Basados en la Representación de 2-Tuplas Lingüísticas 63

2.1. Un Nuevo Sistema de Inferencia Difusa 65

2.2. Algoritmo Evolutivo para el Aprendizaje de la Base de Conocimiento 66

2.2.1. Obtención de la Base de Conocimiento mediante un Proceso de Aprendizaje A Priori de la Base de Datos 66

2.2.2. Algoritmo Evolutivo (Enfoque CHC) 68

2.2.3. Codificación de la Base de Datos 69

2.2.4. Evaluación de los Cromosomas 70

2.2.5. Población Inicial 70

2.2.6. Operadores de Cruce 71

2.2.7. Mecanismo de Reinicialización 72

2.3. Dos Nuevos Métodos *Ad-Hoc* de Generación de Reglas Guiados por Ejemplos y su Integración en el Aprendizaje Evolutivo de la BD A Priori 73

2.3.1. Método de Generación de Reglas Basado en Medias (Algoritmo AV) 75

2.3.2. Método de Generación de Reglas Basado en Moda (Algoritmo MO) 76

2.3.3.	Integración de los Métodos Propuestos en el Aprendizaje Evolutivo de la BC	77
2.4.	Estudio Experimental	78
2.4.1.	Experimentos	79
2.4.2.	Estimación de la Longitud de las Líneas de Baja Tensión	81
2.4.3.	Estimación del Costo de Mantenimiento de la Línea de Media Tensión	83
2.5.	Sumario	85
3.	Ajuste Evolutivo basado en la Representación de 3-Tuplas Lingüísticas: Ajuste-LA	89
3.1.	Ajuste Lateral y de Amplitud de Funciones de Pertenencia	91
3.1.1.	Sistema de Inferencia Difusa	94
3.1.2.	Equilibrio entre Interpretabilidad y Precisión	95
3.2.	Aplicación sobre Reglas Obtenidas a Partir de Ejemplos	96
3.2.1.	Algoritmo Evolutivo para el Ajuste Lateral y de Amplitud de Funciones de Pertenencia	96
3.2.1.1.	Modelo Evolutivo (Enfoque CHC)	97
3.2.1.2.	Esquema de Codificación y Población Inicial	97
3.2.1.3.	Evaluación de los Cromosomas	98
3.2.1.4.	Operador de Cruce	99
3.2.1.5.	Mecanismo de Reinicialización	99
3.2.2.	Selección de Reglas y Ajuste Lateral y de Amplitud de Funciones de Pertenencia	100
3.2.2.1.	Sinergia Positiva entre Ambos Enfoques	100
3.2.2.2.	Algoritmo Evolutivo	102
3.2.3.	Estudio Experimental	105
3.2.3.1.	Experimentos	105
3.2.3.2.	Estimación de la Longitud de las Líneas de Baja Tensión	107

3.2.3.3.	Estimación del Costo de Mantenimiento de la Línea de Media Tensión	111
3.2.3.4.	Análisis sobre el Uso del Nuevo Sistema de Inferencia	113
3.3.	Aplicación sobre Reglas Proporcionadas por un Experto	116
3.3.1.	El Problema de Control de Sistemas HVAC	117
3.3.1.1.	Objetivos y Función de Adaptación	119
3.3.1.2.	Variables y Arquitectura del Controlador Difuso para el sistema HVAC	120
3.3.2.	Aplicación del Ajuste Lateral y de Amplitud al Controlador Difuso del Sistema HVAC	120
3.3.2.1.	Modelo Evolutivo: Enfoque Estacionario	122
3.3.2.2.	Algoritmo Genético para el Ajuste Lateral y de Amplitud	122
3.3.2.2.1.	Esquema de Codificación y Población Inicial	123
3.3.2.2.2.	Operadores Genéticos	124
3.3.2.2.3.	Evaluación de los Cromosomas	124
3.3.2.3.	Algoritmo Genético para el Ajuste Lateral y de Amplitud junto con la Selección de Reglas	126
3.3.2.3.1.	Esquema de Codificación y Población Inicial	127
3.3.2.3.2.	Operadores Genéticos	127
3.3.2.3.3.	Mecanismo de Reinicialización	128
3.3.3.	Experimentos y Análisis de Resultados	128
3.3.3.1.	Estudio sobre los Métodos Basados en el Ajuste Lateral y de Amplitud	130
3.3.3.2.	Análisis Considerando los Resultados Obtenidos por otros Métodos	133
3.4.	Sumario	135

Comentarios Finales	137
A. Resumen y Conclusiones	137
B. Publicaciones Asociadas a la Tesis	140
C. Líneas de Investigación Futuras	141
Apéndices	145
A. Introducción a los Sistemas Basados en Reglas Difusas	147
A.1. Grafos Difusos para Modelado de Sistemas	147
A.2. Sistemas Basados en Reglas Difusas Lingüísticos	151
A.3. Sistemas Basados en Reglas Difusas Aproximativos	152
A.4. Sistemas Basados en Reglas Difusas de Tipo Takagi-Sugeno-Kang .	153
A.5. Análisis de los Componentes de los SBRDs Lingüísticos	156
A.5.1. La Base de Conocimiento	156
A.5.2. La Interfaz de Fuzzificación	157
A.5.3. El Sistema de Inferencia	158
A.5.4. La Interfaz de Defuzzificación	160
B. Algoritmos Genéticos	163
B.1. Introducción	163
B.2. Representación de las Soluciones	165
B.3. El Mecanismo de Selección	166
B.4. El Operador de Cruce	167
B.5. El Operador de Mutación	169
C. Sistemas Basados en Reglas Difusas Evolutivos	171
D. Problemas de Modelado Considerados	175
D.1. Aspectos Generales de la Experimentación	175

D.2. Descripción de las Aplicaciones Consideradas 177

 D.2.1. Estimación de la Longitud de las Líneas de Baja Tensión . 178

 D.2.2. Estimación del Costo de Mantenimiento de la Línea de Media Tensión 179

 D.2.3. Sistemas de Ventilación, Calefacción y Aire Acondicionado (*HVAC Systems*) 181

 D.2.3.1. El Problema del Control de Sistemas HVAC . . . 182

 D.2.3.1.1. Parámetros de Control o Explícitos: Variables del Controlador 183

 D.2.3.1.2. Parámetros Controlados o Implícitos: Objetivos del Problema 184

 D.2.3.2. Control Difuso de Sistemas HVAC 185

 D.2.3.3. El Módulo de Pruebas de ATC 187

 D.2.3.3.1. Objetivos y Función de Adaptación . . . 189

 D.2.3.3.2. Variables y Arquitectura del Controlador Difuso 190

Bibliografía

Índice de figuras

1.1. Modelado de Sistemas mediante SBRDs: interpretabilidad y flexibilidad dependiendo del tipo de SBRD considerado	15
1.2. Equilibrio entre flexibilidad y simplicidad: ML extendido y MD restringido	16
1.3. Definición de funciones de pertenencia alterando sus parámetros básicos	22
1.4. Definición de funciones de pertenencia mediante modificadores lingüísticos	23
1.5. Efectos provocados por un escalado no lineal sobre una partición difusa con funciones de pertenencia triangulares	24
1.6. Pérdida de interpretabilidad al diseñar la BD	25
1.7. Efectos de los modificadores lingüísticos ‘muy’ y ‘más-o-menos’ . .	30
1.8. Translación simbólica de una etiqueta lingüística	32
1.9. Pérdida de interpretabilidad dependiendo de la extensión realizada en el MDL	33
1.10. Pérdida de interpretabilidad al hacer una modificación brusca de las funciones de pertenencia	35
1.11. Modificadores lingüísticos de tipo I	45
1.12. Modificadores lingüísticos de tipo II	46
1.13. Translación simbólica de una etiqueta lingüística y desplazamiento lateral de la función de pertenencia asociada	52
1.14. Esquema de CHC	58

2.1. Esquema de aprendizaje de la BC	67
2.2. Esquema del algoritmo considerando el enfoque CHC	69
2.3. Esquema del comportamiento de los operadores BLX y PCBLX	71
2.4. BD con/sin desplazamientos laterales (negro/gris), BR y parámetros de desplazamiento de un modelo obtenido por el algoritmo GLD+MO (las etiquetas no utilizadas han sido eliminadas en la figura)	83
2.5. BD con/sin desplazamientos laterales (negro/gris), BR y desplazamientos de un modelo obtenido por el algoritmo GLD+MO (las etiquetas no utilizadas han sido eliminadas en la figura)	86
3.1. Variación lateral y de amplitud de la función de pertenencia asociada a la etiqueta s_2	92
3.2. Ejemplo del esquema de codificación considerando el ajuste-LA y la selección de reglas: (a) Enfoque Global; (b) Enfoque Local	103
3.3. Ámbito de aplicación de los operadores	104
3.4. BD inicial, BD ajustada y BR de un modelo obtenido con el método GLA+S en el problema de estimación de la longitud de las líneas de baja tensión (las etiquetas no utilizadas han sido eliminadas en la figura)	110
3.5. BR y desplazamientos laterales de un modelo obtenido con el método LLA+S en el problema de estimación de la longitud de las líneas de baja tensión	110
3.6. BD inicial, BD ajustada y BR de un modelo obtenido con el método GLA en el problema de estimación del costo de mantenimiento de la línea de media tensión	113
3.7. BD inicial, BD ajustada y BR de un modelo obtenido con el método GLA+S en el problema de estimación del costo de mantenimiento de la línea de media tensión	114
3.8. Estructura genérica de un sistema HVAC para un edificio de oficinas	118
3.9. BR inicial y estructura del controlador difuso de verano de GENESYS	121
3.10. BD del controlador difuso de verano de GENESYS	121

3.11. Organigrama del algoritmo genético propuesto	123
3.12. $\delta_i(x)$ cuando $g_i \leq i_i$	125
3.13. $\delta_i(x)$ cuando $g_i > i_i$	126
3.14. BD inicial y ajustada del modelo obtenido con GLA+S (semilla1) .	132
3.15. BR y estructura final del modelo obtenido con GLA+S (semilla1) .	133
A.1. Representación aproximada de relaciones con grafos difusos res- tringidos y sin restricciones	149
A.2. Paralelismo entre SBRDs lingüísticos y SBRDs aproximativos . . .	150
A.3. Estructura básica de un SBRD lingüístico	151
A.4. Comparación gráfica entre BC lingüísticas y BRD aproximativas .	154
A.5. Estructura básica de un sistema basado en reglas difusas TSK . . .	155
B.1. Estructura básica de un algoritmo genético	165
B.2. Ejemplo de aplicación del mecanismo de selección	167
B.3. Ejemplo de aplicación del operador de cruce simple en un punto .	168
B.4. Ejemplo de aplicación del operador de mutación	169
C.1. Sistemas basados en reglas difusas evolutivos	172
D.1. Representación gráfica de posibles particiones difusas con 3, 5 o 7 etiquetas lingüísticas	176
D.2. (a) Dependencia (X_1, Y) y (X_2, Y) en el conjunto de entrenamiento; (b) Dependencia (X_1, Y) y (X_2, Y) en el conjunto de prueba	180
D.3. Estructura genérica de un sistema HVAC para un edificio de oficinas	182
D.4. Estructura genérica de un controlador difuso	186
D.5. Representación y principales características de los módulos de ATC	188
D.6. BR inicial y estructura del controlador difuso de verano de ATC .	191
D.7. BD del controlador difuso de verano de ATC	192

Índice de tablas

1.1. Resultados obtenidos considerando una aproximación simple para la selección de reglas difusas	50
1.2. Resultados obtenidos considerando una aproximación simple para el ajuste lateral de funciones de pertenencia	59
2.1. Métodos considerados para el estudio experimental	79
2.2. Resultados obtenidos en el problema de estimación de la longitud de las líneas con el parámetro $\alpha = 1$ para la función de adaptación	82
2.3. Resultados obtenidos en el problema de estimación del costo de mantenimiento	84
3.1. Métodos considerados para la comparación	106
3.2. Resultados obtenidos en el problema de estimación de la longitud de las líneas eléctricas de baja tensión	108
3.3. Resultados obtenidos en el problema de estimación del costo de mantenimiento de la línea de media tensión	112
3.4. Datos no cubiertos	115
3.5. Resultados obtenidos en el problema de estimación de la longitud de las líneas eléctricas de baja tensión	116
3.6. Controlador inicial y parámetros usados en la función de adaptación (F')	129
3.7. Métodos considerados en la comparación	130
3.8. Resultados obtenidos por los métodos	131

3.9. Métodos considerados para la comparación	134
3.10. Comparación entre los diferentes métodos	135
D.1. Principales características de las aplicaciones consideradas en esta memoria	178
D.2. Notación y universos de discurso en el problema de la estimación de la longitud de las líneas de baja tensión	179
D.3. Notación y universos de discurso en el problema de mantenimiento de las líneas de media tensión	180

Tabla de Acrónimos

AG	—	Algoritmo Genético	29
BC	—	Base de Conocimiento	3
BD	—	Base de Datos	3
BEMS	—	Sistema de Administración de Energía en Edificios (<i>Building Energy Management Systems</i>)	117
BR	—	Base de Reglas	2
BRD	—	Base de Reglas Difusas	14
COR	—	Metodología de COoperación entre Reglas	79
DNF	—	Forma Normal Disyuntiva	27
ECM	—	Error Cuadrático Medio	70
HVAC	—	Sistema de Ventilación, Calefacción y Aire Acondicionado (<i>Heating, Ventilating, and Air Conditioning</i>)	6
FITA	—	Primero Inferir, Después Agregar	55
MDL	—	Modelado Difuso Lingüístico	2
MDLE	—	Modelado Difuso Lingüístico Extendido	2
MDP	—	Modelado Difuso Preciso	2
MDPR	—	Modelado Difuso Preciso Restringido	2
MM	—	Método <i>ad-hoc</i> guiado por ejemplos (Método Mixto)	37
PCBLX	—	Operador de cruce <i>Parent Centric</i> BLX	71
PMV	—	Índice global de confort térmico PMV, ISO 7730 (<i>Predicted Mean Vote index</i>)	119
SBRD	—	Sistema Basado en Reglas Difusas	1
SBRDE	—	SBRD Evolutivo	1
SDEs	—	Sistemas Difusos Evolutivos	57
WM	—	Método <i>ad-hoc</i> guiado por ejemplos de Wang y Mendel	49

Introducción

A Planteamiento

El desarrollo de modelos matemáticos de sistemas reales es un tópico importante en diferentes disciplinas como la ingeniería y las ciencias. La disciplina de investigación que estudia la generación automática de modelos se denomina habitualmente *Identificación de Sistemas* o *Modelado de Sistemas* [Nel01]. Los modelos obtenidos en el marco de este área de trabajo sirven para resolver problemas reales a través de simulaciones, analizar el comportamiento de sistemas, diseñar nuevos procesos para controlarlos, realizar predicciones, etc. De este modo, el desarrollo inadecuado de dichos modelos puede conducir a la obtención de resultados no exitosos y erróneos sobre los sistemas en cuestión.

En la actualidad, gran parte de estos sistemas comparten una serie de características que dificultan su modelado con técnicas tradicionales, tales como: necesidad de una importante precisión, tratamiento y respuestas en tiempo real, carencia de conocimiento formal sobre su funcionamiento, y posesión de un comportamiento fuertemente no lineal, con alto grado de incertidumbre y con una dinámica que varía con el tiempo. Ejemplos de esta clase de problemas son los procesos complejos de los sistemas de ingeniería aeroespacial o bioquímica, aunque también se encuentran en sistemas ecológicos, sociales o correspondientes al área económico-financiera.

En este sentido, la construcción de modelos basados en la Lógica Difusa [Zad65, Zad73] ha adquirido un gran interés gracias a que proporciona más generalidad, poder expresivo y tolerancia frente a la imprecisión. El modelado de sistemas mediante Sistemas Basados en Reglas Difusas (SBRDs) se puede con-

siderar como una vía para modelar un sistema haciendo uso de un lenguaje de descripción basado en la Lógica Difusa con predicados difusos [SY93]. Tal como veremos a lo largo de esta memoria, es posible llevar a cabo varios tipos de modelado empleando distintos tipos de SBRDs, dependiendo del grado de *interpretabilidad* y *precisión* que deseemos que tenga el futuro modelo. Desafortunadamente, estos dos requisitos, el grado de interpretabilidad del modelo para el ser humano y el grado en que ese modelo se aproxima al sistema real, suelen estar enfrentados.

La esencia original de la Lógica Difusa queda recogida en los Sistemas Basados en Reglas Lingüísticas (también conocidos como de tipo Mamdani [Mam74, MA75]), que permiten identificar sistemas obteniendo modelos claramente interpretables por el ser humano. Constituyen un tipo de modelado denominado *Modelado Difuso Lingüístico* (MDL) que, sin embargo, presenta un problema fundamental: su falta de precisión al modelar algunos sistemas muy complejos. Este hecho se debe a la inflexibilidad del concepto de variable lingüística (el tipo de variable empleada [Zad75]), que impone unas fuertes restricciones a la estructura de regla difusa considerada [Bas94]. Por otro lado, en problemas multidimensionales complejos con relaciones de entrada-salida fuertemente no lineales, la Base de Reglas (BR) del modelo puede presentar muchas reglas redundantes, inconsistentes y contradictorias, que provocan que tanto el rendimiento del modelo difuso lingüístico como su interpretabilidad disminuyan.

La alternativa al MDL la constituye el *Modelado Difuso Preciso* (MDP), que se implementa mediante los SBRDs aproximativos [ACCH01a, BD95, BH99, CFM96, CH95, CH97, Koc96] y los de tipo Takagi-Sugeno-Kang [SK88, TS85]. Debido a la estructura de regla difusa considerada por estos dos tipos de sistemas, es posible obtener modelos más precisos, aunque se pierde gran parte de la interpretabilidad asociada al modelo, lo que no es recomendable en muchos casos.

De este modo, la investigación del área en los últimos años ha derivado hacia dos líneas de trabajo complementarias: i) dotar de mayor precisión al MDL, sin perder su interpretabilidad en gran medida [CCHM03a], y ii) dotar de mayor interpretabilidad al MDP, sin disminuir mucho su grado de precisión [CCHM03c]. Esto ha dado lugar a la aparición de dos nuevas áreas en el Modelado Difuso, el *Modelado Difuso Lingüístico Extendido* (MDLE) y el *Modelado Difuso Preciso Restringido* (MDPR).

Centrándonos en la primera de ellas, que será el objeto de estudio de esta

memoria de tesis, se han propuesto distintas posibilidades para mejorar el MDL en la literatura especializada [CCHM03b]. Todos estos enfoques comparten la idea de mejorar la forma en que el modelo difuso lingüístico realiza el razonamiento interpolativo induciendo una mejor cooperación entre las reglas de su base. Esta cooperación entre reglas puede ser promovida a tres niveles, actuando sobre tres componentes distintos del modelo difuso: la Base de Datos (BD), la BR y la Base de Conocimiento (BC) al completo. En esta memoria nos centraremos en los enfoques que actúan sobre la BD. A modo de resumen, que será posteriormente extendido en el primer capítulo de la memoria, mencionaremos los distintos enfoques existentes para el diseño de la BD dependiendo del modo en que se integren en el proceso de derivación:

- *Diseño preliminar [LMV99, Ped01]*: Implica extraer la BD a priori (es decir, antes de derivar las reglas del modelo difuso lingüístico) por inducción a partir del conjunto de ejemplos disponible.
- *Diseño a priori [CHV00, CHV01a, CHMV01, Glo96, IM96, PRO⁺00, RPOP00]*: Esta técnica consiste en disponer de un método de aprendizaje sencillo que diseña, a partir de una BD determinada, otros componentes distintos de ésta (p.ej., la BR). Mediante un proceso de meta-aprendizaje, el método genera distintas BDs y valida su eficacia ejecutando el método de aprendizaje básico.
- *Diseño simultáneo [GJ99, HM95, JvSS99, KS95, MMH97, WHT98, XL00]*: El proceso de diseño de la BD se realiza conjuntamente con otros componentes tales como la BR en un mecanismo simultáneo.
- *Diseño a posteriori [BKC96, CdJH98, CH97, GP99, HLV95, Kar91a, LCT01, LM94, NK99, SKG93]*: Este enfoque, conocido como *ajuste*, consiste en refinar la BD a partir de una definición previa una vez que el resto de componentes ha sido obtenido.

Uno de los enfoques más ampliamente utilizados para mejorar la precisión del MDL es el diseño a posteriori de la BD. En [CCdJH05] podemos encontrar un estudio comparativo entre distintas técnicas de ajuste con las que se obtienen modelos más precisos que los obtenidos mediante un proceso de aprendizaje simple. Recientemente en [AH04] se presentó una nueva técnica de ajuste donde se considera una nueva representación de las reglas lingüísticas para realizar un ajuste lateral evolutivo de las funciones de pertenencia que definen la BD. Este enfoque

se basa en el modelo de representación de 2-tuplas lingüísticas [HM00] que permite realizar un desplazamiento lateral de las etiquetas considerando un único parámetro por etiqueta. De esta forma se alcanzan dos objetivos principales:

- obtener etiquetas que contienen un conjunto de ejemplos con mejor cubrimiento (búsqueda de precisión) y que conservan su forma original, y
- reducir el espacio de búsqueda del proceso de ajuste respecto al ajuste clásico [BOOB02, CHHM01, Gür99, Gür00, HLV95, Jan93, Kar91a], que normalmente considera tres parámetros en el caso de funciones de pertenencia triangulares, facilitando la obtención de modelos óptimos.

Tanto los modelos lingüísticos clásicos como los extendidos se obtienen generalmente mediante un proceso de aprendizaje automático a partir de un conjunto de pares de datos entrada-salida o simulaciones que reflejan el comportamiento del sistema real que se desea modelar. Existe una gran cantidad de métodos de este tipo que combinan distintos paradigmas de aprendizaje y ajuste [ACC⁺00, CCHM03b]. Entre todos ellos, los SBRDs Evolutivos (SBRDEs) destacan por su capacidad para trabajar con diferentes tipos de estructuras solución (modelos) e incluir conocimiento experto [CHHM01]. Además, son capaces de resolver problemas complejos y han presentado buenos resultados en una gran cantidad de problemas.

En esta memoria centramos nuestra atención en esta nueva representación de 2-tuplas lingüísticas con la intención de diseñar dos nuevos SBRDEs, uno para aprendizaje y otro para ajuste de las funciones de pertenencia, con el objetivo principal de mejorar la precisión del MDL dentro del deseado equilibrio entre interpretabilidad y precisión.

B Objetivos

El principal objetivo de esta memoria es mejorar la precisión del MDL manteniendo tanto como sea posible la interpretabilidad de los modelos obtenidos. Para ello, nuestro objetivo se centra en el desarrollo de un método nuevo de aprendizaje y otro de ajuste evolutivo de modelos lingüísticos flexibles basándonos en la *representación de 2-tuplas lingüísticas* y en el *ajuste lateral de las funciones de pertenencia*.

En concreto, el objetivo propuesto se subdivide en los siguientes:

- *Extensión del MDL mediante un enfoque que permita aprender la BC completa a través de un proceso de aprendizaje a priori de la BD.* En esta línea, podríamos realizar un diseño a priori de la BD para obtener la BC completa basándonos en la representación de 2-tuplas lingüísticas. Esta representación reduce el espacio de búsqueda al permitir ajustar cada etiqueta considerando un único parámetro por etiqueta, consiguiéndose así una mejora potencial de la precisión y manteniendo la interpretabilidad a un alto grado.
- *Mejoras en el nuevo mecanismo de inferencia.* Hay ocasiones en las que al buscar modelos con el menor número de reglas posibles (modelos lingüísticos compactos) podría ocurrir que hubieran zonas no cubiertas en el espacio de entrada. De esta forma, no podemos asegurar el buen comportamiento del modelo obtenido para datos de prueba no cubiertos (es decir, la generalización del modelo lingüístico final podría no ser buena para entradas no cubiertas). En este sentido, se pretende diseñar un mecanismo de inferencia nuevo para la consideración de puntos no cubiertos que ayude a la generalización y al tratamiento de puntos de ruido en problemas con pocos datos, mal distribuidos o con ruido.
- *Extensión de la representación de 2-tuplas lingüísticas.* En esta memoria, nos plantearemos flexibilizar la estructura del modelo lingüístico extendiendo la representación de 2-tuplas lingüísticas a una representación que nos permita también variar la amplitud del soporte de las funciones de pertenencia. El principal objetivo de esta propuesta será el de incidir en la estructura del modelo para hacer que los resultados sean más precisos y que la interpretabilidad se mantenga tanto como sea posible.
- *Extensión del MDL mediante un ajuste basado en las 3-tuplas lingüísticas.* El uso de un ajuste basado en la nueva representación de 3-tuplas lingüísticas permite determinar la especificidad de una etiqueta al modificar la amplitud del soporte de las funciones de pertenencia, lo que implica una mejora potencial en la precisión del sistema. Para aprovechar dicha mejora se plantea desarrollar un nuevo método de ajuste capaz de trabajar con esta nueva representación.
- *Mejoras en la interpretabilidad combinando el ajuste basado en 3-tuplas lingüísticas y la selección de reglas.* Se pretende integrar dos técnicas que

persiguen, de manera complementaria, mejorar la precisión y la interpretabilidad del MDL. De esta manera, el ajuste consigue incrementar la precisión de los modelos lingüísticos obtenidos a partir de métodos básicos de MDL mientras que la selección de reglas mejora la interpretabilidad al obtener modelos más simples, que faciliten la legibilidad del mismo.

- *Aplicación del ajuste basado en 3-tuplas lingüísticas a un problema complejo de control.* Se pretende aplicar el ajuste basado en 3-tuplas lingüísticas y su combinación con un mecanismo de selección de reglas al diseño de controladores difusos óptimos dedicados al control de Sistemas de Ventilación, Calefacción y Aire Acondicionado —*Heating, Ventilating, and Air Conditioning* (HVAC) —. En este tipo de problemas, las reglas se obtienen usualmente a partir de expertos, considerando reglas generales basadas en la experiencia y no siempre compatibles con los requisitos del sistema. Nos proponemos refinar un controlador difuso obtenido a partir de expertos con el principal objetivo de mejorar el rendimiento energético manteniendo el confort. A su vez, las reglas obtenidas deben seguir manteniendo cierta interpretabilidad a los ojos de dichos expertos. El problema que se nos plantea presenta una elevada complejidad (gran número de variables y optimización de múltiples criterios), lo que nos permitirá evaluar la potencialidad de la aplicación conjunta del ajuste lateral y de amplitud y la selección de reglas.

C Resumen

Para desarrollar los objetivos propuestos, la memoria está organizada en tres capítulos, una sección de comentarios finales y tres apéndices. La estructura de cada una de estas partes se introduce brevemente a continuación.

En el Capítulo 1, realizaremos un estudio sobre los diferentes enfoques existentes en la literatura para abordar el equilibrio entre interpretabilidad y precisión desde distintos ángulos, tomando como ejemplo el ajuste lateral de funciones de pertenencia y varios métodos para la selección de reglas. Finalmente, se propondrá el enfoque que seguiremos en los siguientes capítulos.

En el Capítulo 2, proponemos un nuevo método de aprendizaje de la BC completa mediante un aprendizaje evolutivo a priori de la BD y basado en la representación de regla de 2-tuplas lingüísticas. Para ello, el capítulo se estruc-

tura en cuatro grandes secciones. En la primera sección, se propone un nuevo mecanismo de inferencia para la consideración de puntos no cubiertos que ayude a la generalización y al tratamiento de puntos de ruido en problemas con pocos datos, mal distribuidos o con ruido. En la segunda, se presenta el esquema de aprendizaje y el algoritmo evolutivo específico propuesto para obtener las BCs basado en la representación de regla de 2-tuplas lingüísticas. Y en la tercera, se discute sobre qué tipo de métodos para la generación de la BR podría favorecer un aprendizaje correcto de las funciones de pertenencia, proponiendo dos métodos nuevos *ad-hoc* guiados por ejemplos diseñados específicamente para esta tarea. Finalmente, se presentan diferentes resultados experimentales.

En el Capítulo 3, se presenta el ajuste basado en 3-tuplas lingüísticas y su combinación con la selección de reglas. Para ello, inicialmente se describe el nuevo modelo de representación basado en 3-tuplas lingüísticas, analizando su influencia en el equilibrio entre interpretabilidad y precisión. El proceso de ajuste se aplicará a la optimización de modelos lingüísticos obtenidos a partir de métodos básicos de MDL así como a la optimización de controladores con reglas lingüísticas obtenidas a partir de expertos. El método básico y su combinación con la selección de reglas también será aplicado para optimizar controladores difusos para sistemas de climatización HVAC en grandes edificios. Debido a que el tiempo de evaluación de una solución para este tipo de problemas es muy elevado, modificaremos el modelo evolutivo del algoritmo básico.

Finalmente, se ha incluido una sección de “Comentarios Finales”, que resume los resultados obtenidos en esta memoria, presentando algunas conclusiones sobre éstos. Además, se comentan algunos aspectos sobre trabajos futuros.

También se incluyen cuatro apéndices, dedicados respectivamente a introducir los SBRDs, los SBRDEs y los problemas reales utilizados en los Capítulos 1, 2 y 3. Finalizamos la memoria con una recopilación bibliográfica que recoge las contribuciones más destacadas en la materia estudiada.

Capítulo 1

Modelado Difuso Lingüístico Extendido: Mejoras a la Precisión

En la actualidad, el Modelado de Sistemas es una de las aplicaciones más importantes en el campo de los SBRDs [Mam74, MA75, Ped96, Zad73]. El Modelado Difuso puede considerarse como un enfoque utilizado para modelar un sistema haciendo uso de un lenguaje descriptivo basado en la Lógica Difusa [Zad65, Zad73] con predicados difusos [SY93]. En este tipo de modelado, la *precisión* y la *interpretabilidad* del modelo obtenido son propiedades contradictorias directamente dependientes del proceso de aprendizaje y/o de la estructura del modelo. Tradicionalmente, de acuerdo a la estructura de regla considerada, es posible distinguir entre dos enfoques de Modelado Difuso claramente contrapuestos: *MDL* y *MDP*, con la interpretabilidad y la precisión del modelo como principal requisito, respectivamente.

El MDL permite identificar sistemas obteniendo modelos claramente interpretables por el ser humano. Sin embargo, presenta un problema fundamental, su falta de precisión al modelar sistemas complejos. En cambio, el MDP posibilita la obtención de modelos más precisos, pero se pierde gran parte de la interpretabilidad asociada a éstos. De este modo, se plantea un vacío significativo entre

ambos tipos de Modelado Difuso cuando se afronta la identificación de un sistema complejo para el que se desea obtener un modelo que sea a la vez preciso e interpretable.

Para solucionar este problema, la investigación del área ha derivado en los últimos años hacia dos líneas de trabajo complementarias: i) dotar de mayor precisión al MDL, sin perder la interpretabilidad asociada al mismo en gran medida [CCHM03a], y ii) dotar de mayor interpretabilidad al MDP, sin dañar excesivamente su grado de precisión [CCHM03c]. Esto ha dado lugar a la aparición de dos nuevas áreas en el Modelado Difuso, el MDLE y el MDPR.

Centrándonos en la primera de ellas, que será el objeto de estudio de esta memoria de tesis, se han propuesto distintas posibilidades para mejorar el MDL en la literatura especializada [CCHM03b]. Todos estos enfoques comparten la idea de mejorar la forma en que el modelo difuso lingüístico realiza el razonamiento interpolativo induciendo una mejor cooperación entre las reglas de su base. Esta cooperación entre reglas puede ser promovida a tres niveles, actuando sobre tres componentes distintos del modelo difuso: la BD, la BR y la BC al completo.

En el primer capítulo de esta memoria, comenzaremos analizando ambos tipos de mejoras al Modelado Difuso clásico, MDLE y MDPR, para después estudiar más detenidamente la primera de ellas, el MDLE. Una vez presentados los distintos enfoques existentes en el área, nos centraremos en aquellos que se han considerado para realizar nuestras propuestas, los enfoques de MDLE que actúan sobre la BD y, más concretamente, en los dos pilares básicos de nuestra metodología, los mecanismos de selección de reglas y el ajuste lateral de funciones de pertenencia. En este capítulo, no se ha incluido una descripción detallada de los SBRDs, relegando un estudio más profundo (tipos, composición y funcionamiento) al Apéndice A.

1.1. Modelado de Sistemas Mediante Lógica Difusa: Modelado Difuso Lingüístico versus Modelado Difuso Preciso

Esta sección está dedicada a un doble propósito. Por un lado, a introducir brevemente las características principales del Modelado de Sistemas mediante

SBRDs. Por otro, a presentar los dos tipos principales de modelo incluidos en el área: modelo difuso lingüístico y modelo difuso preciso. Las dos subsecciones siguientes se dedican respectivamente a describir ambas ideas.

1.1.1. Modelado de Sistemas Mediante Sistemas Basados en Reglas Difusas

En la actualidad, 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], la componen los SBRDs. Este tipo de sistemas constituyen 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 por proposiciones difusas en lugar de proposiciones Booleanas clásicas.

En un sentido muy general, un SBRD es un sistema basado en reglas en el que la Lógica Difusa se puede emplear tanto como herramienta para representar distintas formas de conocimiento sobre el problema a resolver, como para modelar las interacciones y relaciones existentes entre las variables del mismo. La principal aplicación de estos sistemas inteligentes es el Modelado de Sistemas [BD95, Ped96, SY93]. Esto se debe a que existe una amplia gama de sistemas reales para los cuales, debido a su nivel de complejidad o imprecisión, no resulta posible obtener buenos resultados con las herramientas clásicas de modelado. Los SBRDs se han aplicado con éxito a una gran cantidad de problemas reales a lo largo de los últimos años [BD95, GT95, Hir93, Ped96, Wan94].

Así, el modelado empleando conceptos propios de la Lógica Difusa es el típico ejemplo de una técnica que hace uso del conocimiento y de los procesos de deducción propios de los seres humanos. En consecuencia, mediante el uso de la Lógica Difusa es posible capturar la vaguedad del modelo y la incertidumbre del problema, de tal forma que el modelo obtenido se encuentre más ajustado a la realidad que otras formas que no puedan capturar estas cuestiones.

Centraremos nuestra atención en la relación entre variables por medio de reglas difusas del tipo SI-ENTONCES:

SI *la potencia es alta* **ENTONCES** *la temperatura subirá rápidamente.*

Estas reglas permiten establecer una relación lógica entre las variables cua-

litativas de un sistema. Dichos valores cualitativos poseen una interpretación lingüística clara, que se denomina usualmente *etiqueta* o *término lingüístico* [Zad75]. El significado de los términos lingüísticos de las variables de entrada y salida está definido por conjuntos difusos, más precisamente por sus funciones de pertenencia. En este sentido, los conjuntos difusos o sus funciones de pertenencia proveen una interfaz entre las variables numéricas de entrada/salida y las correspondientes variables difusas de las reglas. Este concepto de modelado y análisis de sistemas por medio de variables lingüísticas fue introducido por Zadeh [Zad73] y ha sido desarrollado considerablemente en estos últimos años.

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. En este sentido, el solapamiento de las funciones de pertenencia asegura la generalización de situaciones no capturadas completamente por las reglas, lo que da lugar a que la inferencia pueda considerarse como un proceso de interpolación entre los resultados de las reglas individuales.

Estos modelos tienen varias ventajas [Bab98] sobre otros enfoques inteligentes como las Redes Neuronales [MR90] o las Redes de Base Radial:

- El *Modelado Difuso* integra el procesamiento numérico y, el simbólico en un esquema común y, en consecuencia, no se restringe solamente a áreas que requieren especialización o conocimiento humano. En este sentido, los modelos con SBRDs pueden hacer un uso efectivo de algoritmos de aprendizaje basados en ejemplos (datos de entrada-salida) y ser combinados con técnicas convencionales de regresión [TS85, Wan94].
- La *estructura basada en reglas de los Sistemas Difusos* resulta de gran utilidad en el análisis de los modelos desarrollados a partir de datos numéricos, debido a que estas reglas representan una descripción adecuada del sistema que generan los datos. Este tipo de descripción puede ser confrontada y posiblemente combinada con conocimiento experto. Todo esto ayuda a la comprensión del sistema y a la validación del modelo al mismo tiempo.
- El uso de *términos lingüísticos* en las reglas puede verse como una clase de cuantificación, esto es, resulta posible desarrollar modelos de diferentes niveles de abstracción y precisión para un sistema dado, dependiendo del número de valores cualitativos considerados (granularidad).

De este modo, el Modelado de Sistemas mediante SBRDs se puede considerar

como una vía para modelar un sistema haciendo uso de un lenguaje de descripción basado en la Lógica Difusa con predicados difusos [SY93]. 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. Estos dos requisitos, el grado de interpretabilidad del modelo para el ser humano y el grado en que ese modelo aproxima al sistema real, suelen estar enfrentados. La decisión de cuán interpretable y cuán preciso debe ser el modelo suele depender de las necesidades que surjan a partir de un problema concreto.

1.1.2. Modelado Difuso Lingüístico y Modelado Difuso Preciso

Dependiendo del objetivo principal que se desee satisfacer en el proceso de identificación y del tipo de SBRD que se emplee, podemos distinguir entre dos clases de Modelado Difuso:

- *MDL* — Este tipo de modelado se realiza generalmente por medio de SBRD lingüísticos (también conocidos como de tipo Mamdani [Mam74, MA75], véase la Sección A.2 del Apéndice A). En este caso, el principal requisito es la interpretabilidad y el concepto de variable lingüística [Zad75] desempeña un papel fundamental. Además de su gran facilidad para interpretar el comportamiento del sistema, su estructura proporciona un marco natural para incluir conocimiento experto, por lo que son los más empleados en la actualidad.

Los SBRDs lingüísticos están formados por reglas con la siguiente estructura:

SI X_1 es A_1 y \dots y X_n es A_n **ENTONCES** Y es B ,

donde X_i (Y) son las variables lingüísticas de entrada (salida), y A_i y B las etiquetas lingüísticas con los conjuntos difusos μ_{A_i} y μ_B asociados definiendo su significado. Estos términos lingüísticos se toman de una *semántica global* que define la gama posible de conjuntos difusos usados en cada variable.

Uno de los componentes principales de los SBRDs lingüísticos es la BC, que consta de una BR —que contiene las reglas con la estructura simbólica

mencionada— y una BD —que almacena los conjuntos de términos lingüísticos y las funciones de pertenencia que definen sus significados—. En el Apéndice A, se puede consultar una descripción más detallada de este tipo de sistemas.

- *MDP* — En esta clase de modelado se persigue principalmente la precisión de los modelos obtenidos, dejando a un lado su legibilidad. Para ello, podemos emplear los SBRDs aproximativos, que se caracterizan por el uso directo de variables difusas (véase la Sección A.3 del Apéndice A). Así, cada regla difusa presenta su propia semántica, es decir, las variables toman diferentes conjuntos difusos como valores en lugar de términos lingüísticos. Dado que en los SBRDs aproximativos no se emplea una semántica global, los conjuntos difusos no pueden interpretarse con facilidad. Estos modelos pretenden ser más precisos que los anteriores, es decir, capturan la información del problema de un modo más exacto a costa de la consiguiente pérdida de interpretabilidad.

La estructura de regla considerada es la siguiente:

$$\mathbf{SI} X_1 \text{ es } \widehat{A}_1 \text{ y } \dots \text{ y } X_n \text{ es } \widehat{A}_n \mathbf{ENTONCES} Y \text{ es } \widehat{B},$$

donde \widehat{A}_i y \widehat{B} son conjuntos difusos sin una interpretación lingüística directa.

Los SBRDs aproximativos sustituyen la BC por una Base de Reglas Difusas (BRD) con la estructura mencionada, donde cada regla individual contiene la semántica que la describe. En el Apéndice A se describen estos sistemas en profundidad.

Además, existen otros tipos de SBRDs que pueden emplearse también en el MDP de sistemas. Principalmente, podemos destacar los SBRDs de tipo Takagi-Sugeno-Kang [SK88, TS85] (véase la Sección A.4 del Apéndice A). En este caso, la estructura de regla empleada está constituida por variables lingüísticas en el antecedente y una función polinomial de las variables de entrada en el consecuente. Debido a este tipo de consecuente, se pierde interpretabilidad en el modelo, pero a cambio, se permite un mayor grado de aproximación. Por tanto, estos sistemas son adecuados para realizar el MDP.

En la Figura 1.1 se ilustran los distintos tipos de SBRDs comentados en el campo del Modelado de Sistemas. Como podemos observar, dichos SBRDs tienen

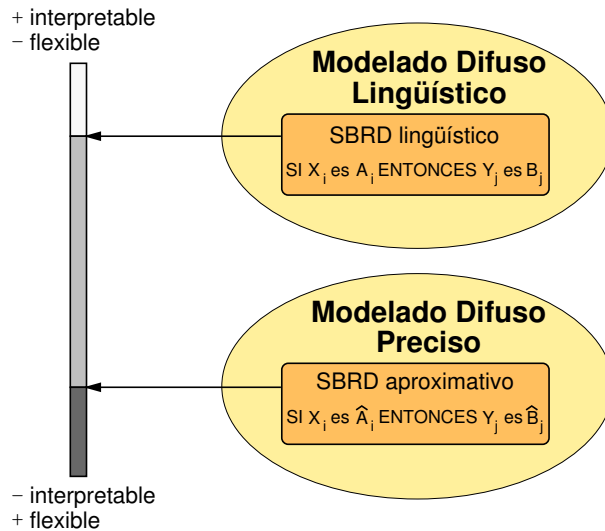


Figura 1.1: Modelado de Sistemas mediante SBRDs: interpretabilidad y flexibilidad dependiendo del tipo de SBRD considerado

diferentes grados de interpretabilidad y flexibilidad según la estructura de regla empleada. El hecho de exigir que el modelo obtenido por un SBRD lingüístico sea fácilmente interpretable implica establecer ciertas restricciones en la estructura de regla, perdiendo así flexibilidad [Bas94]. Por el contrario, relajar tales restricciones, como hacen los SBRDs aproximativos, permite obtener modelos más flexibles aunque provoca una gran pérdida de interpretabilidad.

La *flexibilidad* en el proceso de modelado indica la *capacidad potencial* de aproximarse al problema que se está resolviendo. Desde un punto de vista estadístico, disponer de mayor flexibilidad implica tener un mayor número de grados de libertad en el modelo. Sin embargo, esto no siempre supone obtener resultados precisos ya que esta libertad es un arma de doble filo. Una mayor flexibilidad proporciona capacidad de aproximación al modelo final pero da lugar a que crezcan tanto la complejidad para llevar a cabo el modelado como el riesgo de sobreajuste en la aproximación del sistema real (excesivo grado de aproximación con una mala capacidad de predicción).

Por tanto, en contra de la corriente tradicional dominante en la literatura especializada, el principal aspecto a considerar en el Modelado de Sistemas no debería ser la capacidad del modelo sino la capacidad para generar el modelo.

Así, mayor flexibilidad no significa directamente mayor precisión. Un equilibrio entre la libertad de la estructura del modelo (*flexibilidad del modelo*) y un buen rendimiento del proceso de aprendizaje (*simplicidad del modelado*) podría proporcionar mejor comportamiento, esto es, una mayor precisión (véase la Figura 1.2).

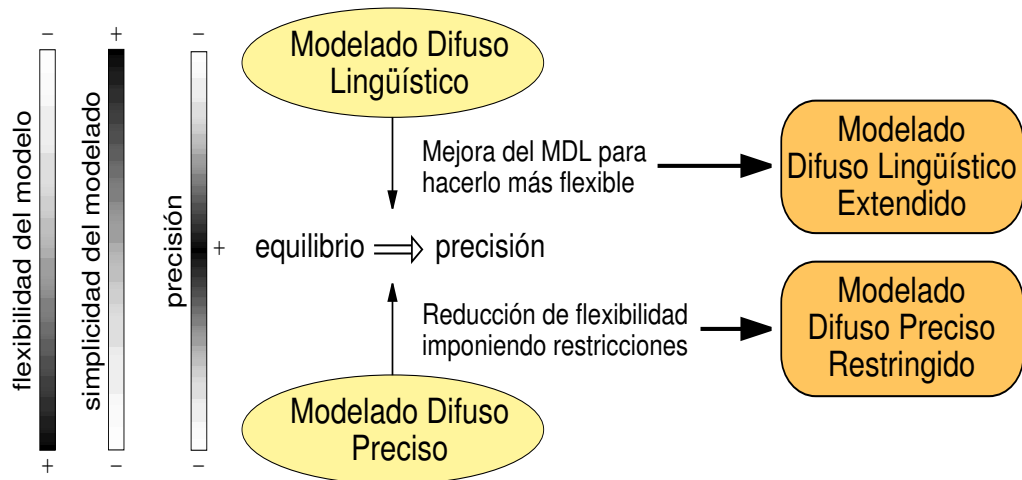


Figura 1.2: Un compromiso entre flexibilidad del modelo y simplicidad del modelado puede proporcionar una mejor precisión. Esto se puede conseguir extendiendo el MDL o restringiendo el MDP

Cuanto más sencillo sea el sistema a modelar, en menor medida será necesario flexibilizar la estructura del modelo para alcanzar un grado de precisión aceptable, de forma que este equilibrio se encontrará más próximo al MDL. Del mismo modo, cuanto más complejo sea el sistema, dicho equilibrio se encontrará más cercano al MDP ya que deberemos sacrificar en mayor medida la interpretabilidad del modelo para alcanzar la flexibilidad suficiente que proporcione un grado de precisión satisfactorio.

De este modo, la investigación del área ha derivado en los últimos años hacia dos líneas de trabajo complementarias para alcanzar este compromiso de flexibilidad y simplicidad:

- i) dotar de mayor precisión al MDL para hacer más flexible el aprendizaje y/o la estructura del modelo considerado, sin perder la interpretabilidad asociada al mismo en gran medida [CCHM03a], y

- ii) dotar de mayor interpretabilidad al MDP, imponiendo restricciones al aprendizaje y/o a la estructura del modelo difuso, sin dañar excesivamente su grado de precisión intrínseco [CCHM03c].

Esto ha dado lugar a la aparición de dos nuevas áreas en el Modelado Difuso, el MDLE y el MDPR. Es importante destacar que, si este equilibrio se alcanza partiendo del MDL, los modelos obtenidos serán más interpretables que partiendo del MDP. Por tanto, parece muy interesante afrontar el Modelado Difuso de Sistemas intentado mejorar, tanto como sea posible, la precisión del MDL sin perder su descripción en exceso. Este será el objetivo principal de la presente memoria.

1.2. Modelado Difuso Lingüístico Extendido: Mejoras a la Precisión

Como hemos comentado con anterioridad, el MDL desarrollado por los SBRDs lingüísticos presenta la ventaja principal de la interpretabilidad asociada al modelo obtenido, que emplea una estructura de regla fácilmente comprensible por el ser humano. Sin embargo, este tipo de modelado presenta un problema fundamental: su falta de precisión al modelar algunos sistemas complejos. Este hecho se debe a cierta inflexibilidad debida al uso de una semántica global que da significado a los conjuntos difusos empleados. En efecto, el uso de variables lingüísticas impone las siguientes restricciones [Bas94, CFM96]:

- Se da una falta de flexibilidad en el SBRD debido a la forma tan rígida en que se particionan los espacios de entrada y salida.
- Cuando las variables de entrada del sistema dependen unas de otras, es muy complicado obtener una partición difusa adecuada de los espacios de entrada.
- La división homogénea de los espacios de entrada y salida es ineficiente y no se adapta adecuadamente cuando se trabaja con espacios hiperdimensionales.

- El tamaño de la BC depende directamente del número de variables y términos lingüísticos que existan en el sistema. La obtención de un SBRD preciso necesita de un aumento significativo de la granularidad, es decir, de la introducción de nuevos términos lingüísticos. Este aumento de la granularidad produce un crecimiento desmesurado del número de reglas, lo que puede provocar que el sistema pierda la capacidad de ser interpretable por el ser humano. Además, en la mayoría de los casos, sería posible obtener un Sistema Difuso equivalente con un número mucho menor de reglas si no se estuviese tan sujeto al particionamiento rígido efectuado sobre el espacio de entrada.

Sin embargo, se pueden realizar distintas mejoras para afrontar estas limitaciones en el marco del MDLE [CCHM03b]. Generalmente, todos estos enfoques comparten la idea de mejorar la forma en que el modelo difuso lingüístico realiza el *razonamiento interpolativo* induciendo una mejor cooperación entre las reglas del modelo aprendido. Esta es una de las características más interesantes de los SBRDs y desempeña un papel clave en su alto rendimiento, siendo una consecuencia de la acción cooperativa entre las distintas reglas lingüísticas que componen el sistema.

Básicamente, se pueden distinguir dos tipos distintos de enfoques según se realicen en:

- El *proceso de derivación del modelo*, extendiendo su diseño a otros componentes distintos de la BR como, por ejemplo, la BD [BKC96, CdJH98, CH97, GP99, Kar91a, LM94, SKG93], o considerando mecanismos más sofisticados para la derivación de BRs con buena cooperación [GP98, CDM99, CCH02].
- La *estructura del modelo*, cambiándola ligeramente para hacerla más flexible y permitir así incrementar la precisión [CH97, CH00, NIT97, PP99].

Las dos subsecciones siguientes están dedicadas a describir brevemente las dos familias mencionadas, repasando algunos de los enfoques propuestos en el marco de cada una de ellas.

1.2.1. Mejoras en el Proceso de Derivación del Modelo Difuso Lingüístico

Tal y como hemos indicado, existen dos variantes incluidas en esta familia de métodos de MDLE. En primer lugar, encontramos aquellos enfoques que extienden el proceso de diseño del modelo difuso lingüístico para incorporar otros componentes como la BD. Por otro lado, existe la posibilidad de mejorar el proceso de derivación de las reglas lingüísticas para conseguir que mejore el nivel de cooperación existente entre las mismas y, consecuentemente, la precisión del modelo final generado. Las dos secciones siguientes describen ejemplos de cada una de las dos variantes.

1.2.1.1. Métodos Basados en la Definición de la Base de Datos

Los métodos básicos de MDL se centran exclusivamente en determinar el conjunto de reglas difusas que forman la BR del modelo [Thr91, WM92]. En estos casos, la BD se obtiene, por lo general, a partir de información de expertos (si está disponible) o mediante un proceso de normalización, y se mantiene fija durante el proceso de derivación de la BR.

Sin embargo, el diseño automático de la BD ha demostrado ser un mecanismo muy apropiado para incrementar la capacidad de aproximación de los modelos difusos lingüísticos. En algunos trabajos, como en [CHV00], se ha probado empíricamente que un buen diseño de la BD puede tener más influencia en el rendimiento del modelo que la derivación de las reglas. Generalmente, el procedimiento consiste en, o bien determinar la forma más adecuada para las funciones de pertenencia que dan significado a los conjuntos difusos asociados a los términos lingüísticos, o bien definir el número de términos lingüísticos empleados en la partición de cada variable, es decir, la granularidad.

A continuación, repasamos los distintos enfoques existentes para el diseño de la BD en función del modo en que se integren en el proceso de derivación y de los efectos que provoquen sobre las funciones de pertenencia. Después, se mencionan algunos aspectos referentes a la interpretabilidad de los modelos obtenidos a partir de los enfoques anteriores.

1.2.1.1.1. Integración del Diseño de la Base de Datos en el Proceso de Derivación Global del Modelo Difuso Lingüístico

El diseño de la BD puede integrarse en el proceso de derivación del SBRD Lingüístico de distintas formas:

- *Diseño preliminar* (aprendizaje de la BD) — Consiste en extraer la BD a priori (es decir, antes de derivar las reglas del modelo difuso lingüístico) por inducción a partir del conjunto de ejemplos disponible. Tradicionalmente, este proceso se ha realizado con técnicas de agrupamiento no supervisado [LMV99, Ped01].
- *Diseño a priori* (aprendizaje de la BC) — Recientemente, ha surgido un nuevo enfoque que realiza el proceso de derivación del modelo difuso lingüístico considerando un método básico de derivación de reglas incrustado [CHV00, CHV01a, CHMV01, Glo96, IM96, PRO⁺00, RPOP00]. Esta técnica consiste en disponer de un método de aprendizaje sencillo que diseña, a partir de una BD determinada, otros componentes distintos de ésta (p.ej., la BR). Mediante un proceso de meta-aprendizaje, el método genera distintas BDs y valida su eficacia ejecutando el método de aprendizaje de reglas básico.
- *Diseño simultáneo* (aprendizaje de la BD junto con otros componentes) — El proceso de diseño de la BD se realiza conjuntamente con otros componentes tales como la BR en un mecanismo simultáneo [GJ99, HM95, JvSS99, KS95, MMH97, WHT98, XL00].
- *Diseño a posteriori* (ajuste de la BD) — Este enfoque, conocido como *ajuste*, consiste en refinar la BD a partir de una definición previa una vez que el resto de componentes ha sido obtenido. Se trata de uno de los procedimientos más extendidos. Habitualmente, el ajuste consiste en variar la forma de las funciones de pertenencia [BKC96, CdJH98, CH97, GP99, HLV95, Kar91a, LCT01, LM94, NK99, SKG93] y persigue como principal requisito mejorar la precisión del modelo lingüístico. No obstante, en algunos casos se realiza otro tipo de refinamiento para mejorar la interpretabilidad como, por ejemplo, la fusión de funciones de pertenencia similares [EV00].

Naturalmente, varios de estos enfoques pueden ser considerados conjuntamente. Por ejemplo, en [LM94] se realiza un diseño de la BD en dos etapas, derivando en primer lugar la BD y la BR mediante un proceso conjunto, y aplicando después un ajuste. En [Jin00], se efectúa primero una generación inicial de la BR para, a continuación, realizar un diseño de la BD en tres etapas (selección de las variables de entrada, ajuste de la BD y reducción de la BR simultáneos, y ajuste fino de la BD).

Los enfoques de diseño preliminar, a priori y a posteriori suelen combinarse con otros métodos para realizar el proceso completo de derivación en varias etapas sucesivas. En cambio, el mecanismo simultáneo de diseño de la BD junto con otros componentes constituye un único proceso de derivación en sí mismo.

La derivación secuencial tiene la ventaja de reducir el espacio de búsqueda, puesto que se trabaja con espacios confinados en cada etapa. Por otro lado, en la derivación simultánea se considera de mejor forma la fuerte dependencia existente entre los distintos componentes. Sin embargo, el proceso se vuelve mucho más complejo en este último caso debido a que el espacio de búsqueda crece significativamente, haciendo fundamental la elección de una técnica de búsqueda apropiada.

Por último, comentaremos que el diseño de la BD dota al proceso de modelado de mayor flexibilidad pero introduce el riesgo de perder interpretabilidad y sobreajustar el problema, por lo que su aplicación ha de ser realizada con cuidado. Describiremos algunos mecanismos para preservar una buena interpretabilidad más adelante, en esta misma sección.

1.2.1.1.2. Aprendizaje/Ajuste de las Funciones de Pertenencia

Podemos distinguir distintos enfoques según las operaciones efectuadas sobre las funciones de pertenencia en el proceso de definición de la BD:

- *Aprendizaje/ajuste de los parámetros de las funciones de pertenencia* — La forma más común de derivar las funciones de pertenencia es alterar los valores de los distintos parámetros que las definen para realizar desplazamientos y/o ensanchamientos de los conjuntos difusos. Esto se puede conseguir ajustando directamente cada uno de los parámetros o usando diferentes factores de escala lineales [BKC96, CH97, GJ99, HLV95, HM95, Jin00, JvSS99, Kar91a, KS95, LM94, NK99, PS01, RPOP00, SKG93, WHT98]. Por ejem-

plo, si consideramos la siguiente función de pertenencia triangular:

$$\mu(x) = \begin{cases} \frac{x-a}{b-a}, & \text{si } a \leq x < b \\ \frac{c-x}{c-b}, & \text{si } b \leq x \leq c \\ 0, & \text{en otro caso} \end{cases}$$

alterar los parámetros a , b y c supone variar la forma del conjunto difuso asociado a la función de pertenencia (véase la Figura 1.3), afectando así al comportamiento del SBRD. Lo mismo ocurre en el caso de los demás tipos de funciones de pertenencia (trapezoidales, gaussianas, sigmoidales, etc.).

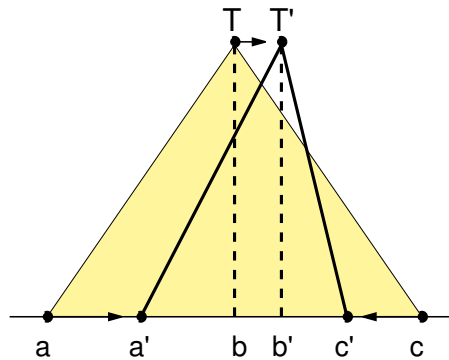


Figura 1.3: Definición de funciones de pertenencia alterando sus parámetros básicos

- *Uso de modificadores lingüísticos* — Otra filosofía de definición de las funciones de pertenencia consiste en el uso de expresiones más flexibles que permitan variar el grado de compatibilidad de los conjuntos difusos de forma no lineal. Para ello, se emplean habitualmente modificadores lingüísticos [CdJH98, GP99, LCT01] (analizaremos más detenidamente estos operadores en la Sección 1.2.2.1). Por ejemplo, se puede obtener una nueva función de pertenencia elevando el valor de pertenencia a la potencia de un parámetro real positivo α , es decir,

$$\mu'(x) = \mu(x)^\alpha, \quad 0 < \alpha.$$

En este caso, el proceso de ajuste consistiría en obtener el valor más adecuado del parámetro α para mejorar el rendimiento del SBRD. Cambiando

el valor de α , es posible definir distintos tipos de funciones de pertenencia. La Figura 1.4 muestra gráficamente el efecto de este enfoque.

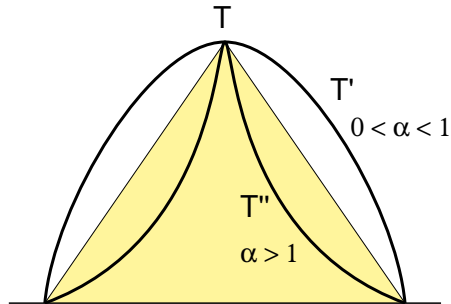


Figura 1.4: Definición de funciones de pertenencia mediante modificadores lingüísticos

- *Manejo del contexto* — Una tercera posibilidad para cambiar las funciones de pertenencia consiste en definir el contexto, es decir, su rango de operación. Se suele poner en práctica mediante *funciones de escala*, que establecen una aplicación entre las variables de entrada-salida y el universo de discurso sobre el que se definen los conjuntos difusos. Desde un punto de vista lingüístico, la función de escala puede ser interpretada como un tipo de información de contexto.

Es posible distinguir entre funciones de escala lineales y no lineales, las cuales son bien conocidas en la Teoría de Control clásica:

- *Contexto lineal* [GG94] — Una función de escala lineal presenta la forma:

$$f(x) = \alpha \cdot x + \beta.$$

El factor de escala α aumenta o reduce el rango de operación, lo que implica la reducción o ampliación de la sensibilidad del controlador/modelo con respecto a la variable de entrada en cuestión, o la correspondiente ganancia en el caso de las variables de salida. El parámetro β desplaza el rango de operación y representa el papel de un desplazamiento de la variable.

- *Contexto no lineal* [GGP98, Kla94, MMH97, PGG97] — La principal desventaja del escalado lineal es la invariabilidad de la distribución relativa de las funciones de pertenencia. El escalado no lineal soluciona el problema al modificar la distribución relativa y cambiar la forma de las funciones de pertenencia con transformaciones no lineales. Una función no lineal de escala habitual para una variable, con simetría con respecto al origen, presenta la siguiente forma:

$$f(x) = \text{signo}(x) \cdot |x|^\alpha.$$

El escalado no lineal aumenta ($\alpha > 1$) o disminuye ($\alpha < 1$) la sensibilidad relativa en la región que rodea al origen y consigue el efecto contrario en los extremos del rango de operación. La Figura 1.5 representa algunos de los efectos provocados por este tipo de escalado.

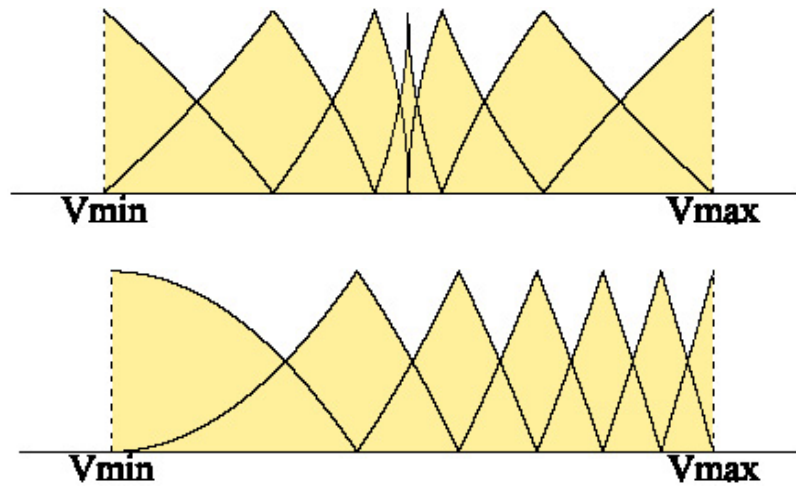


Figura 1.5: Efectos provocados por un escalado no lineal sobre una partición difusa con funciones de pertenencia triangulares

De este modo, se puede definir la BD aplicando factores de escala no lineales en toda la partición difusa [PGG97]. En algunas ocasiones, incluso se emplea un único factor de escala no lineal para toda la BD [NIT97].

Por otro lado, existen métodos de definición de la BD que combinan las dos filosofías de adaptación de contextos, escalado lineal y no lineal [CHMV01].

Finalmente, existen otras contribuciones que proponen métodos para definir no solo la forma de las funciones de pertenencia, sino también el tipo de éstas (triangular, trapezoidal, gaussiana o sigmoïdal) [SEC99].

1.2.1.1.3. ¿Cómo Mantener la Interpretabilidad del Modelo Difuso Lingüístico al Diseñar la Base de Datos?

Ya hemos comentado el hecho de que el diseño de la BD permite mejorar en gran medida la precisión de los modelos difusos lingüísticos. Sin embargo, es importante señalar también que este proceso de diseño puede provocar que las particiones difusas en la BD tomen una forma compleja, lo que puede perjudicar la interpretación del sistema por parte de un experto. La Figura 1.6 recoge un ejemplo en el que se muestran varias funciones de pertenencia con formas complicadas que causan una pérdida de interpretabilidad en el modelo.

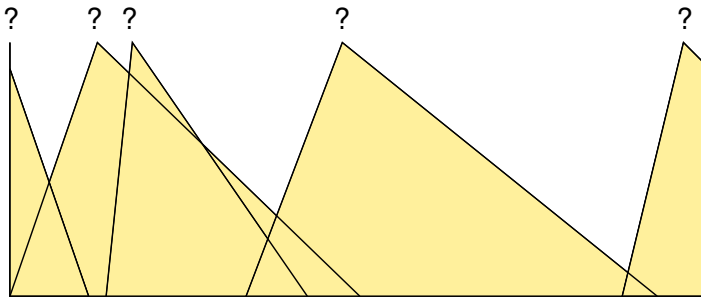


Figura 1.6: Pérdida de interpretabilidad al diseñar la BD

Decidir si una BD concreta es interpretable o no es una tarea difícil y subjetiva. Aún así, algunas investigaciones han estudiado este aspecto proponiendo varias propiedades para asegurar una buena interpretabilidad durante el proceso de optimización de las funciones de pertenencia [CCHM03c, Val99a, Val99b]. Es posible imponer una serie de restricciones de diseño con objeto de obtener una BD que mantenga la comprensibilidad del modelo en la mayor medida posible [BB03, CH97, CL00, EV00]. A continuación resumimos algunas de las propiedades más importantes que han de satisfacer las BDs optimizadas, junto con algunas soluciones posibles para el diseño de la BD:

- *Propiedad de cobertura* — Cada valor del universo de discurso ha de

pertenecer a, al menos, un conjunto difuso. Alternativamente, se puede considerar un criterio más estricto exigiendo un grado mínimo de cobertura para todos los elementos del dominio.

Soluciones posibles — Esta propiedad puede satisfacerse empleando intervalos de variación que aseguren la intersección (“*overlapping*”) entre dos conjuntos difusos consecutivos [CH97] o exigiendo que las particiones difusas sean “fuertes” tras el proceso de optimización [CL00] (es decir, que todo elemento del dominio esté cubierto siempre a grado 1).

- *Propiedad de normalidad* — Cada función de pertenencia debe tener un grado de pertenencia máximo con, al menos, un elemento del universo de discurso. Es decir, los conjuntos difusos deben ser normales.

Soluciones posibles — Puede conseguirse de un modo muy sencillo obligando a que los puntos modales de las funciones de pertenencia de los conjuntos difusos de los extremos de la partición estén contenidos dentro del dominio de la variable.

- *Propiedad de distinguibilidad* — Cada término lingüístico debe tener un significado claro y su conjunto difuso asociado debe especificar claramente un intervalo del universo de discurso. En resumen, las funciones de pertenencia de la partición difusa deben ser lo suficientemente distintas entre sí.

Soluciones posibles — Puede ponerse en práctica restringiendo la variación de los parámetros de las funciones de pertenencia de una forma adecuada [LAT96], combinando funciones de pertenencia similares [EV00] o estableciendo una semántica sobre los términos lingüísticos [Bod00, BB00].

Naturalmente, las restricciones necesarias para asegurar la integridad semántica de la BD hacen menos flexible el proceso de derivación pero, además de mantener la interpretabilidad del modelo, reducen el riesgo de sobreajustar el sistema real.

Cuando se consideran técnicas de optimización y búsqueda para el diseño de la BD, existe otra posibilidad para asegurar las propiedades de integridad consistente en incluir medidas de interpretabilidad en la función objetivo que guía la búsqueda hacia buenas soluciones. Habitualmente se consideran medidas de completitud y consistencia [CH97, GP97, JvSS99], compacidad o similitud [JGSRB01]. En [SKFT01], se propone un criterio más elaborado, denominado concisión, obtenido a partir de la combinación de una medida de entropía difusa, que distingue las formas de las funciones de pertenencia, y una medida de

desviación, que evalúa la discrepancia de una función de pertenencia con respecto a su correspondiente simétrica.

En procesos de diseño a priori de la BD, otro enfoque interesante es considerar el efecto que se produce en el tamaño (la compacidad) de la BR al definir el número de términos lingüísticos y sus funciones de pertenencia asociadas. En [PRO⁺00, RPOP00], los autores proponen usar un Sistema Difuso simple para seleccionar la solución con el balance deseado entre interpretabilidad/compacidad (número de reglas) y precisión (error en la aproximación) entre las distintas BCs generadas.

1.2.1.2. Métodos de Generación de Reglas Orientados a Mejorar la Cooperación

En esta categoría consideraremos aquellos métodos de derivación que, centrándose exclusivamente en el aprendizaje de reglas, consideran un proceso sofisticado con objeto de mejorar la cooperación de la BR generada y, consecuentemente, la precisión del modelo. En este caso, la BD y la estructura del modelo permanecen inalteradas, lo que resulta en la máxima interpretabilidad posible. A continuación describiremos brevemente algunos de los enfoques existentes.

En [GP98], se propone un método para refinar una BR previamente generada compuesta por reglas en forma normal disyuntiva (o reglas DNF – *Disjunctive Normal Form* –) (véase la Sección A.2 del Apéndice A). El proceso heurístico está compuesto por tres pasos consecutivos. En primer lugar, se mejora la precisión del modelo eliminando algunos términos lingüísticos de los antecedentes de las reglas. Entonces, se mejora la capacidad de generalización añadiendo términos lingüísticos a los consecuentes. Por último, se considera la completitud de la BR añadiendo nuevas reglas.

Por otro lado, en [CCH02], se propone la *metodología COR* (cooperación entre reglas), basada en el hecho de que la derivación de la BR se efectúa haciendo uso de un criterio global de rendimiento que considera la acción de las distintas reglas de la base conjuntamente. Este mecanismo de derivación de reglas se aplica sobre un espacio de búsqueda reducido, en el que solamente se consideran un subconjunto concreto de las reglas posibles, seleccionadas mediante la información heurística disponible. Así, las ventajas principales de la metodología COR son su capacidad para incluir información heurística, su flexibilidad para ser aplicada con distintas metaheurísticas y su facilidad de integración con distintos procesos

básicos de derivación de reglas lingüísticas.

Finalmente, los procesos de simplificación de BRs son técnicas de diseño de modelos difusos lingüísticos que se basan en la alteración de una BR previamente generada haciendo uso de conocimiento experto o mediante un método de aprendizaje automático para reducir su tamaño.

En principio, la reducción de la complejidad del modelo es una forma de mejorar su interpretabilidad y no su precisión. Es claro el hecho de que un modelo con un número bajo de reglas en su base es más sencillo de interpretar. Sin embargo, y especialmente en problemas multidimensionales complejos con relaciones de entrada-salida fuertemente no lineales, la BR de los modelos difusos suele presentar un número excesivo de reglas e incluso reglas que presentan mala cooperación que, aparte de ocasionar una disminución de su comprensibilidad, provocan que el rendimiento del modelo difuso lingüístico empeore. En estos casos, la simplificación de reglas ayuda por tanto a mejorar la precisión del modelo.

Podemos dividir los métodos de simplificación de reglas en dos grupos distintos:

1. *Métodos de reducción de reglas*, basados en agregar varias reglas entre sí. En [KNN98], los autores proponen agregar aquellas reglas que sean vecinas, es decir, las reglas difusas donde la etiquetas de cada una de las variables de las reglas son adyacentes. En [HL99], se hace una consideración especial al orden en que se hace la unión. Por otro lado, en los SBRDs de tipo Takagi-Sugeno, también han sido propuestos métodos para simplificar modelos difusos mediante la reducción de reglas [RS01, SBKN98, SH00].
2. *Métodos de selección de reglas*, que se basan en seleccionar un subconjunto de reglas con el mejor nivel de cooperación posible tomando como punto de partida el conjunto completo de reglas. En la literatura podemos encontrar métodos con diferentes algoritmos de búsqueda para encontrar la combinación de reglas más satisfactoria [CH00, HLV98a, INYT95, IMT97, IY03, KKS00].

En [KT01], se propone un proceso heurístico interesante para la selección de reglas donde, por medio de medidas estadísticas, se calcula un factor de relevancia para cada regla difusa que compone el SBRDs lingüísticas para después seleccionar las más relevantes. La filosofía de ordenar las reglas en función de un criterio de importancia y seleccionar un subconjunto de las mejores parece similar a los métodos de transformación ortogonal

considerados por los SBRDs de tipo Takagi-Sugeno [YBY99, YW99].

Para ambos, se han empleado distintas técnicas tales como Redes Neuronales, algoritmos de Clustering, métodos de transformación ortogonal y algoritmos basados en medidas de similitud, entre otros. Por otro lado, los Algoritmos Genéticos (AGs) se han considerado en un gran número de contribuciones para poner en práctica esta tarea, especialmente en la vertiente de selección de reglas. Analizaremos más detenidamente los métodos genéticos de selección de reglas en la Sección 1.3, puesto que serán considerados a lo largo de esta memoria.

1.2.2. Extensiones en la Estructura del Modelo Difuso Lingüístico

Otra forma de mejorar el rendimiento del MDL es extender la estructura usualmente considerada para el modelo difuso lingüístico haciéndola más flexible. En las siguientes subsecciones se muestran algunos ejemplos de flexibilización en esta línea.

1.2.2.1. Uso de Modificadores Lingüísticos

Una posibilidad para relajar la estructura del modelo consiste en incluir ciertos operadores que alteren ligeramente el significado de las etiquetas lingüísticas implicadas en cada regla difusa del sistema. Como Zadeh sugiere en [Zad75], esto se puede realizar manteniendo un grado de descripción adecuado mediante el uso de *modificadores lingüísticos*.

Un modificador lingüístico [BD95, CYP96, MP96] es un operador que altera la función de pertenencia del conjunto difuso asociado a la etiqueta lingüística, obteniendo una definición alternativa con un grado de precisión superior o inferior a la original dependiendo del caso. Dos de los modificadores lingüísticos más conocidos son el de *concentración* ‘muy’ y el de *dilatación* ‘más-o-menos’. Sus expresiones son las siguientes:

$$\mu_T^{\text{muy}}(x) = (\mu_T(x))^2, \quad \mu_T^{\text{más-o-menos}}(x) = \sqrt{\mu_T(x)}$$

y sus efectos al aplicarlos sobre una función de pertenencia triangular se muestran en la Figura 1.7.

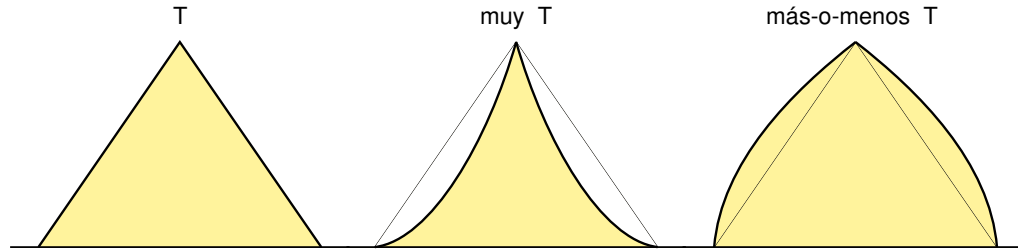


Figura 1.7: Efectos de los modificadores lingüísticos ‘muy’ y ‘más-o-menos’

Así, tendríamos una nueva estructura de regla lingüística que podríamos definir de la siguiente forma:

SI X_1 es $ml_{X_1} A_1$ y ... y X_n es $ml_{X_n} A_n$ **ENTONCES** Y es $ml_Y B$,

donde ml_{X_i} (ml_Y) indica el modificador lingüístico a emplear (incluido el operador identidad) en la correspondiente variable de forma que el grado de pertenencia al término lingüístico viene dado por $\mu_{A_i}^{ml_{X_i}}$ ($\mu_B^{ml_Y}$). Un posible ejemplo de regla que sigue esta estructura es el siguiente:

SI X_1 es *muy* alto y X_2 es bajo **ENTONCES** Y es *más-o-menos* grande.

1.2.2.2. Uso de más de un Consecuente para cada Regla

Este enfoque consiste en permitir que la BR contenga reglas en las que cada combinación de antecedentes tenga varios consecuentes asociados si con ello se mejora la precisión del modelo [CH97, CH00, NIT97]. La estructura de regla sería la siguiente:

SI X_1 es A_1 y ... y X_n es A_n **ENTONCES** Y es $\{B_1, \dots, B_c\}$

donde c es el número de consecuentes distintos considerados en la regla. Un caso particular de este enfoque son las reglas de doble consecuente (es decir, $c = 2$), propuestas en [CH00, NIT97].

El uso de varios consecuentes no influye en el sistema de inferencia del modelo

difuso lingüístico. La única restricción que debemos imponer es que el método de defuzzificación considere el grado de emparejamiento de las reglas disparadas. Por ejemplo, se puede usar la estrategia de defuzzificación del *centro de gravedad ponderado por el grado de emparejamiento* [CHP97] (véase el Apéndice A).

1.2.2.3. Uso de Reglas Ponderadas

El uso de reglas ponderadas consiste en emplear uno o varios parámetros adicionales en cada regla con objeto de dotar al sistema de más grados de libertad para mejorar el razonamiento interpolativo que desarrolla y, por lo tanto, el rendimiento del modelo.

Existen distintas posibilidades dependiendo de que se asocie un peso a cada variable de entrada, con objeto de definir la importancia relativa de cada una de ellas, o un único peso para toda la regla, resultando en una regla difusa con la siguiente estructura:

SI X_1 es A_1 y ... y X_n es A_n **ENTONCES** Y es B con $[W]$

El operador *con*, que liga el peso a una regla, se puede definir de diferentes formas en función de que el peso se aplique a nivel de antecedente (lo más habitual es multiplicar el grado de emparejamiento del antecedente por el correspondiente peso antes de aplicar el operador de implicación), a nivel de consecuente (cambiando la conclusión derivada por el operador de implicación atendiendo al correspondiente peso; por ejemplo, alterando el soporte del conjunto difuso obtenido), o a nivel de regla completa (modalidad en la que el peso indica el grado de importancia de la regla en cuestión en el proceso de inferencia, en lugar de considerar todas las reglas con la misma importancia como en el caso usual).

1.2.2.4. Uso de las 2-Tuplas Lingüísticas

Recientemente se presentó en [AH04] un nuevo modelo de representación de reglas difusas basado en el concepto de translación simbólica [HM00], el cual nos permite realizar un desplazamiento lateral de las etiquetas.

La translación simbólica de un término lingüístico es un número (α) dentro del intervalo $[-0.5, 0.5)$, determinando este intervalo el dominio de la etiqueta cuando

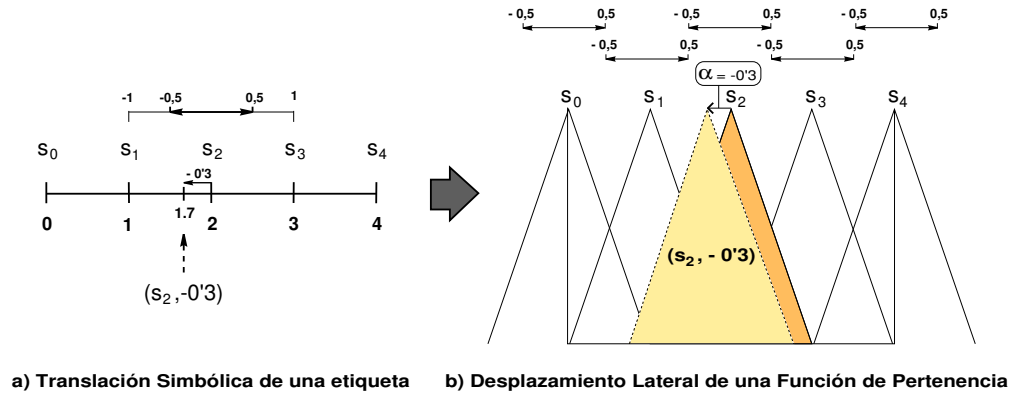


Figura 1.8: Traducción simbólica de una etiqueta lingüística

esta es movida entre sus dos etiquetas laterales adyacentes (ver la Figura 1.8)

Existen dos posibilidades dependiendo de que el desplazamiento de las etiquetas se asocie a nivel de las particiones lingüísticas, con el objetivo de que el desplazamiento de una etiqueta sea el mismo en todas las reglas en las que se encuentre, o a nivel de regla, donde el desplazamiento de una etiqueta es diferente para cada regla en la que aparece. La estructura de regla sería la siguiente:

SI X_1 es (A_1, α_1) y ... y X_n es (A_n, α_n) **ENTONCES** Y es (B, α_Y)

donde α_i (α_Y) indica el desplazamiento lateral asociado a cada etiqueta. En la Sección 1.4 analizaremos con más detalle el uso de las 2-tuplas lingüísticas, puesto que será considerada a lo largo de esta memoria.

1.2.3. Interpretabilidad del Modelado Difuso Lingüístico Extendido

En esta sección vamos a estudiar los efectos que las extensiones analizadas en las secciones anteriores producen en el grado de interpretabilidad de los modelos difusos lingüísticos obtenidos. La Figura 1.9 muestra la pérdida de interpretabilidad derivada de cada extensión considerada. Debemos comentar que el orden mostrado ha de considerarse en términos generales, ya que una acción ligera teóricamente menos interpretable podría ser mejor que otra acción más brusca pero

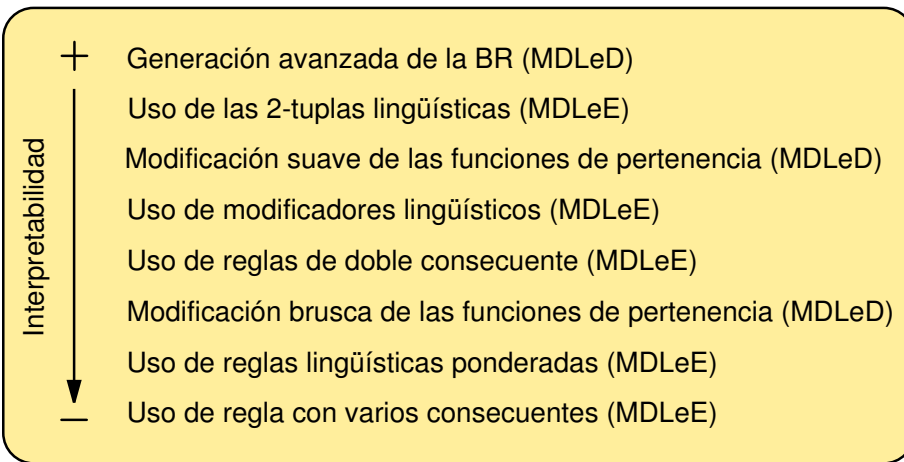


Figura 1.9: Pérdida de interpretabilidad dependiendo de la extensión realizada en el MDL —MDLeD indica MDL extendido en el proceso de derivación y MDLeE indica MDL extendido en la estructura del modelo—

teóricamente más interpretable. Naturalmente, este orden es ambiguo y el único propósito de la Figura 1.9 es formar al lector una idea de la consecuencia en la pérdida de interpretabilidad sufrida al aplicar las diferentes extensiones al MDL.

- *Generación avanzada de la BR* — Obviamente, este tipo de extensiones no suponen ninguna pérdida de interpretabilidad sobre el MDL clásico (a menos que se obtenga un número de reglas muy elevado), ya que se centran bien en una derivación más sofisticada de las reglas difusas lingüísticas, bien en aplicar una simplificación de la BR, lo que en general no altera ni la BD ni la estructura del modelo.

La única alteración en la interpretabilidad puede producirse en aquellos métodos de reducción de reglas basados en transformar reglas clásicas de tipo Mamdani en reglas DNF (véase la Sección A.2 del Apéndice A), como el propuesto en [CA98], puesto que en ese caso sí que se produce un cambio en la estructura del modelo. Sin embargo, el nuevo tipo de regla considerado no dificulta su legibilidad, ya que esta estructura de regla es fácilmente comprensible por el ser humano.

En la Sección 1.3 veremos en detalle la selección de reglas, que permite la obtención de modelos más simples y por lo tanto mejora la interpretabilidad.

- *Uso de las 2-tuplas lingüísticas* — El uso de las 2-tuplas lingüísticas nos permite mantener la forma original de las funciones de pertenencia (simétricas), modificando lateralmente la localización del soporte. Además, los desplazamientos laterales de las etiquetas involucradas están restringidos a un pequeño intervalo, asegurando el solapamiento entre dos etiquetas adyacentes y previniendo el cruce entre sus puntos vértices. Todo esto hace que se mantenga la interpretabilidad global del modelo en un alto grado, aunque la pérdida de la interpretabilidad será mayor cuanto más grandes sean los desplazamientos de las etiquetas.

En la Sección 1.4 veremos en detalle el ajuste lateral de funciones de pertenencia [AH04], donde se realiza una modificación suave de las funciones de pertenencia mediante el uso de las 2-tuplas lingüísticas.

- *Modificación suave de las funciones de pertenencia* — Si se realiza una modificación suave de las funciones de pertenencia (considerando como referencia la partición difusa inicial o una uniforme, si no disponemos de ella), se puede conseguir mejorar el rendimiento del modelo manteniendo un grado de interpretabilidad similar al original.

Precisar cuán suave es la modificación de las funciones de pertenencia es difícil de conseguir. Para ello, podemos atender a diferentes propiedades que garantizan la interpretabilidad de las particiones difusas. Por ejemplo, tendríamos la *completitud* (para asegurar que ningún punto del dominio queda fuera de la partición difusa), la *ordenación semántica* (para mantener el orden de los términos lingüísticos de la partición) y la *dominancia* (para impedir que los puntos del dominio tengan un alto grado de pertenencia a más de dos términos lingüísticos).

Para conseguir que el efecto de la definición de la BD mantenga un alto grado de interpretabilidad se pueden imponer ciertas restricciones [Val99a] tales como usar particiones difusas *fuertes* (es decir, aquellas en las que la suma de los grados de pertenencia de cualquier valor del dominio a todas las etiquetas es 1) [BKC96], considerar intervalos de variación para cada conjunto difuso [CH97], forzar a mantener un orden establecido entre los vértices [CH97, LM94] o usar funciones de pertenencia simétricas [Kar91a, LM94]. Estas restricciones hacen menos flexible el proceso de derivación pero, además de facilitar la legibilidad, reducen el riesgo de sobreajustar el problema (véase la Sección 1.2.1.1.3).

- *Uso de modificadores lingüísticos* — El empleo de modificadores lingüísticos

mantiene un alto grado de legibilidad ya que el efecto de su uso tiene un significado lingüístico directo. La pérdida de interpretabilidad será tanto mayor cuanto mayor sea la gama de modificadores considerados y mayor sea la frecuencia de su uso en las diferentes reglas lingüísticas.

- *Uso de reglas con doble consecuente* — Al usar reglas con dos consecuentes, la interpretación de la acción realizada por cada regla puede confundirse en cierto grado. Sin embargo, debemos comentar que este hecho no constituye una inconsistencia desde el punto de vista del MDL sino simplemente un desplazamiento de las etiquetas implicadas haciendo que la salida final de la regla caiga en una zona intermedia entre ambos consecuentes.

En efecto, supongamos que una combinación específica de antecedentes, « X_1 es A_1 y ... y X_n es A_n », tiene dos consecuentes diferentes asociados, B_1 y B_2 . La regla de doble consecuente resultante podría interpretarse de la siguiente forma [CH00]:

SI X_1 es A_1 y ... y X_n es A_n **ENTONCES** Y está entre B_1 y B_2 .

- *Modificación brusca de las funciones de pertenencia* — Si no se consideran los criterios mencionados con anterioridad, esta tarea puede generar una BD intrincada que podría afectar significativamente a la interpretación del modelo, perdiendo así legibilidad. La Figura 1.10 muestra un ejemplo donde una alteración excesiva de los parámetros de las funciones de pertenencia puede implicar una pérdida de interpretabilidad.

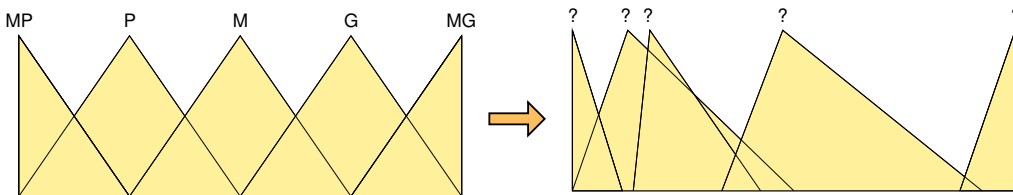


Figura 1.10: Pérdida de interpretabilidad al hacer una modificación brusca de las funciones de pertenencia

- *Uso de reglas lingüísticas ponderadas* — La pérdida de interpretabilidad al usar reglas ponderadas reside principalmente en la dificultad de interpretar la acción real realizada por cada regla en el proceso de razonamiento interpolativo. Además, aplicar un peso a una regla implica, en definitiva, cambiar

las funciones de pertenencia contenidas en la misma regla, lo que supone considerar términos lingüísticos con un significado diferente para cada regla, afectando así seriamente a la interpretabilidad del sistema [NK98, Nau00].

- *Uso de reglas con varios consecuentes* — Por último, las consideraciones de interpretabilidad en este caso son similares a las mencionadas en las reglas con doble consecuente. Sin embargo, usar varios consecuentes (más de dos por regla) hace aún más difícil interpretar el comportamiento del sistema, sobre todo cuando se llega a una situación donde se consideran muchos consecuentes en cada regla o incluso se repiten algunos de ellos. El uso de un número indiscriminado de reglas sin considerar criterios de consistencia sólo puede justificarse desde el punto de vista de la precisión del modelo obtenido.

1.3. Selección de Reglas Lingüísticas con Algoritmos Genéticos: Estudio, Análisis y Ejemplos

En algunas ocasiones es necesario utilizar un conjunto de reglas con elevado número de reglas para obtener un grado de precisión aceptable. Sin embargo, un número excesivo de reglas puede dificultar la comprensión del modelo obtenido. Además, en un conjunto de reglas de gran tamaño podríamos encontrar diferentes tipos de reglas: *reglas irrelevantes*, que no contienen información importante; *reglas redundantes*, cuyas acciones están cubiertas por otras reglas; *reglas erróneas*, que están mal definidas y distorsionan el funcionamiento del SBRD; y *reglas conflictivas*, que perturban el funcionamiento del SBRD cuando coexisten con otras reglas. Este tipo de reglas provoca que tanto el rendimiento del modelo difuso como su interpretabilidad disminuyan. Este tipo de reglas se obtienen normalmente en las siguientes situaciones:

- Cuando la BR se obtiene únicamente a partir de conocimiento experto, presentando normalmente reglas redundantes, conflictivas y erróneas.
- Cuando consideramos un proceso de aprendizaje de reglas difusas con tendencia a generar muchas reglas, pudiendo encontrarse reglas redundantes y conflictivas en la BR. Por ejemplo, métodos como el algoritmo de

Wang y Mendel [WM92] y la estrategia orientada por Espacios de Entrada [NIT97] están guiados por criterios de envoltura que aseguran un nivel de cubrimiento que en algunas ocasiones no es necesario. El Método Mixto (MM) [CCZ04] consiste en añadir reglas al modelo lingüístico generado por el algoritmo de Wang y Mendel en los subespacios difusos de entrada en los que haya ejemplos y para los que no se ha generado ninguna regla, permitiendo mejorar la precisión del modelo lingüístico. Sin embargo, este tipo de métodos son sólo útiles para obtener un conjunto de reglas candidatas a partir de las cuales seleccionar, en una segunda fase, el subconjunto de reglas que mejor cooperan [CH00, IMT97]. En estos casos, hay dos situaciones que favorecen la generación de reglas no deseadas:

- En problemas multidimensionales complejos con relaciones de entrada-salida fuertemente no lineales, donde la BR del modelo suele presentar muchas reglas y es más difícil conseguir la cooperación entre ellas.
- En problemas con un número elevado de dimensiones el número de reglas crece exponencialmente a medida que se añaden nuevas entradas. En estos casos, un conjunto de reglas de gran tamaño podría contener muchas reglas redundantes o incluso reglas conflictivas.

Para resolver este problema, los métodos de reducción de reglas agregan reglas repetidas y/o seleccionan un subconjunto de reglas a partir de un conjunto de reglas difusas dado, con el fin de reducir la complejidad del modelo a la vez que se mantiene (o incluso se mejora) la precisión del SBRD [IY03, KKS00]. Así, las reglas inconsistentes y erróneas que degradan el rendimiento global del sistema son eliminadas, obteniendo un conjunto de reglas difusas que presenta una mayor cooperación y, por lo tanto, una mejora potencial en la precisión del sistema. Además, como se ha dicho en la sección 1.1, en la mayoría de los casos la precisión no es el único requisito impuesto al modelo, llegando a ser la interpretabilidad un aspecto de gran importancia. Reducir la complejidad del modelo lingüístico finalmente obtenido es una forma de mejorar la legibilidad del sistema, es decir, un sistema compacto con pocas reglas requiere un menor esfuerzo para ser interpretado.

Los métodos de reducción de reglas se han formulado usando Redes Neuronales, técnicas de agrupamiento (*clustering*) junto con métodos de transformación ortogonal, y algoritmos basados en medidas de similitud, entre otros [Chi94, HG94, RGB93, SBKN98, SH00, YBY99, YW99]. Combs y Andrews pro-

ponen en [CA98], un enfoque diferente que intenta reducir el crecimiento de la BR transformando reglas difusas elementales en reglas DNF.

Los AGs también han sido aplicados con éxito a la reducción de reglas [CH97, GJ99, INYT95, IMT97, RS00]. Con este tipo de técnicas de búsqueda se puede representar cualquier tipo de regla difusa e incluir conocimiento experto con mucha facilidad [CHHM01]. En esta sección, se realiza un estudio de diferentes técnicas y enfoques genéticos para la selección de reglas lingüísticas.

1.3.1. Estudio de Diferentes Técnicas y Enfoques Evolutivos para la Selección de Reglas Lingüísticas

El uso de un AG para encontrar el subconjunto óptimo de reglas difusas está motivado en las siguientes situaciones:

- La integración de un conjunto de reglas expertas y un conjunto de reglas difusas extraídas mediante métodos de aprendizaje automático [HLV98a].
- La selección de un conjunto de reglas cooperativas a partir de un conjunto de reglas candidatas [CH00, HLV98a, INYT95, IMT97, IY03, KKS00].
- La selección de reglas de una BC previamente definida junto con la selección de las etiquetas para las variables consecuente [CQ98].
- La selección de reglas junto con un refinamiento de las funciones de pertenencia, codificando todo (reglas y parámetros) en un solo cromosoma [GJ99].
- El desarrollo de modelos difusos compactos mediante la reducción de la complejidad, para lo cual se combinan: agrupamiento difuso, reducción de reglas por técnicas ortogonales, simplificación dirigida por similitudes y optimización genética [RS00].

En esta sección, nos centraremos en el segundo enfoque, la selección de un conjunto de reglas cooperativas a partir de un conjunto de reglas candidatas, con el principal objetivo de mejorar la precisión del modelo lingüístico (seleccionando el conjunto de reglas que mejor cooperan) y obtener modelos lingüísticos más simples y, por tanto, más interpretables.

En la siguiente sección, se presenta el algoritmo básico para la selección genética de reglas a partir de un conjunto candidato de reglas difusas, haciendo especial hincapié en las distintas funciones de adaptación encontradas en la literatura especializada (principal diferencia entre las distintas propuestas realizadas). Después, se describe un enfoque específico que integra un ajuste paramétrico de la BD (aprendizaje de modificadores lingüísticos) junto con la selección genética de reglas [CdJH98]. Este enfoque es de especial interés en nuestro caso, puesto que uno de nuestros objetivos es desarrollar técnicas que nos permitan integrar un ajuste paramétrico de la BD junto con la selección de reglas a partir de un conjunto de reglas candidatas.

1.3.2. Selección Genética a Partir de un Conjunto de Reglas Candidatas

Denotemos por S_{Cand} a un conjunto de reglas candidatas de tipo “SI-ENTONCES”. El objetivo principal es seleccionar un subconjunto $S \in S_{Cand}$ que maximice una función de adaptación dada.

El proceso de selección genética se basa en un Algoritmo Genético con codificación binaria que usa cromosomas de tamaño fijo. Enumerando las reglas contenidas en el conjunto S_{Cand} desde 1 hasta m , una cadena con m bits $C = (c_1, \dots, c_m)$ denota la ausencia o presencia en el subconjunto S de las reglas candidatas contenidas en S_{Cand} , tal que,

$$\text{Si } c_i = 1 \text{ entonces } R_i \in S \text{ si no } R_i \notin S$$

La población inicial se genera introduciendo un cromosoma que representa el conjunto de reglas inicial completo S_{Cand} , es decir, con todos los genes tomando 1 como valor, $c_i = 1$. Los cromosomas restantes se inician tomando valores aleatorios entre 0 y 1 con igual probabilidad para ambos.

Dependiendo del tipo de problema al que nos enfrentemos, la función de adaptación $E(\cdot)$ estará orientada a satisfacer distintos objetivos y por lo tanto estará basada en diferentes criterios y medidas. Primero se decodifica cada cromosoma para obtener el conjunto de reglas difusas que representa, el cual se evalúa entonces de acuerdo con los criterios considerados en la función de adaptación $E(\cdot)$. En lo que sigue, se describen posibles funciones de adaptación encontradas en la literatura especializada. Para ello, se asume la existencia de un conjunto de datos de entrenamiento subyacente, utilizado para evaluar los subconjuntos de reglas codificados en los cromosomas.

- *Problemas de modelado:*

El error cuadrático medio sobre un conjunto de datos de entrenamiento, E_{CDE} , representado por la siguiente expresión [CH97, CdH01, CH00]):

$$E(C_j) = \frac{1}{2 \cdot |E_{CDE}|} \sum_{e_l \in E_{CDE}} (ey^l - S(ex^l))^2$$

donde $S(ex^l)$ es el valor de salida obtenido a partir del SBRD considerando el conjunto de reglas difusas codificado en C_j , ex^l representa los valores de las variables de estado, y ey^l es el valor de salida deseado.

Para preservar la completitud del conjunto inicial de reglas difusas también se podría considerar la *propiedad de τ -completitud* [CH97, GP97, JvSS99] (véase la Sección 1.2.1.1.3). De esta manera se podría asegurar que ningún punto del dominio quedase fuera de la partición difusa al eliminar un excesivo número de reglas.

- *Problemas de clasificación:*

El índice de error del clasificador sobre el conjunto de datos de entrenamiento se define como la razón entre el número de errores y el número de ejemplos de entrenamiento [CdJH98]:

$$E(C_j) = \frac{\text{número de errores}}{\text{número de ejemplos de entrenamiento}}$$

con el objetivo de minimizar esta función.

El número máximo de patrones correctamente clasificados usando un número mínimo de reglas [INYT95] se calcula como:

$$E(C_j) = W_{NPC} \cdot NPC(S) - W_S \cdot |S|$$

donde W_{NPC} y W_S son constantes positivas que actúan como factores de ponderación asociados a los dos siguientes objetivos: el número de patrones de entrenamiento correctamente clasificados $NPC(S)$ y el número de reglas de clasificación contenidas en el conjunto de reglas de clasificación lingüísticas $|S|$.

Una optimización restringida [IMT97] podría venir dada por:

- Restricción en el número de reglas: Para fijar un límite superior sobre el número de reglas finalmente seleccionadas N_{reglas} , la función de adaptación sería la siguiente:

$$E(C_j) = W_{NPC} \cdot NPC(S) - W_S \cdot \max\{0, |S| - N_{reglas}\}$$

con $W_{NPC} < W_S$ a fin de penalizar un excesivo número de reglas.

- Restricción en el número de patrones correctamente clasificados:

$$E(C_j) = W_{NPC} \cdot \max\{0, N_{patrones} - NPC(S)\} - W_S \cdot |S|$$

con $W_{NPC} > W_S$ a fin de penalizar un número demasiado pequeño de patrones correctamente clasificados, $N_{patrones}$.

En [IMT97], el aprendizaje se trata como un problema de optimización multicriterio, resuelto mediante un Algoritmo Genético multiobjetivo que considera los siguientes dos objetivos:

$$\text{Maximizar NPC(S)} \quad \text{Minimizar } |S|$$

Finalmente, en [IT01] se añade un tercer objetivo para minimizar el número de proposiciones en el antecedente de las reglas, con la intención de diseñar Clasificadores Basados en Reglas Difusas con una elevada capacidad de clasificación.

- *Problemas de modelado así como de clasificación:*

En [KKS00], se propone una función de adaptación apropiada para modelado y clasificación, combinando los tres siguientes aspectos:

1. Pequeño error en modelado o clasificación.
2. Mínimo número de reglas.
3. Mantenimiento del número de ejemplos cubiertos por el conjunto de reglas original.

Puesto que los criterios (1) y (2) son parcialmente contradictorios, se permite al usuario seleccionar un factor de ponderación $w \in [0, 1]$ que describe sus preferencias en términos de equilibrio entre ambos objetivos en la función de adaptación:

$$E(C_j) = \begin{cases} w \cdot \frac{Err(C_j)^2}{Err(S_{Cand})^2} + \frac{(1-w) \cdot N_R(C_j)}{N_R(S_{Cand})}, & \text{if } Err(S_{Cand}) \neq 0 \\ w \cdot E(C_j)^2 + \frac{(1-w) \cdot N_R(C_j)}{N_R(S_{Cand})}, & \text{if } Err(S_{Cand}) = 0 \end{cases}$$

donde S_{Cand} se corresponde con el conjunto de reglas difusas inicial y C_j denota el subconjunto de reglas de S_{Cand} que se está evaluando en cada momento. $Err(S_{Cand})$ es el error asociado al conjunto de reglas difusas inicial y $Err(C_j)$ el error del subconjunto que se está evaluando. $N_R(S_{Cand})$

y $N_R(C_j)$ son respectivamente el número de reglas en cada uno de dichos conjuntos. En Modelado de Sistemas, la función de error Err es una de las dos siguientes: la suma de los errores absolutos o el error cuadrático medio. Para problemas de clasificación, la función de error Err es el número de clasificaciones incorrectas.

De esta manera, cuando se consideran valores pequeños para el factor de ponderación w , se enfatiza la reducción de reglas en la función de adaptación, mientras que considerando valores elevados de w , se enfatiza la reducción del error en modelado o en clasificación. El conjunto de reglas difusas vacío llega a ser óptimo para $w = 0$, mientras que el óptimo para $w = 1$ es un conjunto de reglas con el menor error de modelado o clasificación, independientemente del número de reglas.

Para considerar el criterio (3), número de ejemplos cubiertos, la función de adaptación previa $E(C_j)$ se multiplica por un factor de penalización $p(C_j)$ definido de la siguiente manera:

$$p(C_j) = \frac{N_{D,max} - N_{D,0}}{N_{D,max} - N_{D,i}}$$

donde $N_{D,max}$ es el número de ejemplos en el conjunto de entrenamiento, $N_{D,0}$ es el número de ejemplos no cubiertos por el conjunto de reglas difusas inicial, y $N_{D,i}$ es el número de ejemplos cubiertos por el conjunto de reglas actual.

Cuando se sigue la metodología MOGUL [CdJHL99], el *proceso de multisimplificación* emplea un esquema de compartición fenotípica [CdH01] que permite que un proceso de selección genética básico genere varios subconjuntos de reglas difusas (en vez de sólo uno) mediante una Técnica de Nichos Secuenciales para funciones de optimización multimodales [BBM93].

1.3.3. Selección Genética de Bases de Reglas Integrando Modificadores Lingüísticos para Modificar las Funciones de Pertenencia

El uso de modificadores lingüísticos se ha descrito en la Sección 1.2.2.1 como una forma de mejorar el rendimiento del sistema sin tener que modificar su estructura lingüística original adaptando la forma de las funciones de pertenencia. En

este campo se han desarrollado varios ejemplos de aplicación de los modificadores lingüísticos a los SBRDs [BD95, CYP96, MP96].

Un modificador lingüístico es una función que altera la forma de la función de pertenencia de un conjunto difuso de manera que incrementa o decrementa la precisión del concepto subyacente. Como ya hemos comentado, dos de los modificadores más conocidos son el de *concentración* ‘muy’ y el de *dilatación* ‘más-o-menos’. Al aplicarse sobre un conjunto difuso A_i^k , sus expresiones son las siguientes:

$$\mu_{A_i^k}^{muy}(x) = \left(\mu_{A_i^k}(x)\right)^2, \quad \mu_{A_i^k}^{más-o-menos}(x) = \sqrt{\mu_{A_i^k}(x)},$$

y sus efectos sobre una función de pertenencia triangular podemos verlos en la Figura 1.7 de la Sección 1.2.2.1.

En esta sección se describe un proceso de selección genética de un conjunto de reglas cooperativas a partir de un conjunto inicial de reglas candidatas, integrando la selección y el aprendizaje de modificadores lingüísticos en un mismo cromosoma.

1.3.3.1. Un Proceso de Multiselección Considerando Modificadores Lingüísticos

En [CdJH98] se presenta un proceso de multiselección genética que, al mismo tiempo que elimina las reglas innecesarias a partir de un conjunto de reglas candidatas S_{Cand} , refina BCs para problemas de clasificación mediante un proceso de aprendizaje de modificadores lingüísticos. Aunque, dicho proceso se propuso como la segunda etapa de un Sistema de Clasificación Basado en Reglas Difusas Evolutivo siguiendo la metodología MOGUL [CdJHL99], también puede ser usado de manera independiente para refinar un conjunto preliminar de reglas difusas de clasificación. Las dos mayores diferencias respecto al enfoque original son las siguientes:

- El proceso de búsqueda no está dedicado únicamente a la simplificación de un conjunto preliminar de reglas difusas de clasificación sino que también es capaz de aprender el mejor conjunto de modificadores lingüísticos asociados a las funciones de pertenencia definidas bien en la BD o bien en la propia BR.

En este caso, mediante el uso de nichos el algoritmo básico se aplica secuencialmente sobre el conjunto inicial de reglas difusas. De esta manera, se pueden seleccionar distintos subconjuntos de reglas con la mejor cooperación posible en diferentes zonas del espacio de búsqueda (nichos), obteniendo tantas soluciones como el usuario desee.

- Cada vez que se obtiene una nueva definición de la BC (p.ej., un subconjunto de reglas difusas de clasificación y un conjunto de modificadores) en una iteración del proceso de multiselección, un proceso de búsqueda local optimiza dicha BC añadiendo o eliminando una regla y/o modificando un modificador lingüístico.

Como se ha dicho, el aprendizaje genético de modificadores lingüísticos se podría llevar a cabo de dos formas diferentes:

- Se obtiene un modificador para cada conjunto difuso relacionado con una etiqueta lingüística de las particiones difusas de la BD. En este caso, dicho conjunto de modificadores es compartido entre todas las reglas de la BR (*Modificadores de tipo I*).
- Se selecciona el mejor conjunto de modificadores para cada regla difusa de clasificación simple de la BR (*Modificadores de tipo II*).

Aunque en el segundo caso la semántica es específica para cada regla individual, la naturaleza descriptiva del Clasificador Difuso se mantiene con ambos enfoques.

Las modificaciones consideradas respecto al proceso básico de selección genética se describen brevemente en la siguiente sección. La Sección 1.3.3.3 introduce el algoritmo del proceso genético de multiselección. Finalmente, la Sección 1.3.3.4 describe una extensión del algoritmo considerando modificadores lingüísticos arbitrarios.

1.3.3.2. Adaptación del Método Básico de Selección Genética

Los principales componentes del proceso básico de selección genética mencionados en la Sección 1.3.2 se mantienen. Sólo cambian el esquema de codificación y la generación de la población inicial.

El *esquema de codificación* genera cromosomas de tamaño fijo con dos partes claramente diferenciadas, una relacionada con la selección de reglas y la otra referida a los modificadores lingüísticos. Enumerando las reglas contenidas en el conjunto inicial de reglas difusas S_{Cand} desde 1 hasta m , se nos presentan dos esquemas de codificación diferentes dependiendo del tipo de aprendizaje que queramos seguir para la obtención de los modificadores:

- Modificadores de tipo I:** La longitud del cromosoma es $h = m + \sum_{i=1}^n l_i$, siendo n el número de variables lingüísticas de entrada y l_i el número de etiquetas lingüísticas para cada variable i . Un cromosoma $C_j = (c_1, \dots, c_h)$ se divide en dos partes. La primera mantiene el esquema de codificación del proceso básico de selección genética que representa el subconjunto de reglas candidatas que formarán parte de la BR final. Por otro lado, la segunda parte contiene tantos genes como términos lingüísticos distintos se consideren para cada variable. En esta segunda parte, cada gen puede adoptar un conjunto posible de valores enteros, representando el valor '0' la ausencia de modificador lingüístico y el resto de valores el conjunto de los posibles modificadores lingüísticos.

La Figura 1.11 describe gráficamente el esquema de codificación y la BC resultante, compuesta por reglas difusas de clasificación con un grado de certeza asociado a todas las clases en el consecuente.

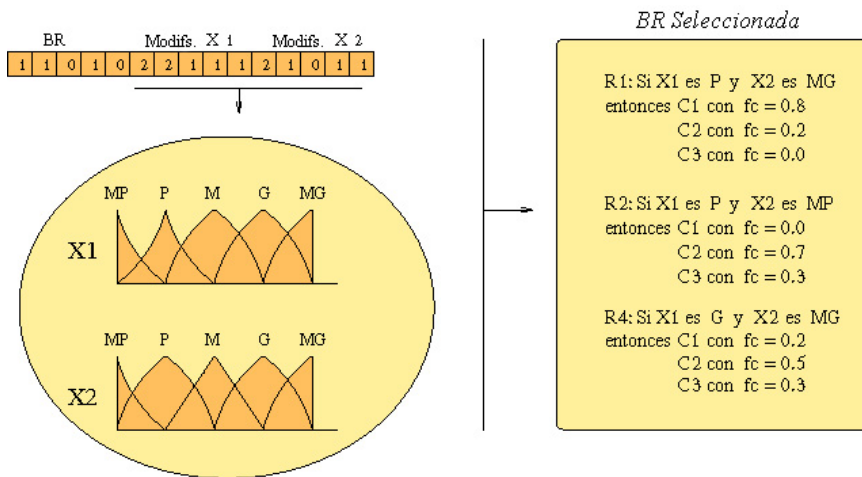


Figura 1.11: Modificadores lingüísticos de tipo I

- *Modificadores de tipo II*: La longitud del cromosoma es $h' = m \cdot (n + 1)$. El cromosoma se divide de nuevo en dos partes. Mientras los primeros m genes representan la misma información que en el caso del proceso de selección genética básico, los $m \cdot n$ restantes representan los modificadores lingüísticos para cada una de las reglas individuales, es decir, cada variable lingüística en cada regla tiene asociado un gen para acumular el modificador lingüístico asociado.

La Figura 1.12 muestra este segundo esquema de codificación así como la forma de la BC resultante.

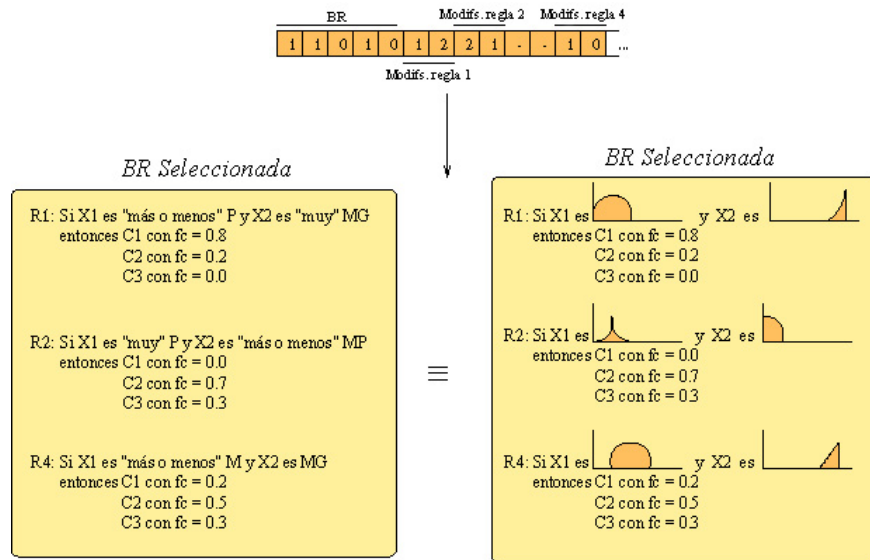


Figura 1.12: Modificadores lingüísticos de tipo II

La *población inicial* (considerando M individuos) se puede dividir en tres bloques diferentes:

- Se introduce un cromosoma que se corresponde con el conjunto inicial de reglas difusas S_{Cand} , es decir, con todo $c_i = 1, i \in \{1, \dots, m\}$, y con todos los genes que codifican los modificadores lingüísticos iniciados con el valor '0':

$$C_1 \rightarrow \text{BC original} .$$

2. Para cada posible modificador lingüístico considerado, se incluye en la población un cromosoma que en su primera parte represente el conjunto de las reglas candidatas al completo y con todos los genes de la segunda parte iniciados con el valor de dicho modificador. Suponiendo la existencia de $k - 1$ modificadores diferentes:

$$C_2 \dots C_k \rightarrow \text{BR original e idéntico modificador (modifs. } 2 \dots k) .$$

3. Los cromosomas restantes se generan aleatoriamente:

$$C_{k+1} \dots C_M \rightarrow \text{todo aleatorio} .$$

La *función de adaptación* considera dos criterios: una medida del error global de clasificación y un criterio que penaliza el incumplimiento de la propiedad de completitud [CH97, GP97, JvSS99]. Se define el *grado de completitud del conjunto de entrenamiento* para el conjunto de T reglas codificado en C_j sobre el conjunto de ejemplos E_p , como sigue:

$$GCCCE(C_j, E_p) = \bigcap_{e_l \in E_p} C_{C_j}(e_l) \text{ t.q. } C_{C_j}(e_l) = \bigcup_{i=1..T} R_i(e_l) ,$$

donde $R_i(e_l)$ es el grado de emparejamiento de la regla i y el ejemplo e_l , y τ es el mínimo grado de completitud del conjunto de entrenamiento aceptado para el conjunto de reglas difusas de clasificación final. De esta manera, la expresión final de la función de adaptación es:

$$F(C_j) = \begin{cases} E(C_j), & \text{si } GCCCE(C_j, E_p) \geq \tau \\ 1, & \text{en otro caso} \end{cases}$$

donde

$$E(C_j) = \frac{\text{número de errores}}{\text{número de casos}} .$$

Además, un *algoritmo de búsqueda local* optimiza de manera individual las definiciones de cada una de las BCs obtenidas, insertando o eliminando una regla y/o cambiando un modificador lingüístico, para inducir cambios que potencialmente mejoren el comportamiento de la BC. Como se puede observar, este es un proceso de optimización rápido y sencillo.

La búsqueda local se lleva a cabo al final de cada iteración del proceso de multiselección. Para reducir el espacio de búsqueda, el proceso de optimización

se divide en dos fases: primero se optimiza la BR y después se identifica el mejor conjunto de modificadores lingüísticos. En primer lugar, la selección de reglas se optimiza por medio de una búsqueda en el espacio de las BRs con una distancia de *hamming* de valor uno a la mejor solución actual, es decir, con una regla más o una regla menos en la BR. Finalmente, el proceso de búsqueda se extiende a la región de las soluciones con una distancia de *hamming* uno a la definición de los modificadores lingüísticos de la BC actual.

1.3.3.3. Algoritmo para el Proceso de Multiselección Genética

La selección de diferentes BRs junto con los modificadores lingüísticos asociados se lleva a cabo por medio de la Técnica de Nichos Secuenciales [BBM93] de la forma mostrada en [CdH01]. El algoritmo del proceso de multiselección genética que considera dicha técnica se muestra a continuación:

1. *Inicialización: Se iguala la función de adaptación modificada para multiselección a la función de adaptación de la selección básica: $F'(C_j) \leftarrow F(C_j)$.*
2. *Se ejecuta el proceso de selección genética básico, usando la función de adaptación modificada.*
3. *Se ejecuta el proceso de optimización local para optimizar la definición de la BC generada. Se añade la BC finalmente obtenida al conjunto solución Sol.*
4. *Se actualiza la función de adaptación para penalizar las soluciones similares al individuo actual, produciendo una nueva función de adaptación modificada.*
5. *Si Sol no contiene el número deseado de conjuntos de reglas difusas, se vuelve al paso 2, en otro caso se acaba.*

1.3.3.4. Extensión Paramétrica

Alternativamente, el modificador lingüístico aplicado a una función de pertenencia puede definirse mediante un parámetro simple a de acuerdo a

$$\mu_{A_i^k}(x)^a$$

con diferentes interpretaciones de este operador dependiendo del rango del parámetro a . Para valores $a \in (1, \infty)$ la etiqueta lingüística modificada es más precisa que la original. Por el contrario, para valores $a \in (0, 1)$, el efecto conseguido es justamente el opuesto, es decir, el modificador incrementa la imprecisión de la etiqueta lingüística original.

En [GP99] también se analiza el uso de este tipo de modificadores en el SBRDE *SLAVE*.

1.3.4. Un Ejemplo Simple de Selección de Reglas

Una aproximación simple para la selección de un subconjunto de reglas con buena cooperación es el proceso de selección propuesto en [KKS00] —proceso que será utilizado en nuestros experimentos con el nombre de algoritmo de selección de reglas (S)—. Por supuesto, el algoritmo de selección se aplica sobre un conjunto de reglas previamente aprendido. Para ello, se utiliza el proceso de generación propuesto por Wang y Mendel (WM) [WM92] para derivar una BR lingüística preliminar con la estructura clásica.

El proceso propuesto está basado en el algoritmo básico para la selección genética a partir de un conjunto de reglas candidatas presentado en la Sección 1.3.2. Así, el Algoritmo Genético utilizado sigue una codificación binaria en la que cada gen indica si una regla es considerada o no para pertenecer a la BR final (valores ‘1’ o ‘0’ respectivamente). Como esquema de selección se considera el procedimiento de muestreo universal estocástico de Baker [Bak87] junto con un mecanismo de selección elitista (que asegura mantener el mejor individuo de la generación previa). Los operadores genéticos utilizados son el cruce en dos puntos junto con la mutación clásica, descritos en el Apéndice B.

Hemos escogido unos valores genéricos de parámetros para realizar una pequeña experimentación preliminar que muestra la aplicabilidad de este enfoque de selección de reglas. La población genética está compuesta por 61 individuos, el algoritmo se ejecuta durante 1.000 generaciones y las probabilidades de cruce y

mutación se fijan a 0,6 y 0,2 respectivamente. Las particiones difusas consideradas están formadas por cinco términos lingüísticos con funciones de pertenencia triangulares (véase la Figura D.1 del Apéndice D para el caso en que se consideran siete etiquetas).

Hemos ejecutado el método para resolver el problema de estimación de la longitud de las líneas eléctricas de baja tensión mostrado en el Apéndice D. Para ello se han considerado un *modelo de validación cruzada de 5 particiones*, es decir, se han generado 5 particiones aleatorias a partir del conjunto de datos al 20 % y se combinan cuatro de ellas (80 %) para formar el conjunto de entrenamiento y la restante se utiliza para prueba, obteniéndose 5 particiones distintas. Para cada una de las 5 particiones el método propuesto se ha ejecutado 6 veces, lo que supone 30 ejecuciones de cada método en total.

Los resultados obtenidos por el método propuesto se muestran en la Tabla 1.1, donde $\#R$ indica el número medio de reglas y ECM_{ent} y ECM_{prue} indican la media del error cuadrático medio cometido sobre los conjuntos de datos de entrenamiento y prueba respectivamente (véase el Apéndice D).

Tabla 1.1: Resultados obtenidos en el problema de estimación de la longitud de las líneas eléctricas de baja tensión (selección de reglas)

Método	$\#R$	ECM_{ent}	ECM_{pru}
WM	12,4	234.712	242.147
WM-S	10,0	226.135	241.883

Como puede observarse, considerando un método simple de selección de reglas se ha obtenido un modelo más simple, en el que se han eliminado de media 2,4 reglas respecto al modelo original. Esto representa una reducción del 19,35 % del total de reglas. Además, la selección de las reglas que mejor cooperan implica una leve mejora de aproximadamente el 4 % en entrenamiento y mantiene la capacidad de generalización del modelo (o incluso la mejora). Esta mejora en la precisión de los modelos (además de la reducción del número de reglas) hace que esta técnica sea ideal para ser combinada con distintos métodos de post-procesamiento como puede ser el ajuste clásico o lateral de funciones de pertenencia (en la misma forma que se muestra en la sección 1.3.3).

1.4. Ajuste Lateral de Funciones de Pertenencia: Estudio, Análisis y Ejemplos

Como se ha dicho, uno de los procedimientos más extendidos para mejorar la precisión de los SBRDs es el *ajuste*. Este consiste en refinar la BD a través de una definición previa una vez que la BR ha sido obtenida a partir de conocimiento experto, métodos guiados por ejemplos o por simulaciones del sistema. Habitualmente, el ajuste consiste en variar la forma de las funciones de pertenencia y persigue como principal requisito mejorar la precisión del modelo lingüístico.

Los métodos de ajuste clásico [BOOB02, CHHM01, HLV95, Jan93, Kar91a] suelen representar los distintos parámetros que definen las funciones de pertenencia asociadas a las etiquetas que componen la BD, considerando 3 parámetros en el caso de funciones de pertenencia triangulares [CHHM01]. En problemas con un número elevado de dimensiones esto nos puede llevar a modelos de ajuste con demasiados parámetros, lo que podría afectar al buen funcionamiento del método de optimización utilizado.

En [AH04], se propuso un nuevo modelo de ajuste de SBRDs basado en el esquema de representación de 2-tuplas lingüísticas propuesto en [HM00], que permite el desplazamiento lateral del soporte de una etiqueta y mantiene la interpretabilidad asociada al SBRD lingüístico obtenido. Esta propuesta también introduce un nuevo modelo de representación de reglas basado en el concepto de translación simbólica [HM00], que permite desplazar lateralmente el soporte de las etiquetas considerando un único parámetro (pequeños desplazamientos a la izquierda/derecha de las funciones de pertenencia originales) con el objetivo de:

- obtener etiquetas que contengan un conjunto de ejemplos con mejor cubrimiento (búsqueda de precisión) y que conservan su forma original, y

- reducir el espacio de búsqueda del proceso de ajuste respecto al ajuste clásico [BOOB02, CHHM01, Gür99, Gür00, HLV95, Jan93, Kar91a], que normalmente considera 3 parámetros en el caso de funciones de pertenencia triangulares, facilitando la obtención de modelos óptimos.

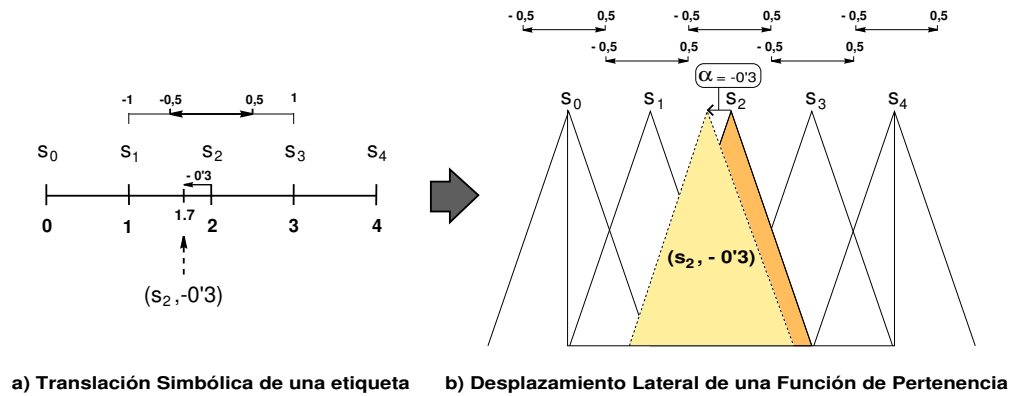


Figura 1.13: Translación simbólica de una etiqueta lingüística y desplazamiento lateral de la función de pertenencia asociada

1.4.1. Translación Simbólica de una Etiqueta

La translación simbólica de una etiqueta es un número (α) dentro del intervalo $[-0.5, 0.5]$, representando este intervalo el dominio de la etiqueta cuando ésta es movida entre sus dos etiquetas laterales adyacentes (ver la Figura 1.13.a). Consideremos un conjunto de etiquetas S ($S = \{s_1, s_2, \dots, s_n\}$) representando una partición difusa. Formalmente, para representar la translación simbólica de una etiqueta en S tenemos el par,

$$(s_i, \alpha_i), \quad s_i \in S, \quad \alpha_i \in [-0.5, 0.5].$$

Actualmente, la translación simbólica de una etiqueta implica el desplazamiento lateral de la función de pertenencia asociada. Como ejemplo, la Figura 1.13 muestra la translación simbólica de una etiqueta representada por el par $(s_2, -0.3)$ junto con el desplazamiento lateral de la función de pertenencia correspondiente.

1.4.2. Representación de Regla Basada en 2-Tuplas Lingüísticas

En [HM00], tanto el modelo de representación de 2-tuplas lingüísticas como los elementos necesarios para la comparación y agregación de información son presentados y aplicados en el marco de trabajo de la Toma de Decisión. Para ver su aplicación en el contexto de los SBRDs, consideremos un problema de control

sencillo con dos variables de entrada, una variable de salida y una BD definida por expertos, determinando las funciones de pertenencia para las siguientes etiquetas:

$$\begin{aligned} \text{Error} &\rightarrow \{N, C, P\}, \quad \nabla \text{Error} \rightarrow \{N, C, P\}, \\ \text{Potencia} &\rightarrow \{B, M, A\} . \end{aligned}$$

Basándonos en la definición de la BD, un ejemplo de regla clásica y regla con la representación de 2-tuplas puede ser el siguiente:

Regla Clásica:

Si **Error** es Cero y ∇ **Error** es Positivo
entonces **Potencia** es Alta.

Regla con la Representación de 2-Tuplas:

Si **Error** es (Cero, 0'3) y ∇ **Error** es (Positivo, -0'2)
entonces **Potencia** es (Alta, -0'1).

Si lo analizamos desde el punto de vista de la interpretabilidad, podríamos interpretar la regla ajustada como:

Si el **Error** es “mayor que Cero” y la **Variación del Error** es “menor que Positiva”
entonces la **Potencia** es “un poco menor que Alta”.

En [AH04], se proponen dos enfoques diferentes para el ajuste, uno más interpretable y otro más preciso:

- *Ajuste Global de la Semántica.* El ajuste es aplicado a nivel de las particiones lingüísticas. En todas las reglas en las que aparece el par $(X_i, \text{etiqueta})$ se mantiene el mismo valor de ajuste (α) . Por ejemplo, X_i es (Alto, 0'3) presentará el mismo valor en todas las reglas en las que inicialmente se encontrase el par “ X_i es Alto”.

Considerando este enfoque, el SBRD mantiene su interpretabilidad global. Este enfoque es análogo al ajuste clásico de la BD considerando reglas difusas descriptivas [CH97], es decir, existe una colección de conjuntos difusos globales que es común para el conjunto completo de reglas. Por lo tanto, este enfoque obtiene modelos más interpretables pero menos precisos que los obtenidos con el enfoque local.

- *Ajuste Local de las Reglas.* El ajuste es aplicado a nivel de regla. El par $(X_i, \text{etiqueta})$ es ajustado de forma diferente para cada regla en la que aparezca, basado en la medida de calidad asociada al método de ajuste (normalmente el error del sistema):

Regla k: X_i es (Alto, 0'3) (mayor que Alto)

Regla j: X_i es (Alto, -0'2) (menor que Alto)

Con este enfoque se produce cierta pérdida de la interpretabilidad global del modelo y debe ser interpretado desde un punto de vista local. Este enfoque es análogo al ajuste clásico de reglas difusas aproximativas [CH97], es decir, para cada regla se considera una semántica particular. Sin embargo, en este caso, las etiquetas ajustadas están definidas respecto a las etiquetas iniciales, manteniendo la interpretabilidad global en un cierto grado. De cualquier modo, este enfoque presenta mayor precisión y menor interpretabilidad que el enfoque global.

El principal logro es que con esta representación pasamos de considerar 3 parámetros por etiqueta [CHHM01, BOOB02, BKC96, HLV95, Jan93, Kar91a, Zhe92] a un único parámetro por etiqueta para la translación simbólica, reduciendo la complejidad del problema de ajuste y facilitando la obtención de modelos óptimos. Hay que destacar que esta reducción se produce tanto en el enfoque local como en el global. Por lo tanto, el ajuste lateral global reduce el espacio de búsqueda respecto al ajuste clásico de la BD y el ajuste lateral local reduce el espacio de búsqueda respecto al ajuste clásico de reglas difusas aproximativas. Desde el punto de vista de la interpretabilidad, el enfoque global logra obtener modelos más interpretables y el enfoque local consigue obtener modelos más precisos.

Otro aspecto importante es que el aprendizaje de los parámetros para los desplazamientos mantiene la forma original de las funciones de pertenencia (en este caso triangulares y simétricas). De esta forma, a partir del parámetro α aplicado a cada etiqueta, podemos obtener las funciones de pertenencia triangulares equivalentes, por lo que un SBRD basado en 2-tuplas lingüísticas puede ser representado como un SBRD Mamdani clásico [Mam74, MA75].

1.4.3. Sistema de Inferencia Difusa

Una vez que el modelo basado en la representación de 2-tuplas es transformado al SBRD Mamdani clásico (obteniendo las funciones de pertenencia a partir de las 2-tuplas aprendidas), podríamos considerar un razonamiento difuso clásico (ver el Apéndice A). En este caso, el razonamiento difuso utilizado es un esquema FITA (*primero inferir, después agregar*) con la *t-norma mínimo* como operador de implicación y conjunción [BD95], y la estrategia del *centro de gravedad ponderado por el grado de emparejamiento* como operador de defuzzificación [CHP97].

1.4.4. Principales Diferencias entre el Ajuste Clásico y el Ajuste Lateral

Como se ha dicho, el ajuste clásico de funciones de pertenencia normalmente ajusta cada uno de los 3 ó 4 parámetros asociados a la función de pertenencia de cada etiqueta (triangular o trapezoidal respectivamente). Esto hace que el número de parámetros que maneja crezca exponencialmente en función del número de variables y del número de etiquetas por variable.

El número de etiquetas que se consideran normalmente en cada variable es de 5 ó 7 etiquetas, y nunca más de 9. Por otro lado, el número de variables depende de cada problema específico, lo que provoca que la complejidad del problema aumente exponencialmente cuando aumenta el número de variables del sistema. Esto supone un problema para los métodos de ajuste.

Los métodos de ajuste más conocidos son los que han sido formulados usando Redes Neuronales y AGs. En el primero de los casos podemos destacar el método ANFIS [Jan93], el cual necesita representar en la segunda capa todas las posibles combinaciones de reglas, lo que provoca un desbordamiento de memoria cuando hay más de 4 ó 5 variables de entrada. Considerando AGs, es necesario utilizar un cromosoma con un elevado número de genes para representar todos los parámetros involucrados, lo que nos lleva a problemas de convergencia en los métodos evolutivos [Gol89, Hol75].

Estas consideraciones justifican el diseño de métodos de ajuste que decrementen la complejidad de los métodos clásicos para ajustar los parámetros de las funciones de pertenencia. Esta reducción de la complejidad provoca un mejor comportamiento de los métodos de ajuste que facilita la obtención de óptimos de carácter más global.

Por otro lado, el ajuste clásico podría no mantener un grado de interpretabilidad global (ver la Sección 1.2.3). En este sentido, es importante destacar que el proceso de diseño podría llevar a particiones difusas complejas en la BD, lo que podría dificultar la interpretación del sistema por un experto. En la Sección 1.2.3, la Figura 1.10 muestra un ejemplo de funciones de pertenencia con formas complejas, perdiendo la interpretabilidad global del modelo lingüístico desde el punto de vista del MDL.

Como comentamos en la Sección 1.2.1.1.3, determinar si una BD es o no interpretable es una tarea subjetiva y difícil. Aun así, algunos investigadores han estudiado esta cuestión proponiendo varias propiedades para asegurar una buena interpretabilidad durante el proceso de optimización de las funciones de pertenencia [CCHM03c, Val99a, Val99b]. Podemos aplicar varias restricciones en el proceso de diseño para obtener una BD considerando una o más de estas propiedades y mantener la comprensibilidad del modelo lingüístico en la mayor medida posible [BB03, CH97, CL00, EV00].

Naturalmente, las restricciones necesarias para asegurar la integridad semántica de la BD hacen que el proceso de derivación sea menos flexible, pero mantienen la interpretabilidad del modelo y reducen el riesgo de sobreajustar el sistema real. Por lo tanto, es muy importante aplicar estas restricciones de una forma inteligente para mantener tanto como sea posible la potencialidad/habilidad del proceso de ajuste para mejorar el comportamiento del sistema.

Desde el punto de vista de la interpretabilidad, el ajuste lateral:

- mantiene la forma original de las funciones de pertenencia, modificando lateralmente la localización del soporte, y
- el desplazamiento lateral de las etiquetas involucradas está restringido a un pequeño intervalo, asegurando el solapamiento entre dos etiquetas adyacentes en algún grado pero evitando el cruce entre sus puntos vértice.

1.4.5. Algoritmo Genético para el Ajuste Lateral

La definición automática de un sistema difuso puede ser considerada como un proceso de búsqueda o de optimización. Hoy en día, los Algoritmos Evolutivos, particularmente los AGs [Gol89, Hol75], son considerados las técnicas de búsqueda más conocidas y utilizadas. Además, la codificación genética que utilizan permiten incluir conocimiento experto previamente y utilizarlo para dirigir la búsqueda.

Por esta razón, los Algoritmos Evolutivos han sido utilizados satisfactoriamente durante los últimos años, conduciendo a la aparición de los llamados Sistemas Difusos Evolutivos (SDEs) [CHHM01, CGH⁺04] .

Los Algoritmos Evolutivos en general y, en particular los AGs, han sido ampliamente utilizados en el ajuste de los SBRDs, dando lugar a los denominados SBRDEs (ver el Apéndice C). En [AH04], consideraron el uso de los AGs para diseñar el ajuste lateral, concretamente utilizaron el modelo genético de CHC [Esh91]. CHC es un AG que presenta un buen equilibrio entre exploración y explotación, siendo una buena elección en problemas con espacios de búsqueda complejos.

Este modelo hace uso de un proceso de *Selección basado en Población* para realizar una búsqueda global adecuada, el cual para formar la siguiente población selecciona los M mejores individuos de entre los M padres y sus correspondientes hijos. Para introducir diversidad en la búsqueda, en lugar del bien conocido operador de mutación, CHC hace uso de dos mecanismos: un mecanismo de prevención de incesto y un proceso de reinicialización.

El mecanismo de prevención de incesto es considerado para determinar cuándo se aplica el operador de cruce, es decir, dos padres son cruzados si su distancia de hamming dividida entre 2 es superior a un valor umbral establecido por el experto, L . Al considerar un esquema de codificación real, necesitamos transformar cada gen mediante un Código Gray para poder calcular la distancia de hamming. Así, el valor umbral es inicializado a la mayor distancia posible entre dos padres dividida por cuatro, y es decrementado en uno cada vez que no se incluye un nuevo individuo en la población en una generación. El algoritmo aplica el mecanismo de reinicialización cuando L llega a un valor menor que 0. El algoritmo considerado por dicha técnica se muestra a continuación:

1. Se genera la población inicial con M cromosomas e inicializamos el número de evaluaciones máximo y el valor umbral L .
2. Se aplica el operador de cruce para generar los descendientes, teniendo en cuenta el mecanismo de prevención de incesto.
3. Se ejecuta el proceso de selección basado en población para generar la nueva población.
4. Se actualiza el valor umbral si no hay nuevos individuos en la población.
5. Si L es menor que 0 aplicamos el proceso de reinicialización.

6. Si no se ha alcanzado el número máximo de evaluaciones se vuelve al paso 2, en otro caso se acaba.

En la figura 1.14 podemos ver el esquema de este algoritmo.

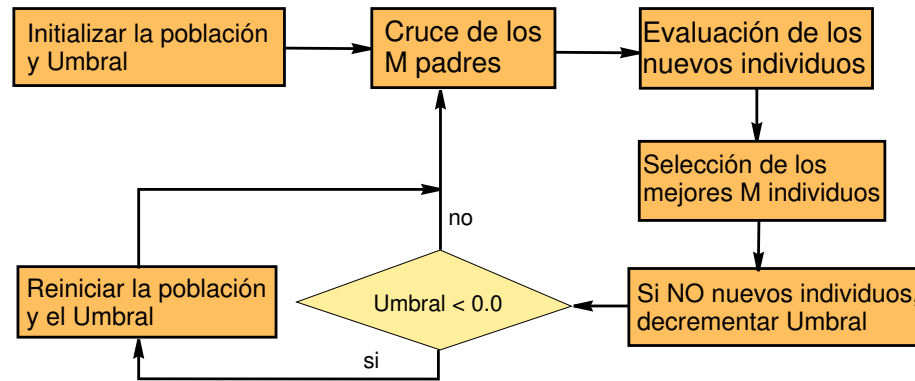


Figura 1.14: Esquema de CHC

1.4.6. Un Ejemplo Simple de Ajuste Lateral de Funciones de Pertenencia

Por supuesto, el algoritmo de ajuste lateral de funciones de pertenencia se aplica sobre un conjunto de reglas previamente aprendido. Para ello, se utiliza el proceso de generación de reglas de Wang y Mendel (WM) [WM92] para derivar una BR lingüística preliminar con la estructura clásica. Se comparará contra el ajuste clásico de funciones de pertenencia (T) [CH97], que considera 3 parámetros por etiqueta.

Para realizar una pequeña experimentación preliminar que muestre la aplicabilidad del modelo presentado se ha escogido unos valores genéricos para los parámetros. La población genética está compuesta por 51 individuos, se ejecuta durante 50.000 evaluaciones, se utilizan 30 bits por gen para la codificación Gray usada en el mecanismo de prevención de incesto y probabilidad de cruce 0,6 en el algoritmo T de ajuste clásico. Las particiones difusas consideradas están formadas por cinco términos lingüísticos con funciones de pertenencia triangulares (véase la Figura D.1 del Apéndice D para el caso en que se consideran cinco etiquetas).

El método ha sido ejecutado para resolver el problema real de la estimación del costo de mantenimiento de la línea de media tensión mostrado en el Apéndice D. Al igual que en el ejemplo de la selección de reglas, se ha considerando un *modelo de validación cruzada de 5 particiones*, y para cada una de las 5 particiones los métodos se han ejecutado 6 veces, lo que supone 30 ejecuciones de cada método en total.

La Tabla 1.2 muestra los resultados obtenidos por el método de envoltura WM, el método de ajuste clásico de funciones de pertenencia (T) y el método de ajuste lateral propuesto (GL y LL, enfoque global y local respectivamente), donde $\#R$ indica el número de reglas y ECM_{ent} y ECM_{prue} indican la media del error cuadrático medio cometido sobre los conjuntos de datos de entrenamiento y prueba respectivamente (véase el Apéndice D).

Tabla 1.2: Resultados obtenidos en el problema de estimación del costo de mantenimiento de la línea de media tensión (ajuste lateral)

Método	$\#R$	ECM_{ent}	ECM_{pru}
WM	65,0	57.605	57.934
T	65,0	18.602	22.666
GL	65,0	23.064	25.654
LL	65,0	3.664	5.858

Como puede observarse, el método LL muestra una reducción del 93,63% respecto a los modelos inicialmente obtenidos con WM y de un 80,3% respecto al ajuste clásico (T). Además, al mejorar la cooperación entre las reglas se produce una mejora en la capacidad de generalización de los modelos de aproximadamente el 90% y el 74% respecto a WM y T respectivamente. Esta mejora en la precisión es debida al uso de un único parámetro por etiqueta, reduciendo el espacio de búsqueda respecto a un ajuste clásico en el que normalmente se consideran 3 o 4 parámetros por etiqueta (en el caso de funciones de pertenencia triangulares o trapezoidales respectivamente). Así, el método LL presenta un buen equilibrio entre la complejidad del espacio de búsqueda y los resultados obtenidos. Sin embargo, el método GL no consigue mejorar los resultados obtenidos por T, esto es debido a que en problemas con espacio de búsqueda de complejidad media/baja el método GL no tiene la suficiente capacidad para sacar ventaja con respecto al

resto de las aproximaciones en este problema.

1.5. Descripción de las Propuestas Consideradas en los Sigüientes Capítulos

Después de haber analizado el uso de la selección de reglas y del ajuste lateral, vamos a finalizar este capítulo proponiendo varias líneas de trabajo que serán desarrolladas con profundidad en el resto de capítulos que componen la presente memoria:

- Proponemos un nuevo método para obtener BCs completas por medio de un aprendizaje *a priori* de la BD basado en la representación de reglas con las 2-tuplas lingüísticas [HM00]. El proceso de aprendizaje está basado en un proceso evolutivo que aprende el número óptimo de etiquetas para cada variables y los desplazamientos laterales para cada una de las etiquetas. Para cada BD generada por el algoritmo evolutivo se ejecuta un proceso de generación de reglas rápido para obtener la BR. El método que se considera como método de generación de reglas lingüísticas inicial es el método de WM.

Adicionalmente, y con el objetivo de mejorar la capacidad de generalización de los métodos, se presenta un nuevo sistema de inferencia para el caso de que haya ejemplos de entrada que no esten cubiertos por la BR.

Seleccionar el proceso de generación de reglas para el nuevo método de aprendizaje es un aspecto difícil e importante. Como hemos comentado, inicialmente presentamos el método utilizando el algoritmo de WM como primera aproximación para realizar la generación de las reglas. Sin embargo, para hacer un mejor estudio, proponemos dos nuevos métodos *ad-hoc* guiados por ejemplos para la generación de reglas, analizando su influencia en la técnica de aprendizaje de la BC.

- Por otro lado, extendemos los modelos de representación de las 2-tuplas lingüísticas para realizar también un ajuste de la amplitud del soporte de las funciones de pertenencia, con el objetivo principal de mejorar la precisión del sistema y mantener la interpretabilidad tanto como sea posible respecto

al ajuste lateral. Para ello se presenta una nueva representación simbólica con tres valores (s, α, β) mediante los cuales se representa una etiqueta. El método de ajuste consiste en la optimización de los parámetros α y β para cada etiqueta considerada en la BD.

Además, la combinación del ajuste lateral y de amplitud del soporte de las funciones de pertenencia puede presentar una sinergia positiva con métodos de selección de reglas. Por ello, presentamos un algoritmo evolutivo que nos permita combinar el ajuste lateral y de amplitud del soporte de las funciones de pertenencia con un mecanismo de selección de reglas.

Capítulo 2

Aprendizaje Automático de Sistemas Basados en Reglas Difusas Precisos y Compactos Basados en la Representación de 2-Tuplas Lingüísticas

Como hemos visto en el Capítulo 1, uno de los problemas asociados al modelado difuso lingüístico es su falta de precisión al modelar algunos sistemas muy complejos. Esto es debido a la inflexibilidad del concepto de variable lingüística, que impone duras restricciones sobre la estructura de las reglas difusas [Bas94]. Esta desventaja lleva algunas veces al MDL a alejarse del deseado equilibrio entre interpretabilidad y precisión, perdiéndose la utilidad del modelo finalmente obtenido.

Sin embargo, se pueden realizar distintas modificaciones para mejorar la precisión del MDL al mismo tiempo que se preserva su interpretabilidad intrínseca (ver la Sección 1.2 del Capítulo 1). Una forma eficiente es obtener la BC completa

a través de un proceso de aprendizaje *a priori* de la BD, es decir, un proceso para aprender la BD y un método simple para generar el conjunto de reglas asociado a cada definición de la BD. Muchos trabajos basados en este tipo de aprendizaje utilizan los AGs para aprender los parámetros de la BD.

Para facilitar el proceso de optimización genética de los parámetros de la BD recientemente se presentó un nuevo modelo de representación de reglas difusas. Éste se basa en la representación de 2-tuplas lingüísticas que permite el desplazamiento lateral del soporte de una etiqueta considerando un único parámetro (ver la Sección 1.4 del Capítulo 1), reduciendo el espacio de búsqueda y facilitando la obtención de modelos óptimos.

En este capítulo proponemos un nuevo método para obtener la BC completa por medio de un proceso de aprendizaje evolutivo *a priori* de la BD basado en la representación de 2-tuplas lingüísticas. Este método consiste en un proceso evolutivo que nos permite obtener el número óptimo de etiquetas para cada variable junto con y el desplazamiento lateral de cada etiqueta. De esta manera, para cada definición de la BD generada por el algoritmo evolutivo, se ejecuta un proceso rápido de generación de reglas que obtiene la BR correspondiente y valida su eficacia. Además, con el objetivo de mejorar la capacidad de generalización de los modelos obtenidos, proponemos un nuevo sistema de inferencia que considera los ejemplos de entrada no cubiertos.

Para el método propuesto, la selección del proceso de generación de reglas es un aspecto importante. El método de aprendizaje de la BC completa que proponemos se presenta utilizando el algoritmo de Wang y Mendel [WM92] como modelo básico. Para realizar un estudio más detallado proponemos dos métodos nuevos de generación de reglas *ad-hoc* guiados por ejemplos, analizando su influencia en la técnica de aprendizaje de la BC al ser comparados con otros métodos avanzados de generación de la BR.

Este capítulo se organiza como sigue. En la siguiente sección propondremos el nuevo sistema de inferencia para considerar los ejemplos de entrada no cubiertos. En la Sección 2.2, se introduce el esquema de aprendizaje considerado y proponemos el nuevo algoritmo de aprendizaje evolutivo para obtener la BC completa. En la Sección 2.3 presentaremos los dos métodos nuevos *ad-hoc* guiados por ejemplos para la generación de reglas y explicaremos como pueden ser integrados en el algoritmo evolutivo. En la Sección 2.4, presentamos un estudio experimental considerando los dos problemas eléctricos descritos en el Apéndice D. Finalmente, se incluye un breve resumen del capítulo en la Sección 2.5.

2.1. Un Nuevo Sistema de Inferencia Difusa

Como vimos en la Sección 1.4.2 del Capítulo 1, una vez que aplicamos los desplazamientos laterales sobre las funciones de pertenencia, el modelo basado en la representación de 2-tuplas lingüísticas se transforma en un SBRD Mamdani equivalente, pudiendo considerarse un razonamiento difuso clásico. En nuestro caso, el método de razonamiento difuso que se utiliza es la *t-norma del mínimo* jungando el papel de operador de implicación y de conjunción, y la estrategia del *centro de gravedad ponderado por el grado de emparejamiento* como operador de defuzzificación [CHP97] (esquema FITA).

Sin embargo, al buscar modelos con el menor número de reglas posibles (modelos lingüísticos compactos) podría ocurrir que hubieran zonas no cubiertas en el espacio de entrada. Teniendo en cuenta que el algoritmo de aprendizaje está guiado por medidas de error, los datos de entrenamiento no cubiertos normalmente provocan errores muy altos en el sistema y finalmente son cubiertos. Sin embargo, en problemas reales con una fuerte no linealidad, con un pequeño número de datos de entrenamiento (normalmente no totalmente representativos de la superficie del modelo) e incluso con datos con ruido, no está asegurado un buen comportamiento del modelo obtenido con datos de prueba no cubiertos (es decir, la generalización del modelo lingüístico final podría no ser buena para entradas no cubiertas). En este sentido, para considerar los datos de entrada no cubiertos en el cálculo de la salida del sistema, se aplicará el siguiente mecanismo cuando se encuentren puntos no cubiertos:

1. Se identifica la regla más cercana al punto no cubierto (distancia euclídea normalizada al vértice de las etiquetas). Las coordenadas no cubiertas del punto son remplazadas por el valor del vértice de la etiqueta correspondiente. La distancia euclídea es normalizada por la distancia máxima entre el vértice de las diferentes etiquetas consideradas para cada coordenada en la correspondiente BR.
2. Se identifica la segunda regla más cercana. Entonces, si las etiquetas de los consecuentes de ambas reglas se solapan en algún grado, solo se infiere con la regla más cercana ya que será la más representativa en un subespacio donde no se presentan fuertes cambios en el dominio de salida.
3. En otro caso, la salida final del SBRD será obtenida considerando las dos reglas más cercanas, ya que se detectan fuertes cambios en este subespacio

del dominio de salida. Para ello, las coordenadas del punto que inicialmente están cubiertas son desplazadas hacia la segunda regla, pero asegurando un grado de cubrimiento mínimo de la regla más cercana (aproximando estas coordenadas hacia el extremo de la etiqueta correspondiente de la regla más cercana a un 10 % del tamaño del soporte). Por ejemplo, supongamos que e_i es una coordenada del punto no cubierto e que inicialmente está cubierta por la etiqueta correspondiente de la regla más cercana $\{a_i^{1st}, b_i^{1st}, c_i^{1st}\}$ (extremo izquierdo, vértice y extremo derecho), y que $\{a_i^{2nd}, b_i^{2nd}, c_i^{2nd}\}$ son los puntos de definición de la etiqueta correspondiente a la segunda regla más cercana. Entonces, el nuevo valor de e'_i se calcula de la siguiente manera:

$$e'_i = \begin{cases} a_i^{1st} + (c_i^{1st} - a_i^{1st}) * 0'1, & Si \ b_i^{2nd} < b_i^{1st}, \\ c_i^{1st} - (c_i^{1st} - a_i^{1st}) * 0'1, & Si \ b_i^{2nd} > b_i^{1st}, \\ e_i, & Si \ b_i^{2nd} = b_i^{1st}. \end{cases}$$

4. Finalmente inferimos con los nuevos puntos de entrada obtenidos considerando la BC completa.

2.2. Algoritmo Evolutivo para el Aprendizaje de la Base de Conocimiento

En esta sección presentamos el esquema de aprendizaje y el algoritmo evolutivo específico basado en la representación de regla de 2-tuplas lingüísticas (ver la Sección 1.4 del Capítulo 1) propuesto para obtener la BC completa.

2.2.1. Obtención de la Base de Conocimiento mediante un Proceso de Aprendizaje A Priori de la Base de Datos

Cuando generamos la BC de un SBRD lingüístico se nos presentan dos problemas diferentes:

- el aprendizaje de la BD, por ejemplo, la especificación de los universos de discurso, el número de etiquetas y las funciones de pertenencia asociadas a ellas, y

- la generación de la BR, por ejemplo, determinar el número y la composición de las reglas finales.

En la Figura 2.1 podemos ver las formas más usuales para relizar el aprendizaje automático de la BC completa. Una forma eficiente de generar la BC completa de un SBRD consiste en obtener la BD y la BR por separado, basandonos en un aprendizaje *a priori* de la BD [FJ96, ACCH01b, CHV01a, CHV01b, Sim02, Ped04, TW04]. Esta forma de trabajar nos permite aprender el contexto más adecuado [CHV01b, ACCH01b] para cada partición difusa, lo que es necesario en distintas situaciones contextuales (diferentes aplicaciones) y para modelos de extracción de reglas diferentes.

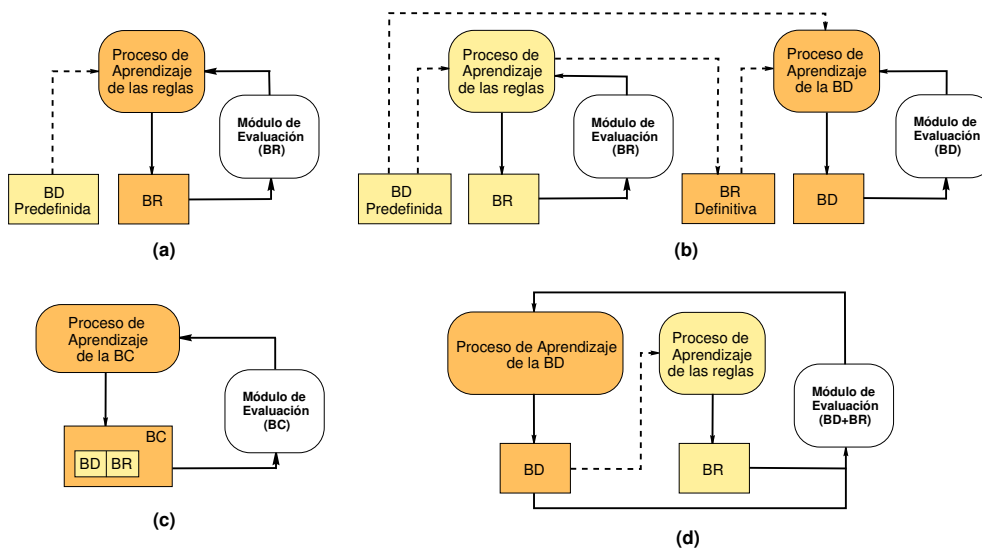


Figura 2.1: Esquema de aprendizaje de la BC

Aunque se podrían considerar distintas técnicas de optimización para el aprendizaje *a priori* de los parámetros de la BD, para esta propuesta consideraremos un algoritmo evolutivo para realizar el aprendizaje. Así, el esquema de aprendizaje considerado para el aprendizaje de la BC completa está compuesto por dos componentes principales (ver la Figura 2.1d):

- Un proceso evolutivo para aprender la BD, que nos permite definir:
 - El número de etiquetas para cada variable lingüística.

- El desplazamiento lateral de cada etiqueta.
- Un método rápido *Ad-hoc* guiado por ejemplos para generar la BR para cada definición de la BD generada por el proceso evolutivo. Así, la acción cooperativa de los dos componentes permite obtener finalmente la definición completa de la BC (BD y BR). Para presentar la propuesta utilizaremos el algoritmo de WM [WM92] como algoritmo básico de generación de reglas.

2.2.2. Algoritmo Evolutivo (Enfoque CHC)

Como se ha dicho, los Algoritmos Evolutivos en general y, en particular los AGs, han sido ampliamente utilizados para generar SBRDs. En esta propuesta consideraremos un AG específico para realizar el aprendizaje, el algoritmo CHC [Esh91]. El algoritmo CHC es un AG que presenta un buen equilibrio entre exploración y explotación, siendo una buena elección en problemas con espacios de búsqueda complejos. Este modelo genético hace uso de un mecanismo de *Selección basado en Población*, donde para formar la siguiente población se seleccionan los M mejores individuos de entre los M padres y sus correspondientes hijos (donde M es el tamaño de la población). Para provocar diversidad en la población, el enfoque CHC hace uso de un mecanismo de prevención de incesto y un mecanismo de reinicialización en lugar del bien conocido operador de mutación. Las bases del enfoque CHC han sido explicadas en la Sección 1.4.5 del Capítulo 1.

Considerando el esquema de aprendizaje propuesto en la subsección anterior, el algoritmo CHC tiene que definir tanto la granularidad de las particiones lingüísticas como los desplazamiento laterales de las etiquetas involucradas. La Figura 2.2 muestra un esquema del algoritmo propuesto considerando el enfoque CHC.

En las siguientes subsecciones veremos en detalle cada uno de los componentes necesarios para diseñar este proceso. Estos componentes son:

- codificación de la BD,
- evaluación de los cromosomas,
- generación de la población inicial,
- operadores de cruce (junto con la prevención de incesto considerada) y

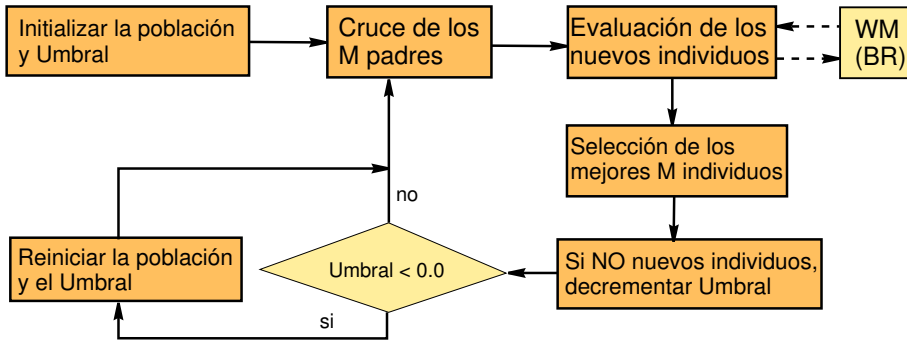


Figura 2.2: Esquema del algoritmo considerando el enfoque CHC

- mecanismo de reinicialización.

2.2.3. Codificación de la Base de Datos

Para representar las dos partes, *granularidad* y *parámetros de translación*, hemos considerado un esquema de codificación doble ($C = C_1 + C_2$):

- Número de Etiquetas (C_1): Esta parte es un vector de números enteros de tamaño n (donde n es el número de variables del sistema). Los posibles números de etiquetas para cada variable dependen del problema que estemos resolviendo y están establecidos por el experto del sistema (normalmente son $\{3, \dots, 9\}$ para las n variables):

$$C_1 = (m^1, \dots, m^n) .$$

- Desplazamientos Laterales (C_2): Esta parte es un vector de números reales de tamaño $n * 9$ (n variables con un máximo de 9 etiquetas lingüísticas por variable) en el que se codifican los desplazamientos laterales de las distintas etiquetas para cada variable. Por supuesto, si un cromosoma no tiene el número máximo de etiquetas en alguna de las variables, el espacio reservado para los valores de las etiquetas sobrantes es ignorado. Así, la parte C_2 tendrá la siguiente estructura (donde cada gen es el valor ajustado de la etiqueta correspondiente):

$$C_2 = (\alpha_1^1, \dots, \alpha_{m^1}^1, \dots, \alpha_1^n, \dots, \alpha_{m^n}^n)$$

2.2.4. Evaluación de los Cromosomas

Como hemos comentado anteriormente, para evaluar un cromosoma determinado se aplicará el bien conocido método de generación de reglas de Wang y Mendel [WM92] sobre la BD codificada en el cromosoma. Para decodificar la BD, se realiza un particionamiento difuso fuerte considerando los valores para la granularidad de la parte C_1 . Después, cada función de pertenencia es desplazada a su nueva posición considerando los valores de desplazamiento de la parte C_2 . Una vez que hemos obtenido la BC completa, y utilizando el sistema de inferencia presentado en la Sección 2.1, calculamos el Error Cuadrático Medio (ECM) y minimizamos la siguiente función:

$$F_C = w_1 \cdot ECM + w_2 \cdot NR,$$

donde, NR es el número de reglas de la BC obtenida (para penalizar un número elevado de reglas), $w_1 = 1$ y w_2 se calcula a partir del ECM y el número de reglas de la BC generada a partir de la BD con el número máximo de etiquetas para cada variable (normalmente 9) y sin considerar los desplazamientos laterales,

$$w_2 = \alpha \cdot (ECM_{max_lab} / NR_{max_lab}) ,$$

donde α es proporcionado por el experto del sistema y determina el equilibrio entre precisión y complejidad. Para valores mayores que 1.0 se buscan modelos lingüísticos con pocas reglas, para valores menores que 1.0 se buscan modelos lingüísticos con alta precisión. Una buena elección neutral es 1.0 (buena precisión y no muchas reglas).

2.2.5. Población Inicial

La población inicial esta compuesta por dos partes diferentes (las dos con el mismo número de cromosomas):

- En la primera parte, cada cromosoma tiene el mismo número de etiquetas para todas las variables del sistema, poniendo todos los parámetros de translación a cero.
- En la segunda parte, el único cambio es que cada variable podrá tener diferente número de etiquetas.

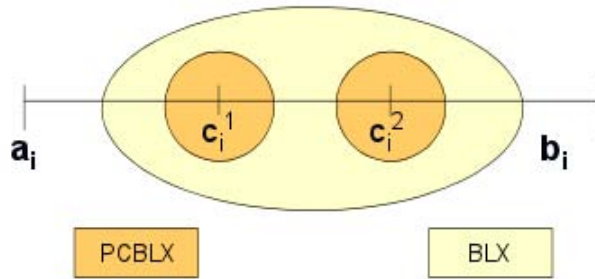


Figura 2.3: Esquema del comportamiento de los operadores BLX y PCBLX

Al no tener CHC operador de mutación los parámetros de translación no se modifican, lo que permite obtener el número de etiquetas más prometedor para cada variable. El algoritmo trabaja en este sentido hasta que se llega a la primera reinicialización.

2.2.6. Operadores de Cruce

Para obtener dos hijos a partir de los dos padres seleccionados se ha utilizado un operador de cruce diferente dependiendo de la granularidad de los padres:

- Cuando los padres tienen distinta granularidad para alguna de sus variables, se genera aleatoriamente un punto de cruce en la parte C_1 y aplicamos el operador clásico de cruce en 1 punto (ver Sección B.4 del Apéndice B) sobre las dos partes, C_1 y C_2 (exploración).
- Cuando los padres tienen la misma granularidad para cada variable, se aplica un operador basado en el concepto de entornos (los descendientes son generados alrededor de los padres) sobre la parte C_2 (explotación). Este tipo de operadores presentan una buena cooperación cuando son introducidos en modelos evolutivos en los que se fuerza la convergencia presionando sobre los descendientes (como es el caso de CHC). Particularmente, nosotros utilizaremos el operador *Parent Centric BLX* (PCBLX) [HLS03], basado en el operador $BLX-\alpha$ [ES93]. La Figura 2.3 muestra el comportamiento de este tipo de operadores.

El operador PCBLX se describe como sigue. Supongamos que $X = (x_1 \cdots x_n)$ y $Y = (y_1 \cdots y_n)$, ($x_i, y_i \in [a_i, b_i] \subset \mathfrak{R}, i = 1 \cdots n$), son dos cromosomas con codificación real que quieren ser cruzados. El operador PCBLX genera los siguientes dos descendientes:

- $O_1 = (o_{11} \cdots o_{1n})$, donde o_{1i} es un valor elegido aleatoriamente (uniformemente) dentro del intervalo $[l_i^1, u_i^1]$, con $l_i^1 = \max\{a_i, x_i - I_i\}$, $u_i^1 = \min\{b_i, x_i + I_i\}$, y $I_i = |x_i - y_i|$.
- $O_2 = (o_{21} \cdots o_{2n})$, donde o_{2i} es un valor elegido aleatoriamente (uniformemente) dentro del intervalo $[l_i^2, u_i^2]$, con $l_i^2 = \max\{a_i, y_i - I_i\}$ y $u_i^2 = \min\{b_i, y_i + I_i\}$.

Por otro lado, el mecanismo de prevención de incesto sólo se considera para determinar si se aplica o no el operador PCBLX. En nuestro caso, dos padres serán cruzados si su distancia de hamming dividida por 2 es mayor que un umbral predeterminado, L . Al considerar un esquema de codificación real (la parte C_2 será cruzada) necesitamos transformar cada gen considerando un Código Gray (código binario) con un número fijo de bits por gen ($BITSGENE$), que es determinado por el experto del sistema. Así, el valor umbral es inicializado como:

$$L = (\#GenesC_2 \cdot BITSGENE)/4,0.$$

donde $\#GenesC_2$ es el número de genes de la parte C_2 . Siguiendo el esquema original de CHC, L es decrementado en uno cada vez que no se introduce un nuevo individuo en la población en una generación. Debido a que la convergencia es muy lenta, en nuestro caso, L también será decrementada en uno cuando no se consiga ninguna mejora respecto al mejor cromosoma de la generación anterior.

2.2.7. Mecanismo de Reinicialización

Al no utilizarse un operador de mutación se considera un mecanismo de reinicialización para evitar los óptimos locales [Esh91], que será aplicado cuando el valor umbral L sea menor que cero. En este caso, se copian en todos los cromosomas de la población la parte C_1 de la mejor solución global e inicializamos aleatoriamente los parámetros de la parte C_2 dentro del intervalo $[-0'5, 0'5)$. Además, si la mejor solución global se modifica respecto a la última reinicialización ésta es también incluida en la población (la explotación sólo continua mientras hay convergencia). Este modo de operar fué inicialmente propuesto por los autores de

CHC como una posible mejora del algoritmo cuando este es aplicado en algunos tipos de problemas [Esh91].

2.3. Dos Nuevos Métodos *Ad-Hoc* de Generación de Reglas Guiados por Ejemplos y su Integración en el Aprendizaje Evolutivo de la BD A Priori

Como hemos comentado anteriormente, la selección del proceso de generación de reglas en el aprendizaje *a priori* de la BD es un aspecto importante. Este método debería permitir al proceso de aprendizaje obtener BCs precisas y, al mismo tiempo, compactas. Además, este método debe tener un tiempo de ejecución lo más corto posible ya que se ejecuta cada vez que una BD es evaluada. En esta sección discutiremos sobre que tipos de métodos podrían favorecer este comportamiento, proponiendo dos métodos nuevos *ad-hoc* guiados por ejemplos y diseñados específicamente para esta tarea.

Para seleccionar el método de generación de la BR en este problema podemos distinguir dos posibilidades:

1. *La primera es utilizar métodos avanzados para obtener reglas con la mejor precisión posible.* En [CCZ04], los autores analizaron diferentes métodos *ad-hoc* guiados por ejemplos para proponer un nuevo enfoque, llamado MM que presenta una mayor capacidad de aproximación. Este se basa en combinar un método guiado por ejemplos (el algoritmo WM [WM92]) con un método guiado por rejilla difusa (*Input Space oriented Strategy*, ISS [IMT97]), añadiendo reglas al modelo lingüístico obtenido por el algoritmo WM en los subespacios de entrada difusos que tienen ejemplos y para los que no se ha generado una regla. Aunque de primeras este enfoque podría parecer una buena elección, la utilización de este tipo de métodos avanzados en nuestro proceso de aprendizaje presenta algunas desventajas que debemos tener en cuenta. Por un lado, el tiempo computacional necesario para este tipo de métodos es mayor que el que se necesita para métodos más simples. Además, la mejora en precisión obtenida por un método sofisticado a menudo provoca un incremento en el número final de reglas (modelos menos interpretables). Por otro lado, algunos estudios [CHV01a, CHV01b, ACCH01b] han mostrado que el comportamiento del sistema es más sensible al aprendizaje de la

BD que a la composición de la BR. Por lo tanto, no está claro que la generación de BRs más elaboradas favorezca el aprendizaje de la BD respecto al uso de BRs más simples, ya que las BRs obtenidas podrían sesgar el aprendizaje de una BD óptima.

2. *La segunda posibilidad es utilizar algoritmos simples y rápidos que favorezcan el aprendizaje de las funciones de pertenencia.* Este tipo de métodos obtienen rápidamente un conjunto de reglas básicas basadas en los ejemplos con el mejor cubrimiento en cada subespacio difuso. Por lo tanto, la calidad del conjunto de reglas obtenido depende directamente de la obtención de una definición de la BD satisfactoria para cubrir bien los ejemplos más representativos del comportamiento del sistema. Esta forma de trabajar lleva al aprendizaje *a priori* de la BD a obtener BDs mejores y BRs más simples, es decir, BCs más precisas y compactas. Además, la generación de modelos más simples es una forma de evitar el sobreaprendizaje, lo que facilita la obtención de modelos que también presenten una buena capacidad de generalización [IY03]. Por esta razón, un algoritmo básico y simple como WM funciona tan bien cuando es integrado en un método basado en el aprendizaje *a priori* de la BD [CHV01a, CHV01b].

Puesto que nuestro principal objetivo es obtener SBRDs precisos pero también compactos, y también es un factor importante el tiempo de cálculo, enfocaremos nuestra atención hacia métodos que pertenezcan a la segunda posibilidad, es decir, métodos simples que favorezcan el aprendizaje de las funciones de pertenencia. Un ejemplo de este tipo de métodos es el algoritmo WM, considerado en la sección anterior como el primer enfoque para derivar la BR. En las siguientes subsecciones propondremos dos métodos nuevos *ad-hoc* simples guiados por ejemplos que permiten generar modelos más simples y manteniendo la misma o similar precisión. Están basados en una selección más general de los consecuentes, considerando un conjunto de los ejemplos mejor cubiertos en lugar de sólo considerar el ejemplo mejor cubierto. El uso de consecuentes más generales también mejora la capacidad de generalización de los modelos obtenidos, reduciendo el efecto de los puntos de ruido.

Además, en los experimentos y con el propósito de hacer una comparativa, también consideraremos el algoritmo MM para generar la BR dentro del esquema de aprendizaje propuesto en la Sección 2.2.

2.3.1. Método de Generación de Reglas Basado en Medias (Algoritmo AV)

Este método persigue obtener consecuentes más generales basándose en la media ponderada de las salidas de los ejemplos cubiertos a un cierto grado, siendo éstas ponderadas por el grado de emparejamiento con el antecedente de la reglas. Al utilizar la salida media se decrementa la influencia de los puntos de ruido. El método está basado en la existencia de una BD predefinida y un conjunto de ejemplos de entrenamiento entrada/salida $E = \{e_1, \dots, e_l, \dots, e_m\}$ con $e_l = (x_1^l, \dots, x_{n-1}^l, y^l)$, $l \in \{1, \dots, m\}$, donde m es el tamaño del conjunto de datos, y $n-1$ es el número de variables de entrada. La BR se genera siguiendo los siguientes pasos:

- Inicialmente la BR está vacía.
- Para cada ejemplo e_l en E :
 1. Generar el antecedente de la regla con las etiquetas que mejor cubren los datos de entrada del ejemplo $(x_1^l, \dots, x_{n-1}^l)$.
 2. Si no hay una regla con el mismo antecedente en la BR:
 - a) Seleccionar los ejemplos con un grado de emparejamiento ¹ mayor que δ , donde $\delta \in [0.5, 1]$ es un valor proporcionado por el experto del sistema. Si no hay ningún ejemplo con un grado de emparejamiento mayor que δ , seleccionamos todos los ejemplos que estén cubiertos en algún grado.
 - b) Calcular \bar{M} como la media de las salidas de los ejemplos seleccionados ponderada por su grado de emparejamiento.
 - c) Generar el consecuente de la regla con la etiqueta que mejor cubre \bar{M} .
 - d) Añadir la regla obtenida a la BR.

El parámetro δ determina como de generales o de específicos son los consecuentes obtenidos respecto a los ejemplos cubiertos. Al depender este parámetro del problema que estamos resolviendo, de la granularidad y de las posiciones de las funciones de pertenencia, este parámetro será obtenido junto con la BD en

¹Aplicando sobre el antecedente obtenido la *t-norma mínimo* como operador de conjunción.

el proceso evolutivo. Al final de esta sección explicaremos como el método propuesto es incluido en el aprendizaje evolutivo *a priori* de la BD. Este método es algo más lento que el algoritmo WM ya que se tiene que calcular el grado de emparejamiento de todos los ejemplos para el antecedente de cada regla.

2.3.2. Método de Generación de Reglas Basado en Moda (Algoritmo MO)

Este método persigue obtener consecuentes más generales seleccionando la etiqueta modal de entre todas las etiquetas propuestas por los ejemplos que son cubiertos por el antecedente de las reglas a un cierto grado. Así, los puntos de ruido, que normalmente aparecen con poca frecuencia, no serían considerados para calcular la salida. Este método también se basa en la existencia de una BD predefinida y un conjunto de datos de entramiento entrada-salida, E . El algoritmo consiste en los siguientes pasos:

- Inicialmente la BR está vacía.
 - Para cada ejemplo e_i en E :
 1. Generar el antecedente de la regla con las etiquetas que mejor cubren los datos de entrada del ejemplo $(x_1^l, \dots, x_{n-1}^l)$.
 2. Si no hay una regla con el mismo antecedente en la BR:
 - a) Seleccionar los ejemplos con un grado de emparejamiento¹ mayor que δ , donde $\delta \in [0,5, 1]$ es un valor proporcionado por el experto del sistema.
 - b) Si se ha seleccionado algún ejemplo:
 - Calcular la etiqueta que mejor cubre la salida de cada ejemplo seleccionado, contando el número de veces que cada etiqueta de salida es seleccionada.
 - Generar el consecuente de la regla con la etiqueta de salida modal, es decir, la etiqueta de salida que más veces ha sido seleccionada.
- Si no se seleccionó ninguno:
- Generar el consecuente de la regla exactamente como en WM (el consecuente de la regla generada para el ejemplo con mayor grado de cubrimiento sobre las n variables).

c) Añadir la regla obtenida a la BR.

Como en el método de la sección anterior, el parámetro δ se obtendrá junto con la BD en el proceso evolutivo. Este método puede ser implementado exactamente igual que el algoritmo WM pero contando la frecuencia de los consecuentes propuestos si el grado de emparejamiento es mayor que δ y seleccionando finalmente las etiquetas modales. Por lo tanto, este es más rápido que el algoritmo AV y muy similar al algoritmo WM.

2.3.3. Integración de los Métodos Propuestos en el Aprendizaje Evolutivo de la BC

Para considerar estos algoritmos en el enfoque propuesto para el aprendizaje *a priori* de la BD, se reemplaza directamente el algoritmo WM por los algoritmos AV o MO y se codifica el parámetro δ junto con la BD, por lo que el método propuesto en la Sección 2.2 debe incluir también el aprendizaje del parámetro δ . A continuación, explicamos solo los cambios necesarios respecto a este algoritmo:

- *Esquema de Codificación* — El *esquema de codificación* es modificado para añadir el parámetro δ que será considerado para obtener la BR:

$$C = C_1 + C_2 + \delta .$$

- *Población Inicial* — La población inicial se genera como en el caso original pero asignándole a los parámetros δ un valor aleatorio dentro del intervalo $[0.5, 1]$.
- *Operador de Cruce* — Se considera el operador de cruce presentado en la Sección 2.2.6, generando aleatoriamente el parámetro δ dentro del intervalo $[0.5, 1]$ cruzan ambas partes, C_1 y C_2 . Por otro lado, cuando sólo se cruza la parte C_2 , el operador PCBLX también se aplica sobre los parámetros δ .
- *Mecanismo de Reinicialización* — Se aplica como en el algoritmo original pero poniendo los parámetros δ a un valor aleatorio dentro del intervalo $[0.5, 1]$. Si la mejor solución global se ha modificado respecto a la última reinicialización ésta también es incluida en la población junto con su parámetro δ .

El parámetro δ sólo es necesario en el proceso de generación de reglas, pero una vez que el proceso de aprendizaje haya finalizado y se haya obtenido la BC con representación de 2-tuplas, este parámetro deja de ser necesario.

2.4. Estudio Experimental

Para evaluar el buen comportamiento de los algoritmos propuestos (aprendizaje *a priori* de la BD considerando los algoritmos WM, AV o MO) se han considerado los dos problemas eléctricos descritos en detalle en el Apéndice D:

- *Estimación de la longitud de las líneas de baja tensión.* Este problema tiene dos variables de entrada, lo que supone un espacio de búsqueda pequeño (baja complejidad). Sin embargo, este problema es muy interesante ya que el sistema es *fuertemente no lineal* y los datos disponibles se limitan a un número reducido de ejemplos que además contienen dos puntos con *ruido*. Todas estas desventajas hacen que modelar esta superficie sea realmente complicado, lo que provoca un fuerte sobreaprendizaje en los modelos obtenidos.
- *Estimación del costo de mantenimiento de la línea de media tensión.* Este problema tiene cuatro variables de entrada y el conjunto de datos disponible está compuesto por un número representativo de ejemplos bien distribuidos. En este caso, los métodos de aprendizaje suelen obtener un número considerable de reglas. Además, este problema tiene *un espacio de búsqueda grande* (complejidad media/alta).

Resolver estos dos problemas correctamente no es tarea fácil, en general, los métodos que presentan una buena capacidad de aproximación no generalizan muy bien en problemas del mundo real con pocos datos, mal distribuidos e incluso con ruido, por lo que es fácil sobreajustar los modelos obtenidos. En este sentido, los métodos propuestos presentan una buena capacidad de aproximación (especialmente en el segundo problema), al mismo tiempo que una buena capacidad de generalización (especialmente en el primer problema). En las siguientes secciones realizamos un análisis de los métodos propuestos.

2.4.1. Experimentos

En los siguientes tres párrafos se presenta una breve descripción de los distintos métodos estudiados en el análisis experimental (la Tabla 2.1 resume las principales características de cada método):

Tabla 2.1: Métodos considerados para el estudio experimental

Ref., Año	Método	Tipo de Aprendizaje
[WM92], 1992	WM	Método AHGE
[CCZ04], 2004	MM	Método AHGE mejorado basado en WM
[CCdVH05], 2005	COR	Reglas Cooperativas usando el algoritmo SA-MP
[AH04], 2004	WM+GL	Ajuste lateral global a partir de WM
Métodos que consideran el aprendizaje <i>a priori</i> de la BD		
[CHV01a], 2001	Gr-MF	Gr. + Parámetros de las FP + BR por WM
[CHV01b], 2001	GA-WM	Gr. + Factores de Escala + Dominios + BR por WM
[CCHV04], 2004	GA-COR	Gr. + Factores de Escala + Dominios + BR por COR
Algoritmos estudiados		
—	GLD-MM	Gr. + Parámetros Laterales Globales + BR por MM
—	GLD-WM	Gr. + Parámetros Laterales Globales + BR por WM
—	GLD-AV	Gr. + Parámetros Laterales Globales + BR por AV
—	GLD-MO	Gr. + Parámetros Laterales Globales + BR por MO

AHGE: *Ad-Hoc* Guiado por Ejemplos; SA-MP: Sistema de Hormigas Mejor-Peor;
Gr.: Granularidad; FP.: Funciones de Pertenencia

- Los métodos propuestos son identificados como GLD-WM, GLD-AV y GLD-MO (presentados en la Sección 2.2 y 2.3 respectivamente). El algoritmo GLD-MM se considera reemplazando directamente el algoritmo WM por el método MM [CCZ04] en el algoritmo GLD-WM, y se incluye a efectos comparativos.
- Los algoritmos WM [WM92], COR [CCdVH05] (con el Sistema de Hormigas Mejor-Peor) y MM [CCZ04] son considerados como un método de generación de reglas simple y dos avanzados para obtener mejores BRs a partir de

una BD predefinida. También mostraremos los resultados obtenidos por el método de ajuste WM+GL [AH04], basado en la representación de 2-tuplas lingüísticas. Todos estos métodos serán considerados como referencia, ya que los algoritmos propuestos se basan en ellos. Las particiones lingüísticas iniciales están compuestas por *cinco términos lingüísticos* con funciones de pertenencia triangulares y distribución uniforme.

- Por otro lado, también estudiaremos tres métodos que utilizan el aprendizaje a priori de la BD para obtener una BC completa. El primero, Gr-MF [CHV01a], aprende la granularidad (número de etiquetas) de las particiones difusas y los parámetros de las funciones de pertenencia (sus tres puntos de definición). Los algoritmos GA-WM [CHV01b] y GA-COR [ACCH01b] aprenden la granularidad, factores de escala y los dominios (es decir, el dominio de la variable o el rango donde realizar el particionamiento difuso) para cada variable del sistema. Estos métodos obtienen la BR utilizando respectivamente los algoritmos WM y COR.

Para llevar a cabo los experimentos hemos considerado un *modelo de validación cruzada con 5 particiones*, es decir, se generan 5 particiones de datos² al 20% y se combinan cuatro de ellas (80%) para formar el conjunto de entrenamiento y la restante se utiliza para prueba, obteniéndose 5 particiones distintas. Para cada una de las 5 particiones cada método se ha ejecutado 6 veces, lo que supone 30 ejecuciones de cada método en total. En las tablas de resultados de las siguientes secciones se muestran los resultados medios de las 30 ejecuciones de cada método. Además, se aplica un test *t-student* (con 95% de confianza) para comprobar si las diferencias que hay entre los resultados mostrados en las tablas son significativas.

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). Finalmente, los siguientes valores son los que hemos considerado como parámetros para cada método: 50 individuos, 50.000 evaluaciones, 30 bits por gen para la codificación Gray y el conjunto $\{3, \dots, 9\}$ como número posible de etiquetas para todas las variables del sistema; 0,2 y 0,6 como probabilidad de mutación y

²Los conjuntos de datos están disponibles en: <http://decsai.ugr.es/~casillas/fmlib/>.

de cruce en el caso de los algoritmos Gr-MF, GA-WM y GA-COR; para reducir el tiempo de ejecución del algoritmo GA-COR los autores proponen utilizar solo 2.000 evaluaciones. En esta memoria utilizaremos el mismo número de evaluaciones en los dos problemas. El factor α de la función de adaptación se ha fijado a 1 en los métodos propuestos (GLD) para ambos problemas. Sin embargo, para obtener modelos con diferentes niveles de precisión y simplicidad, en el segundo problema (problema con más variables y reglas) también probaremos con $\alpha = 3$.

2.4.2. Estimación de la Longitud de las Líneas de Baja Tensión

Los resultados obtenidos por los métodos analizados se muestran en la Tabla 2.2, donde $\#R$ es el número medio de reglas, ECM_{ent} y ECM_{pru} son el error medio obtenido sobre los conjuntos de entrenamiento y prueba respectivamente, σ_{ent} y σ_{pru} son la desviación típica de los resultados de entrenamiento y prueba respectivamente, $h:m:s$ es el tiempo medio de ejecución en un Intel Centrino (1.73 GHz, 512 MB de RAM) y donde $t-test$ representa la siguiente información:

★ representa el mejor resultado medio.

+ significa que el mejor resultado tiene significativamente mejor comportamiento que el método que tenga asignado este símbolo.

= denota que los resultados son estadísticamente iguales acorde al test $t-student$ con un 95 % de confianza.

Analizando los resultados presentados en la Tabla 2.2 se puede resaltar que:

- Aunque los métodos propuestos (métodos GLD) no obtienen los mejores resultados en entrenamiento, el equilibrio entre aproximación y generalización es bastante bueno en un problema con ruido y con pocos datos de ejemplo. Teniendo en cuenta este hecho y los elevados errores de prueba del resto de los métodos (en menos medida WM+GL y GA-COR), podríamos afirmar que estos métodos presentan sobreaprendizaje mientras que los métodos GLD se acercan más al verdadero comportamiento del sistema. Además, los algoritmos GLD-WM, GLD-AV y GLD-MO obtienen los modelos con menor número de reglas.

- El uso de un método de generación de reglas más avanzado (MM), mejora ligeramente el error en entrenamiento de WM aunque añade muchas más reglas. Aunque los métodos GLD favorecen la generación de modelos más simples, el algoritmo GLD-MM presenta un número más elevado de reglas y una menor capacidad de generalización que el resto usando métodos más simples de generación de reglas. Por ello, en este problema, el uso de este algoritmo no implica ninguna ventaja.
- Como podemos ver, el tiempo computacional de los métodos de generación de reglas afecta notablemente al tiempo de los métodos de aprendizaje *a priori* de la BD. Dicho efecto puede verse en los métodos GA-COR y GLD-MM. No obstante, en el caso de GA-COR se obtiene el mejor error de entrenamiento.

La Figura 2.4 muestra una de las 30 BCs obtenidas por el algoritmo GLD-MO en este problema. Esta figura muestra cómo pequeñas variaciones en las funciones de pertenencia permiten obtener importantes mejoras en el comportamiento de

Tabla 2.2: Resultados obtenidos en el problema de estimación de la longitud de las líneas con el parámetro $\alpha = 1$ para la función de adaptación

Método	#R	ECM _{ent}	σ_{ent}	t-test	ECM _{pru}	σ_{pru}	t-test	h:m:s
WM	12,4	234.712	32.073	+	242.147	24.473	+	00:00:00,01
MM	19,2	232.974	32.471	+	244.763	23.141	+	00:00:00,2
COR	22,0	180.995	7.794	+	220.320	32.492	+	00:00:04
WM+GL	12,4	166.674	11.480	+	189.216	14.743	+	00:01:03
Gr-MF	21,9	157.083	5.426	+	242.913	59.205	+	00:01:31
GA-WM	15,8	160.441	6.616	+	210.444	46.773	+	00:01:24
GA-COR	12,6	152.313	3.590	★	193.052	25.561	+	02:37:49
GLD-MM	11,2	160.374	5.020	+	182.139	15.531	+	00:01:47
GLD-WM	8,8	162.295	4.059	+	177.507	16.234	=	00:01:26
GLD-AV	8,7	159.689	6.324	+	172.881	22.449	=	00:01:54
GLD-MO	7,9	163.696	4.696	+	172.765	18.895	★	00:01:25

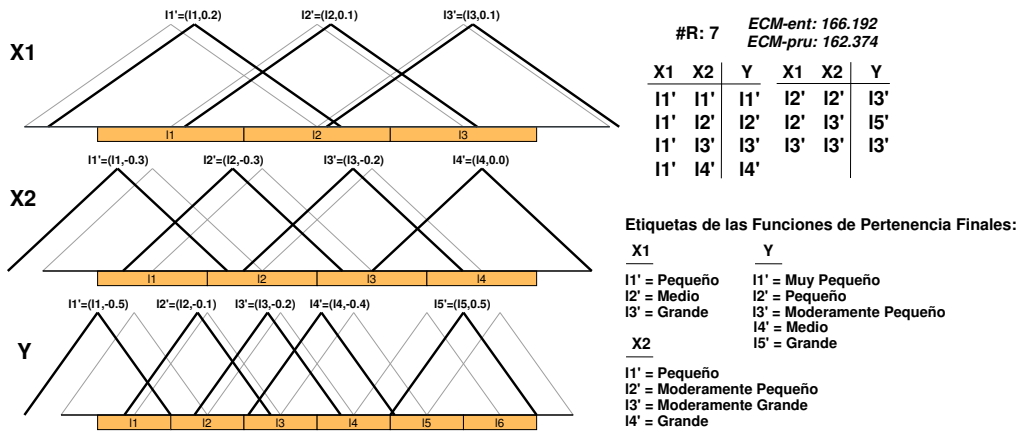


Figura 2.4: BD con/sin desplazamientos laterales (negro/gris), BR y parámetros de desplazamiento de un modelo obtenido por el algoritmo GLD+MO (las etiquetas no utilizadas han sido eliminadas en la figura)

los SBRDs obtenidos. Así, las dos variables de entrada presentan respectivamente tres y cuatro etiquetas cuyas funciones de pertenencia están más o menos uniformemente distribuidas, lo que facilita al experto encontrar su significado correspondiente. La variable de salida presenta cinco etiquetas que están balanceadas hacia la izquierda, representando una mayor concentración de ejemplos con salidas pequeñas (ver la Figura D.2 del Apéndice D). Sin embargo, de nuevo las etiquetas están más o menos bien distribuidas a la izquierda y a la derecha de la etiqueta central, por lo que puede ser fácil asignarle un nombre a las mismas.

2.4.3. Estimación del Costo de Mantenimiento de la Línea de Media Tensión

En la Tabla 2.3 se muestran los resultados obtenidos por los métodos analizados (este tipo de tabla fue descrita en la subsección anterior). Analizando los resultados presentados en la Tabla 2.3 podemos resaltar los siguientes hechos:

- En este problema, las desventajas de utilizar el método de generación de reglas más avanzado son aún más obvias. En este caso, el método MM presenta mejoras significativas en entrenamiento y prueba respecto al algoritmo

Tabla 2.3: Resultados obtenidos en el problema de estimación del costo de mantenimiento

Método	#R	ECM _{ent}	σ_{ent}	t-test	ECM _{pru}	σ_{pru}	t-test	h:m:s
WM	65	56.136	1.498	+	56.359	4.686	+	00:00:00,02
MM	266	44.958	1.926	+	45.598	6.553	+	00:00:00,4
COR	41	39.640	566	+	41.683	1.599	+	00:01:00
WM+GL	65	23.064	1.479	+	25.654	2.611	+	00:08:15
Gr-MF	93,9	16.726	2.403	+	18.824	3.651	+	00:07:53
GA-WM	51,1	23.014	2.143	+	24.090	3.667	+	00:10:26
GA-COR	17,8	20.360	1.561	+	22.830	3.259	+	36:45:41
Métodos propuestos con el parámetro $\alpha = 1$ en la función de adaptación								
GLD-MM	261,7	9.617	1.025	*	11.825	2.168	*	14:19:02
GLD-WM	57,5	10.218	1.044	=	12.088	1.972	=	00:09:10
GLD-AV	60,4	11.856	2.014	+	13.604	2.969	+	00:22:06
GLD-MO	54,9	10.568	1.017	=	12.718	2.001	=	00:09:20
Métodos propuestos con el parámetro $\alpha = 3$ en la función de adaptación								
GLD-MM	154,1	12.093	1.995	+	14.020	2.606	+	12:41:02
GLD-WM	41,2	13.074	2.040	+	15.196	2.757	+	00:09:23
GLD-AV	36,4	14.868	2.939	+	16.885	4.095	+	00:13:32
GLD-MO	34,4	13.687	2.108	+	16.050	2.095	+	00:07:55

WM pero a costa de obtener un número excesivo de reglas y de incrementar el tiempo computacional. Esto hace que el método GLD-MM tarde más de 14/12 horas para obtener un modelo con más de 200/100 reglas y sin mejoras significativas respecto a los métodos que utilizan algoritmos más simples. Un hecho parecido se puede observar en GA-COR, aunque en este caso se obtuvieron modelos muy compactos con una precisión aceptable para dicho número de reglas.

- Los métodos propuestos en este trabajo muestran una importante reducción del ECM sobre los datos de entrenamiento y de prueba en un problema con un espacio de búsqueda grande. Esto es debido a que la representación

de 2-tuplas lingüísticas permite reducir el espacio de búsqueda respecto al aprendizaje clásico de funciones de pertenencia, facilitando la generación de modelos óptimos. Véase el método Gr-MF, que teóricamente podría obtener como mínimo los mismos resultados del método GLD-WM y que aprende los tres puntos de definición de las funciones de pertenencia, siendo una generalización del método GLD-WM.

- En el caso de los métodos GLD-AV y GLD-MO, se obtuvieron modelos más simples y con una precisión similar a los obtenidos por los métodos GLD-WM o GLD-MM. Además, los modelos lingüísticos obtenidos mantienen la interpretabilidad a un buen nivel al mantener la forma original de las funciones de pertenencia. En este sentido, podemos destacar el método GLD-MO respecto a AV, al obtener el menor número de reglas, el mejor error y el menor tiempo de computación en los dos problemas.

La Figura 2.5 presenta la BC obtenida por el método GLD-MO en una de sus 30 ejecuciones sobre este problema y con $\alpha = 3$. Analizando el modelo lingüístico podemos observar una configuración de la BD similar a la obtenida en el problema anterior. Las funciones de pertenencia se encuentran más o menos bien distribuidas, lo que facilita asignar un significado a las etiquetas correspondientes.

2.5. Sumario

En este capítulo hemos propuesto un nuevo método de aprendizaje de la BC por medio de un aprendizaje evolutivo *a priori* de la BD (granularidad y parámetros de translación) basado en el modelo de representación de reglas de las 2-tuplas lingüísticas y en un nuevo sistema de inferencia. Además, se han propuesto dos métodos nuevos *ad-hoc* guiados por ejemplos para analizar su influencia y la de otros métodos de generación de reglas en el enfoque de aprendizaje propuesto. Así, podemos concluir:

- La utilización del esquema de aprendizaje junto con el modelo de representación de reglas de las 2-tuplas lingüísticas y el nuevo sistema de inferencia permite realizar una importante reducción del espacio de búsqueda, facilitando la obtención de modelos lingüísticos más precisos y compactos.

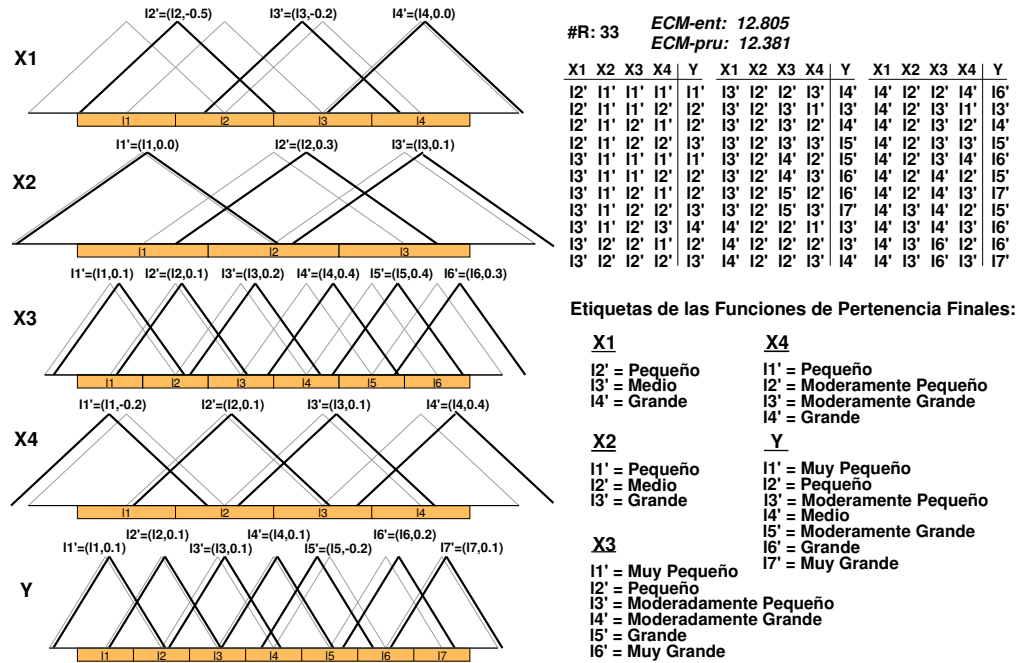


Figura 2.5: BD con/sin desplazamientos laterales (negro/gris), BR y desplazamientos de un modelo obtenido por el algoritmo GLD+MO (las etiquetas no utilizadas han sido eliminadas en la figura)

- El uso de un método de generación de reglas simples para la búsqueda de las reglas básicas que mejor cubran los datos de ejemplo favorece el aprendizaje de una BD mejor y la generación de BRs con un número menor de reglas. Al influir más el aprendizaje de la BD que la composición de la BR en el comportamiento del sistema, este tipo de métodos facilitan la obtención de modelos más precisos y compactos.
- Además, al considerar un enfoque global y al mantener la forma de las funciones de pertenencia iniciales, la interpretabilidad de los modelos obtenidos se mantiene en un alto grado respecto al aprendizaje clásico de sistemas difusos.

Finalmente, destacar que el uso de diferentes valores de α para penalizar el número de reglas en el segundo problema pone de manifiesto la existencia de

modelos óptimos con diferentes niveles de precisión y simplicidad.

Capítulo 3

Ajuste Evolutivo basado en la Representación de 3-Tuplas Lingüísticas: Ajuste-LA

Como comentamos en la Sección 1.4 del Capítulo 1, para mejorar la precisión de los SBRDs (manteniendo el equilibrio deseado entre interpretabilidad y precisión) recientemente se ha presentado un método nuevo de ajuste de SBRDs que permite realizar un ajuste lateral evolutivo de las funciones de pertenencia. Dicho método está basado en la representación de 2-tuplas lingüísticas, que permite la translación simbólica de una etiqueta considerando un único parámetro por etiqueta.

Durante todo el proceso de ajuste la amplitud del soporte de las funciones de pertenencia permanece fija. La amplitud determina la especificidad de una etiqueta y permite una mejora potencial de la precisión del modelo, ya que puede ayudar a determinar la región de mejor cubrimiento de la etiqueta.

En este capítulo se presenta una nueva representación simbólica con 3 valores (s, α, β) que representan respectivamente una etiqueta, el desplazamiento lateral y la variación en amplitud del soporte de esta etiqueta. Basándonos en esta nueva representación, presentamos un nuevo método de ajuste que nos permite realizar un ajuste evolutivo Lateral y de Amplitud de las funciones de pertenencia (ajuste-

LA).

Por otro lado, como se ha dicho en la Sección 1.3 del Capítulo 1, los métodos de reducción de reglas permiten mejorar la interpretabilidad de los modelos al agregar reglas repetidas y/o seleccionar un subconjunto de reglas a partir de un conjunto de reglas difusas dado. La combinación de los métodos de ajuste con métodos de selección de reglas puede presentar una sinergia positiva, reduciendo el espacio de búsqueda, facilitando la legibilidad del sistema e incluso mejorando su precisión. De hecho, recientemente se presentó un trabajo [AAFB⁺06] donde se pone de manifiesto la sinergia positiva que existe entre el ajuste lateral de funciones de pertenencia y la selección de reglas, donde los modelos obtenidos por la combinación de las dos técnicas presentaban más o menos la misma (incluso mejor en algunos casos) precisión y un número menor de reglas que los modelos que obtienen las técnicas por separado, mejorándose así el equilibrio entre interpretabilidad y precisión.

Por ello, en este capítulo el ajuste-LA también se combina con un método de selección de reglas con el objetivo de mejorar la precisión del sistema y mantener al mismo tiempo parte de la interpretabilidad, tanto como sea posible, respecto al ajuste lateral. El uso de un nuevo parámetro β mejora la precisión del SBRD pero provoca la pérdida de parte de la interpretabilidad, ya que en principio no tiene un significado asociado. Sin embargo, el método de selección de reglas extrae las reglas ajustadas que son más útiles (las reglas con una mejor información), lo que permite obtener BCs más simples y, por lo tanto, aumentar la interpretabilidad de los modelos. Esta forma de trabajar proporciona un equilibrio razonable entre precisión e interpretabilidad.

Una vez presentados los algoritmos básicos, estos serán utilizados para resolver distintos problemas de modelado y control. Dependiendo de como se obtenga el SBRD inicial para estos problemas podemos distinguir dos situaciones:

1. *SBRDs generados a partir de un proceso aprendizaje automático de reglas.* El SBRD inicial se obtiene con un proceso de aprendizaje automático a partir de un conjunto de pares de datos de entrada/salida que reflejan el comportamiento del sistema real que se desea modelar.
2. *SBRDs proporcionados por expertos.* El conjunto de reglas difusas candidatas se obtiene a partir de expertos, las cuales serán posteriormente redefinidas mediante el ajuste-LA y el ajuste-LA junto con la selección de las reglas. Basándonos en los algoritmos presentados para el post-

procesamiento de SBRDs generados a partir de métodos automáticos, (ajuste-LA y ajuste-LA con selección de reglas), modificaremos su modelo evolutivo para aplicarlos a la optimización de controladores difusos para sistemas de climatización HVAC en grandes edificios. En la Sección D.2.3 del Apéndice D podemos encontrar una descripción detallada del problema.

El capítulo se organiza en las siguientes secciones. En primer lugar, en la Sección 3.1, se describe el nuevo modelo de representación basado en 3-tuplas lingüísticas, presentando el sistema de inferencia utilizado y analizando su influencia en el equilibrio entre interpretabilidad y precisión. A continuación, en la Sección 3.2 se presenta el método evolutivo para realizar el ajuste-LA y su combinación con un mecanismo de selección de reglas aplicado a modelos lingüísticos obtenidos a partir de métodos básicos de MDL, realizando un análisis sobre el sistema de inferencia utilizado. En la Sección 3.3, se aplica el método de ajuste-LA y su combinación con la selección de reglas a la optimización de controladores difusos para sistemas de climatización HVAC en grandes edificios. Por último, se incluye un breve resumen del capítulo en la Sección 3.4.

3.1. Ajuste Lateral y de Amplitud de Funciones de Pertenencia

Este enfoque es una extensión del ajuste lateral [AH04] para permitir también un ajuste de la amplitud del soporte de las funciones de pertenencia. Para adaptar los desplazamientos laterales y las amplitudes de los soportes de las funciones de pertenencia proponemos una nueva representación de regla que considera 2 parámetros, α y β , que representan respectivamente el desplazamiento lateral y la variación de la amplitud de una etiqueta. Así, cada etiqueta puede ser representada por una 3-tupla (s, α, β) , donde α es un número en el intervalo $[-0'5, 0'5)$ que expresa el dominio de una etiqueta cuando esta es movida entre sus dos etiquetas adyacentes (como en la representación de 2-tuplas), y β es también un número en el intervalo $[-0'5, 0'5)$ que permite incrementar o reducir la amplitud del soporte de una etiqueta hasta un 50% de su tamaño original. Consideremos un conjunto de etiquetas $S = \{s_1, s_2, \dots, s_n\}$ que representan una partición difusa. Formalmente tenemos la tripleta,

$$(s_i, \alpha_i, \beta_i), \quad s_i \in S, \quad \{\alpha_i, \beta_i\} \in [-0'5, 0'5)$$

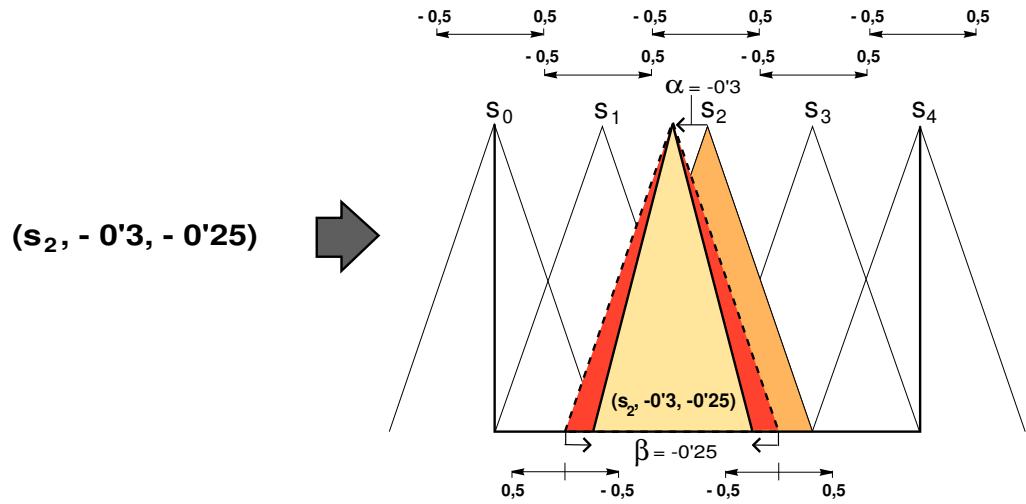


Figura 3.1: Variación lateral y de amplitud de la función de pertenencia asociada a la etiqueta s_2

Por ejemplo, la figura 3.1 muestra la etiqueta con la representación de 3-tuplas $(s_2, -0'3, -0'25)$ junto con su desplazamiento lateral y la variación de la amplitud de la función de pertenencia correspondiente. Supongamos que c_{s_2} y a_{s_2} son el extremo derecho e izquierdo del soporte de s_2 , y que Sop_{s_2} es su tamaño. El soporte de la nueva etiqueta $s'_2 = (s_2, -0'3, -0'25)$ puede calcularse de la siguiente forma:

$$Sop_{s'_2} = Sop_{s_2} + \beta * Sop_{s_2}, \text{ donde } Sop_{s_2} = c_{s_2} - a_{s_2}$$

En [AH04], se propusieron 2 representaciones diferentes para el ajuste lateral de funciones de pertenencia, un enfoque global y otro local. Nosotros también consideraremos estas dos posibilidades para el ajuste-LA, una más interpretable y otra más precisa:

- Ajuste Global de la Semántica. En este caso el ajuste es aplicado a nivel de las particiones lingüísticas. Así, en todas las reglas en las que aparece el par $(X_i, \text{etiqueta})$ se mantienen los mismos valores ajustados. Por ejemplo, X_i es (Alto, 0'3, 0'1) presentará los mismos valores en todas las reglas en las que el par " X_i es Alto" fuera inicialmente considerado.

- Ajuste Local de las Reglas. En este caso, el ajuste se aplica a nivel de regla. El par $(X_i, \text{etiqueta})$ es ajustado de forma diferente para cada regla en la que aparezca:

Regla k: X_i es (Alto, 0'3, 0'1)

Regla j: X_i es (Alto, -0'2, 0'25)

Hay que destacar que en ambos enfoques, global y local, se consideran funciones de pertenencia triangulares simétricas y una inferencia difusa FITA [CHP97], por lo que no tiene sentido un ajuste de la amplitud de los consecuentes. Por lo tanto, el parámetro β solo será aplicado sobre los antecedentes. Así, si consideramos el mismo problema de control simple de la Sección 1.4.2 del capítulo 1, un ejemplo de regla con la representación de 3-tuplas es (la variación de la amplitud se aplica solo en los antecedentes):

Si el **Error** es (Cero, 0'3, 0'1) y la ∇ **Error** es (Positiva, -0'2, -0'4)
Entonces la **Potencia** es (Alta, -0'1)

Si la analizamos desde el punto de vista de la interpretabilidad, podríamos interpretar los desplazamientos laterales como en los capítulos anteriores. Sin embargo, no está claro un significado para el factor de amplitud β . En cualquier caso, si las funciones de pertenencia están más o menos bien distribuidas y no se producen cambios grandes en las amplitudes, un experto quizás podría renombrar estas etiquetas para darles un significado más o menos representativo.

Por otro lado, el uso del factor β (amplitud) es parecido al uso de factores de escala no lineales [CdJH98, LCT01] o de modificadores lingüísticos [CdJH98, GP99]. Sin embargo, hay algunas diferencias con respecto a estos enfoques:

- a) Si utilizamos factores de escala no lineales o modificadores lingüísticos un ejemplo que estuviera cubierto por una etiqueta no puede dejar de estarlo y *vice versa*, lo que impone algunas restricciones a la búsqueda. Determinar la amplitud de una función de pertenencia es una forma de decidir qué ejemplos son cubiertos y cuales no, agrupando mejor un conjunto de datos. Por

lo tanto, ajustar la amplitud del soporte de una función de pertenencia puede ayudar a,

- reducir el número de ejemplos negativos (cubiertos en los antecedentes pero no en los consecuentes),
 - incrementar el número de ejemplos positivos (cubiertos en los antecedentes y consecuentes),
 - reducir el número de reglas si se considera un método de selección de reglas.
- b) Al revés que los factores de escala no lineales o que los modificadores lingüísticos, el ajuste de la amplitud del soporte mantiene la forma de las funciones de pertenencia (triangular y simétrica). Así, a partir de los parámetros α y β aplicados a cada etiqueta podríamos obtener las funciones de pertenencia equivalentes, por lo que el SBRD final podría representarse como un SBRD clásico de tipo Mamdani [Mam74, MA75].

3.1.1. Sistema de Inferencia Difusa

El razonamiento difuso utilizado es el *mínimo* como operador de implicación y conjunción, y la estrategia del *centro de gravedad ponderado por el grado de emparejamiento* como operador de defuzzificación [CHP97] (esquema FITA). Este tipo de inferencia es aplicado una vez que el modelo de representación de 3-tuplas es representado como su SBRD Mamdani clásico equivalente.

Teniendo en cuenta que el soporte de las funciones de pertenencia que componen las reglas puede ser reducido y desplazado y que también puede utilizarse un método de selección de reglas, podría ocurrir que hayan zonas en el espacio de entrada que no estén cubiertas. Los datos de entrenamiento no cubiertos provocan errores muy grandes en el sistema y finalmente son cubiertos como resultado del proceso de optimización. Sin embargo, en problemas reales con una fuerte no linealidad, con un pequeño número de datos de entrenamiento (normalmente no totalmente representativos de la superficie del modelo) e incluso con datos con ruido, no está asegurado un buen comportamiento del modelo obtenido con datos de prueba no cubiertos (es decir, la generalización del modelo lingüístico final podría no ser buena para entradas no cubiertas). Además, cuando se presentan puntos de ruido en los datos de entrenamiento, el proceso de optimización podría llegar a no cubrirlos para mejorar el comportamiento con los datos fia-

bles disponibles. Para evitar estos problemas, consideraremos el nuevo sistema de inferencia presentado en la Sección 2.1 del Capítulo 2.

3.1.2. Equilibrio entre Interpretabilidad y Precisión

Como hemos comentado anteriormente, la amplitud del soporte de las funciones de pertenencia permanece fija en el ajuste lateral. Sin embargo, en el método que proponemos también realizamos un ajuste de la amplitud con el objetivo de determinar la especificidad o generalidad de una etiqueta. Esto tiene algunas ventajas (mejoras en la precisión) y desventajas (perdida de interpretabilidad) respecto al ajuste lateral, donde solo se ajusta la posición de las funciones de pertenencia:

- Ventajas:
 1. Hay una mejora potencial de la precisión respecto al ajuste lateral ya que la representación de 3-tuplas permite un mayor grado de libertad.
 2. Determinar el subconjunto de puntos de entrenamiento que debería ser cubierto es más fácil, consiguiendo reglas más específicas y consistentes.
- Desventajas:
 1. Los modelos obtenidos son menos interpretables que los obtenidos con la representación de 2-tuplas. Sin embargo, al poder ser transformados en un sistema Mamdani clásico, los modelos obtenidos considerando la representación de 3-tuplas pueden ser considerados como mínimo tan interpretables como los modelos obtenidos con los métodos de ajuste clásico, que ajustan los 3 puntos de definición de las funciones de pertenencia.
 2. El espacio de búsqueda se incrementa respecto al ajuste lateral, haciendo más difícil obtener modelos óptimos. Sin embargo, este enfoque todavía supone una reducción del espacio de búsqueda respecto al ajuste clásico (utiliza 1 parámetro menos), y es aún bien manejado mediante un uso correcto de la técnica de búsqueda.

Tanto el ajuste lateral como el ajuste-LA presentan un buen equilibrio entre interpretabilidad y precisión. Sin embargo, el método que proponemos se encuentra más cercano a la precisión, mientras que el ajuste lateral está más próximo

a la interpretabilidad. La elección de como de interpretable o como de preciso debe ser el modelo depende normalmente de las necesidades del usuario para un problema específico, lo que condicionará la selección del tipo de ajuste.

A partir de ahora, debemos tener en cuenta que cuando se comenten diferencias entre el ajuste clásico, el ajuste lateral y el ajuste-LA, se considerará por separado el enfoque global y local, es decir, por un lado se compararán los enfoques globales y por otro lado se compararán los enfoques locales. Así, el ajuste-LA global reduce el espacio de búsqueda respecto al ajuste clásico de la BD y el ajuste-LA local reduce el espacio de búsqueda respecto al ajuste clásico de reglas difusas aproximativas. Desde el punto de vista de la interpretabilidad también podemos considerarlos por separado, ya que el enfoque global obtiene modelos más interpretables y el enfoque local obtiene modelos más precisos.

En la siguiente sección se presenta el método evolutivo para el ajuste-LA basado en la representación de 3-tuplas con enfoque global y local para el ajuste automático basado en ejemplos.

3.2. Aplicación sobre Reglas Obtenidas a Partir de Ejemplos

En este caso, el ajuste-LA se aplica sobre SBRDs obtenidos a partir de métodos de aprendizaje automático. En primer lugar, veremos el algoritmo para realizar el ajuste-LA. En segundo lugar, se propone un nuevo método que nos permite combinar el ajuste-LA con un mecanismo de selección de reglas.

3.2.1. Algoritmo Evolutivo para el Ajuste Lateral y de Amplitud de Funciones de Pertenencia

Teniendo en cuenta estas características interesantes de los AGs (ver la Sección 1.4.5 del Capítulo 1), en esta sección proponemos un nuevo SBRDE para realizar el ajuste-LA de funciones de pertenencia.

Para ello consideramos el modelo evolutivo de CHC [Esh91]. CHC es una AG que presenta un buen equilibrio entre exploración y explotación, siendo una buena elección en problemas con espacios de búsqueda complejos. A continuación explicaremos los componentes necesarios para el diseño del proceso de ajuste

evolutivo. Estos son:

- Modelo evolutivo de CHC.
- Esquema de codificación y población inicial.
- Evaluación de los cromosomas.
- Operador de cruce.
- Mecanismo de reinicialización.

3.2.1.1. Modelo Evolutivo (Enfoque CHC)

Las bases del enfoque CHC han sido explicadas anteriormente en la Sección 1.4.5 del Capítulo 1. Para este método, al considerarse muchos más parámetros, el valor inicial del umbral puede ser muy grande, lo que puede provocar problemas de convergencia. Con el objetivo de hacer independiente la reducción del umbral L_T del número total de genes ($\#Genes$) y del número de bits por gen utilizados en el Código Gray ($BITSGENE$), en nuestro caso, el umbral será decrementado cada vez en un $\varphi\%$ de su valor inicial (siendo φ determinado por el usuario, normalmente un 10%).

La Figura 1.14 del Capítulo 1 muestra un esquema para este algoritmo.

3.2.1.2. Esquema de Codificación y Población Inicial

Teniendo en cuenta que en [AH04] se propusieron dos tipos diferentes de ajuste (ajuste global de la semántica y ajuste local de las reglas), hay dos clases distintas de esquema de codificación. En ambos se utiliza una codificación real (ver la Sección B.2 del Apéndice B), es decir, los parámetros reales son las unidades de representación del AG (genes). Dependiendo del tipo de ajuste que queramos realizar, consideraremos uno u otro de los siguientes esquemas de codificación:

- Ajuste Global de la Semántica: Se considera un vector con los parámetros de las particiones difusas, parámetros de desplazamiento lateral (C^L) y parámetros de amplitud (C^A). Supongamos el siguiente número de etiquetas por variable: (m^1, \dots, m^n) , donde n es el número de variables del sistema ($n - 1$ variables de entrada y 1 variable de salida). Así, un cromosoma

tiene la siguiente forma (donde a cada gen se asocia al valor ajustado de la correspondiente etiqueta),

$$C_T = (C^L + C^A),$$

$$C^L = (c_{11}^L, \dots, c_{1m^1}^L, \dots, c_{n1}^L, \dots, c_{nm^n}^L),$$

$$C^A = (c_{11}^A, \dots, c_{1m^1}^A, \dots, c_{(n-1)1}^A, \dots, c_{(n-1)m^n}^A).$$

- **Ajuste Local de las Reglas:** Se considera un vector con los parámetros de desplazamiento lateral (C^L) y de amplitud (C^A) de las reglas. Supongamos que el SBRD tiene N reglas, (R_1, R_2, \dots, R_N) con n variables de sistema ($n - 1$ variables de entrada y 1 variable de salida). Entonces, la estructura del cromosoma es,

$$C_T = (C^L + C^A),$$

$$C^L = (c_{11}^L, \dots, c_{1n}^L, \dots, c_{N1}^L, \dots, c_{Nn}^L),$$

$$C^A = (c_{11}^A, \dots, c_{1n-1}^A, \dots, c_{N1}^A, \dots, c_{Nn-1}^A).$$

Para hacer uso de la información disponible, el SBRD inicial obtenido a partir de métodos de aprendizaje automático se incluye en la población como una solución. Así, la población inicial está compuesta por un individuo con todos sus genes a 0 (sin desplazamientos ni variaciones de la amplitud para representar la solución inicial), y por el resto de los individuos inicializados asignando a cada uno de sus genes un valor aleatorio dentro del intervalo $[-0.5, 0.5]$.

3.2.1.3. Evaluación de los Cromosomas

Para evaluar un cromosoma se utiliza el bien conocido ECM:

$$ECM = \frac{1}{2 \cdot |E|} \sum_{l=1}^{|E|} (F(x^l) - y^l)^2,$$

donde $|E|$ es el tamaño del conjunto de datos, $F(x^l)$ es la salida obtenida por el SBRD decodificado a partir del cromosoma cuando se considera el ejemplo l -ésimo, e y^l es la salida deseada.

3.2.1.4. Operador de Cruce

El operador de cruce esta basado en el concepto de entornos (los descendientes son generados alrededor de sus padres). Como se ha dicho, este tipo de operadores presentan una buena cooperación cuando son introducidos en modelos evolutivos, forzando la convergencia mediante una presión sobre los descendientes (como es el caso de CHC). La Figura 2.3 del Capítulo 2 representa el comportamiento de este tipo de operadores, mostrando como los genes de los descendientes se encuentran alrededor de los genes de sus padres (PCBLX), o alrededor de la zona amplia determinada por los genes de los padres BLX- α [ES93]. Particularmente, en este caso se considera el operador PCBLX que esta basado en el operador BLX- α [HLS03].

El operador PCBLX se ha descrito en la Sección 2.2.6 del Capítulo 2 y se define de la siguiente manera. Supongamos que $X = (x_1 \cdots x_n)$ y $Y = (y_1 \cdots y_n)$, ($x_i, y_i \in [a_i, b_i] \subset \mathbb{R}, i = 1 \cdots n$), son dos cromosomas con codificación real que quieren ser cruzados. El operador PCBLX genera los siguientes dos descendientes:

- $O_1 = (o_{11} \cdots o_{1n})$, donde o_{1i} es un valor elegido aleatoriamente (uniformemente) dentro del intervalo $[l_i^1, u_i^1]$, con $l_i^1 = \max\{a_i, x_i - I_i\}$, $u_i^1 = \min\{b_i, x_i + I_i\}$, y $I_i = |x_i - y_i|$.
- $O_2 = (o_{21} \cdots o_{2n})$, donde o_{2i} es un valor elegido aleatoriamente (uniformemente) dentro del intervalo $[l_i^2, u_i^2]$, con $l_i^2 = \max\{a_i, y_i - I_i\}$ y $u_i^2 = \min\{b_i, y_i + I_i\}$.

Los padres X y Y serán nombrados de forma diferente: X se llamará *padre femenino*, y Y se denominará *padre masculino*. Así, para que el método pueda generar dos descendientes primero considera X como padre femenino (Y como masculino) y después como padre masculino (Y como femenino).

3.2.1.5. Mecanismo de Reinicialización

Para escapar de óptimos locales este algoritmo utiliza un mecanismo de reinicialización [Esh91]. En este caso, el mejor cromosoma se mantiene y el resto de la población es generada sumando a cada gen del mejor cromosoma un valor aleatorio dentro del intervalo $[-0.125, 0.125]$. Si el valor resultante es menor (o mayor) que -0.5 (0.5) éste es remplazado por el valor extremo -0.5 (0.5). Se siguen

los principios de CHC [Esh91], aplicando el proceso de reinicialización cuando el valor umbral (L_T) es menor que 0 (ver la Figura ?? de la Sección 3.2.1.1).

3.2.2. Selección de Reglas y Ajuste Lateral y de Amplitud de Funciones de Pertenencia

Como se comentó en la Sección 1.3 del Capítulo 1 las técnicas de selección de reglas simplifican los modelos obtenidos e incluso mejoran su precisión. Este tipo de técnicas para la reducción de reglas podrían ser fácilmente combinadas con otras técnicas de post-procesamiento para obtener SBRDs más compactos y precisos. En este sentido, hay algunos trabajos en los que se ha combinado la selección de reglas con el ajuste de funciones de pertenencia, codificando todo (reglas y parámetros) en un mismo cromosoma [AAFGH06, AAFB⁺06, CCdJH05, GJ99] (véase la Sección 1.3.3 del Capítulo 1 donde se presenta un método que combina la selección de reglas con el ajuste de funciones de pertenencia mediante el uso de modificadores lingüísticos). En este apartado se combina la selección de reglas con el ajuste-LA de funciones de pertenencia.

3.2.2.1. Sinergia Positiva entre Ambos Enfoques

Hay varias razones que justifican la sinergia positiva entre la selección de reglas y el ajuste de las funciones de pertenencia. Algunas de ellas son:

- Los procesos de ajuste se ven afectados cuando se incluyen reglas conflictivas y/o erróneas en la BR inicial. Cuando la BR del modelo que está siendo ajustado contiene reglas malas (incrementan gravemente el error del sistema), el proceso de ajuste permite reducir el efecto de este tipo de reglas, adaptándolas junto con el resto para evitar su mal comportamiento. Esto impone fuertes restricciones que reducen la habilidad del proceso para obtener modelos lingüísticos precisos. Además, en algunos casos también afecta a la interpretabilidad del modelo, ya que las funciones de pertenencia que componen las reglas malas no tienen ni la mejor forma ni la mejor localización para representar la información que se está modelando.

Este hecho aumenta junto con la complejidad del problema (es decir, en problemas con un elevado número de variables o reglas) y cuando el método de generación de reglas no asegura la generación de reglas con buena

calidad (por ejemplo, cuando la BR se obtiene solo a partir de conocimiento experto). En estos casos, el proceso de ajuste es muy complicado debido a que la habilidad del proceso de búsqueda se dedica a reducir el mal comportamiento de algunas reglas en lugar de mejorar el comportamiento del resto. En estos casos, la selección de reglas podría ayudar al mecanismo de ajuste eliminando las reglas que realmente empeoran la precisión del modelo.

- En algunas ocasiones las reglas redundantes no se pueden eliminar utilizando únicamente el proceso de selección de reglas. Esto es debido a que en algunas ocasiones este tipo de reglas pueden reforzar la acción de algunas reglas pobres, mejorando la precisión del modelo. El ajuste de las funciones de pertenencia puede hacer innecesario el refuerzo que realizan estas reglas, ayudando al proceso de selección de reglas a eliminar las reglas redundantes.

Por lo tanto, la combinación de la selección de reglas con el ajuste podría provocar importantes mejoras en la precisión del sistema, al mismo tiempo que se mejora la interpretabilidad (el número de reglas disminuye) [CCdJH05, GJ99]. Sin embargo, en algunos casos el espacio de búsqueda considerado es demasiado grande cuando combinamos las dos técnicas, lo que puede provocar la obtención en modelos subóptimos [CCdJH05].

En la siguiente sección proponemos la selección de un conjunto de reglas cooperativas a partir de un conjunto de reglas difusas candidatas junto con el aprendizaje de los parámetros α y β . Esta propuesta persigue los siguientes objetivos:

- Mejorar la precisión del modelo lingüístico, seleccionando el conjunto de reglas que mejor cooperan al mismo tiempo que el ajuste-LA mejora la localización y la amplitud de las funciones de pertenencia.
- Obtener modelos lingüísticos más simples y, por lo tanto, más comprensibles, mediante la eliminación de las reglas innecesarias.
- Favorecer la acción combinada del ajuste y de las estrategias de selección (lo que implica un espacio de búsqueda grande) mediante la simplificación del espacio de búsqueda provocada por el ajuste-LA (solo dos parámetros por etiqueta).

3.2.2.2. Algoritmo Evolutivo

Para seleccionar el subconjunto de reglas que mejor cooperan y obtener los parámetros α y β , consideramos un AG que condifica todo en cada cromosoma (reglas y parámetros). Este algoritmo se basa en el método propuesto en la Sección 3.2.1, considerando también el modelo evolutivo de CHC [Esh91].

Para ello tenemos que tener en cuenta la existencia de genes binarios (selección de reglas) y reales (desplazamientos laterales y variación de la amplitud) en un mismo cromosoma. Por lo tanto, el algoritmo propuesto en la Sección 3.2.1 se modificará para poder considerar un esquema de codificación doble y para aplicar los operadores genéticos apropiados a cada una de las partes del cromosoma. Así, los cambios considerados poder realizar la selección de reglas junto con el ajuste son los siguientes:

- *Esquema de Codificación* — Consideramos un *esquema de codificación* doble para los dos procesos, ajuste-LA y selección de reglas:

$$C = C_T + C_S .$$

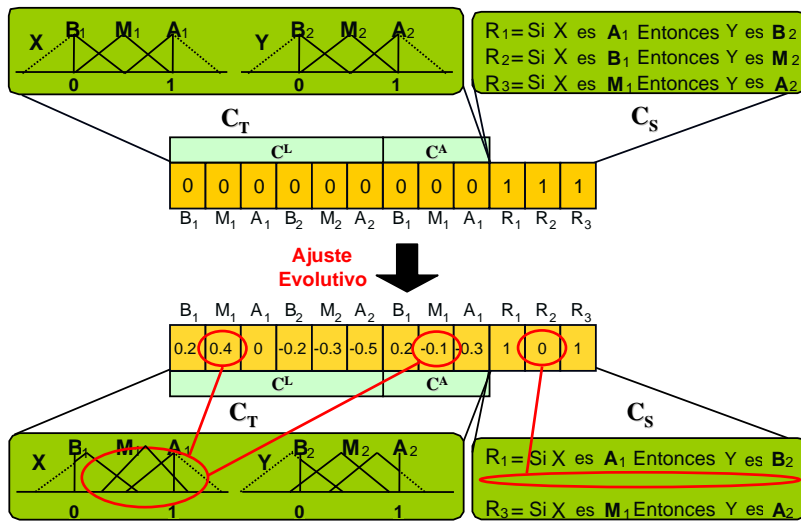
En este caso, el enfoque previo (parte C_T) es combinado con la selección de reglas mediante la unión del vector binario C_S que determina qué regla está seleccionada y cual no (alelos '1' y '0' respectivamente).

Considerando las N reglas del conjunto de reglas preliminar/candidatas, la parte $C_S = (c_1, \dots, c_N)$ del cromosoma representa un subconjunto de reglas que formarán la BR final, tal que:

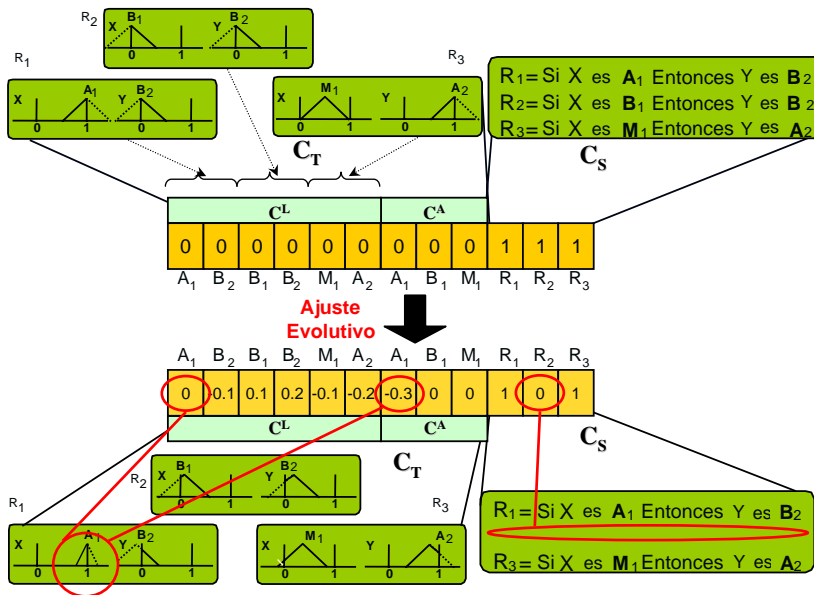
$$\text{Si } c_i = 1 \text{ Entonces } (R_i \in BR) \text{ Sino } (R_i \notin BR) ,$$

donde R_i es la i -ésima regla del conjunto de reglas candidatas y BR es la base de reglas final. La Figura 3.2 muestra gráficamente un ejemplo de correspondencia entre un cromosoma y la BC asociada cuando el ajuste-LA es combinado con la selección de reglas.

- *Población inicial* — La población inicial se obtiene introduciendo un cromosoma con todos sus genes a '0.0' en la parte C_T y todos a '1' en la parte C_S del cromosoma, y generando aleatoriamente el resto de cromosomas dentro de los intervalos $[-0.5, 0.5)$ y $\{0, 1\}$ respectivamente.



a) Enfoque Global



b) Enfoque Local

Figura 3.2: Ejemplo del esquema de codificación considerando el ajuste-LA y la selección de reglas: (a) Enfoque Global; (b) Enfoque Local

- *Operador de Cruce* — El operador de cruce que se utiliza es el operador de cruce basado en entornos presentado en la Sección 3.2.1 en la parte C_T combinado con el operador de cruce HUX [Esh91] en la parte C_S (ver la Figura 3.3). El operador de cruce HUX intercambia exáctamente la mitad de los alelos que son diferentes en los padres (los genes que se van a intercambiar son seleccionados aleatoriamente entre todos los genes que son distintos en los padres). Este operador asegura la máxima distancia entre los descendientes y sus padres (exploración).

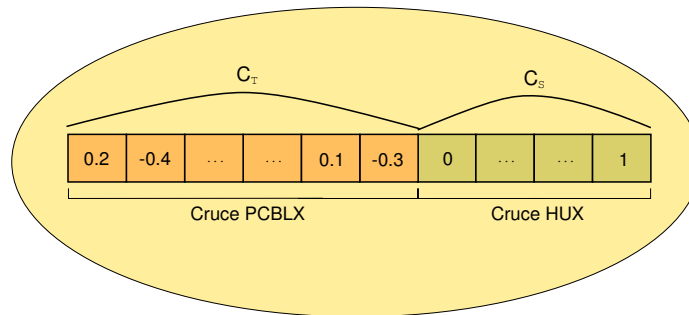


Figura 3.3: Ámbito de aplicación de los operadores

En este caso, la prevención de incesto es aplicada considerando un valor umbral distinto para cada parte, L_T aplicado a la parte C_T (como en la Sección 3.2.1) y L_S aplicado a la parte C_S (el operador de cruce HUX solo es aplicado si la distancia de hamming entre los dos padres, en la parte C_S , dividida por 2 es mayor que L_S). L_S se inicializa como:

$$L_S = \#GenesC_S/4,0,$$

donde $\#GenesC_S$ es el número de genes en C_S . Finalmente, los descendientes son generados de la siguiente forma:

1. Si no se aplica ninguno de los operadores de cruce, ningún descendiente es generado.
2. Si solo se aplica el operador de cruce PBLX se generan dos descendientes, copiando la parte C_S de sus padres respectivamente.
3. Si solo se aplica el operador de cruce HUX se generan dos descendientes, copiando la parte C_T de sus padres respectivamente.

4. Si se aplican los dos operadores de cruce (PBLX y HUX) se generan dos descendientes combinando los dos obtenidos de la parte C_T con los dos obtenidos para la parte C_S .

Si no se introduce un individuo nuevo en la población por la aplicación del operador PBLX, L_T es decrementado como se comenta en la Sección 3.2.1. Si no se introduce un individuo nuevo en la población por la aplicación del operador HUX, L_S se decrementa en 1.

- *Mecanismo de reinicialización* — Para evitar caer en óptimos locales, el operador de reinicialización es aplicado cuando los valores umbral L_T y L_S son menores que 0. En este caso, se copia en todos los cromosomas la parte C_S del mejor cromosoma y los parámetros de la parte C_T son generados sumando a cada gen del mejor cromosoma un valor aleatorio dentro del intervalo $[-0.125, 0.125)$. Una vez fijada la parte C_S (en la primera reinicialización), no tiene sentido seguir utilizando el umbral L_S .

3.2.3. Estudio Experimental

Para evaluar el comportamiento de los cuatro métodos propuestos (ajuste local y global con y sin selección de reglas), consideramos los dos problemas [CHS99] descritos en el Apéndice D: La estimación de la longitud de las líneas de baja tensión y la estimación del costo de mantenimiento de la línea de media tensión.

Como hemos comentado al principio de la sección, el conjunto de reglas difusas inicial se obtendrá a partir de métodos de aprendizaje automático. En ambos problemas, para obtener el conjunto de reglas candidatas se utiliza el algoritmo de Wang y Mendel [WM92]. Para ello, consideraremos particiones difusas triangulares simétricas. Una vez que la BR inicial ha sido generada, se aplicarán los distintos algoritmos de post-procesamiento propuestos. En las siguientes secciones resolveremos los dos problemas y analizaremos los métodos propuestos.

3.2.3.1. Experimentos

Los algoritmos considerados para realizar los experimentos se describen brevemente en la Tabla 3.1. El uso de dos de ellos conectados por el símbolo '+' en la tablas de resultados indica que los dos métodos son aplicados como una combinación. Por ejemplo, GLA+S indica un ajuste-LA global combinado con la

Tabla 3.1: Métodos considerados para la comparación

Ref.	Método	Descripción
[WM92]	WM	Algoritmo Específico de Envoltura de Ejemplos
[KKS00]	S	Algoritmo de Selección de Reglas
[CH97]	T	Algoritmo Clásico de Ajuste de Funciones de Pertenencia
[CCdJH05]	PAL	Algoritmo de Ajuste de Parámetros, Dominios (Factores de Escala) y Modificadores Lingüísticos
[AH04]	GL	Algoritmo de Ajuste Lateral Global de Funciones de Pertenencia
[AH04]	LL	Algoritmo de Ajuste Lateral Local de Funciones de Pertenencia
	GLA	Algoritmo de Ajuste Lateral y de Amplitud Global de Funciones de Pertenencia
	LLA	Algoritmo de Ajuste Lateral y de Amplitud Local de Funciones de Pertenencia

selección de reglas.

Como comentamos anteriormente, utilizaremos el método WM para obtener la BR inicial que va a ser ajustada. Los métodos de ajuste son aplicados una vez que la BR ha sido obtenida. T es un algoritmo clásico para el ajuste de parámetros de las funciones de pertenencia. El método de ajuste PAL ha sido comparado con métodos de ajuste de parámetros, factores de escala y de modificadores lingüísticos y con cualquier combinación de ellos dos a dos, obteniendo mejores resultados en todos los casos [CCdJH05]. Por esta razón, solo se ha considerado el método PAL en este estudio. Y finalmente, los métodos GL y LL realizan un ajuste lateral global y local de las funciones de pertenencia respectivamente.

Las particiones lingüísticas consideradas se componen de *cinco términos lingüísticos* con forma triangular (Figura D.1 del Apéndice D). Se utiliza el *centro de gravedad ponderado por el grado de emparejamiento* como operador de defuzzificación y la *t-norma del mínimo* como operador de implicación y conjunción.

Los experimentos se han realizado considerando un *modelo de validación cruzada de 5 particiones*, es decir, se generan 5 particiones de datos¹ al 20% y se combinan cuatro de ellas (80%) para formar el conjunto de entrenamiento y la restante se utiliza para prueba, obteniéndose 5 particiones distintas. Para cada una de las 5 particiones cada método se ha ejecutado 6 veces, lo que supone 30 ejecuciones de cada método en total. En las tablas de resultados de las siguientes secciones se muestran los resultados medios de las 30 ejecuciones de cada método. Además, se aplica un test *t-student* (con 95% de confianza) para comprobar si las diferencias que hay entre los resultados mostrados en las tablas son significativas.

Para facilitar las comparaciones, se han seleccionado parámetros estándar, en vez de utilizar parámetros muy específicos para cada método. 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). Finalmente, los siguientes valores son los que hemos considerado como parámetros para cada método: 51 individuos, 50.000 evaluaciones, 30 bits por gen para la codificación Gray y $\varphi = 10; 0,2$ y $0,6$ como probabilidad de mutación y de cruce respectivamente, y $0,35$ para el factor a del operador de cruce max-min-aritmético [HLV98b] que utilizan T y PAL.

3.2.3.2. Estimación de la Longitud de las Líneas de Baja Tensión

Los resultados obtenidos por los métodos analizados se muestran en la Tabla 3.2, donde $\#R$ es el número medio de reglas, ECM_{ent} y ECM_{pru} son el error medio obtenido sobre los conjuntos de entrenamiento y prueba respectivamente, σ_{ent} y σ_{pru} son la desviación típica de los resultados de entrenamiento y prueba respectivamente, y *t-test* representa la siguiente información:

★ representa el mejor resultado medio.

+ significa que el mejor resultado tiene significativamente mejor comportamiento que el método que tenga asignado este símbolo.

= denota que los resultados son estadísticamente iguales acorde al test *t-student* con un 95% de confianza.

¹Los conjuntos de datos están disponibles en: <http://decsai.ugr.es/~casillas/fmlib/>.

Tabla 3.2: Resultados obtenidos en el problema de estimación de la longitud de las líneas eléctricas de baja tensión

Método	#R	ECM _{ent}	σ_{ent}	t-test	ECM _{pru}	σ_{pru}	t-test
WM	12,4	234.712	32.073	+	242.147	24.473	+
S	10,0	226.135	19.875	+	241.883	19.410	+
T	12,4	158.662	6.495	+	221.613	29.986	+
PAL	12,4	141.638	4.340	+	189.279	19.523	=
T+S	8,9	156.313	2.967	+	193.477	49.912	=
PAL+S	10,6	145.712	5.444	+	191.922	16.987	=
GL	12,4	166.674	11.480	+	189.216	14.743	=
LL	12,4	139.189	3.155	+	191.604	18.243	=
GL+S	9,0	160.081	7.316	+	189.844	22.448	=
LL+S	10,5	141.446	3.444	+	186.746	15.762	=
GLA	12,4	157.604	9.158	+	185.810	18.812	★
LLA	12,4	133.076	4.330	★	191.945	16.456	=
GLA+S	10,2	155.404	9.264	+	189.472	20.393	=
LLA+S	10,4	134.541	5.752	=	189.057	20.106	=

Si analizamos los resultados presentados en la Tabla 3.2 podemos resaltar los siguientes hechos:

- Considerando estas técnicas, la BR inicial se obtiene a partir de un método de generación (WM) que nos proporciona pocas reglas (12.4 de 25 reglas posibles), lo que nos permite obtener modelos ajustados compactos y precisos.
- En las particiones de prueba muchos de los métodos muestran un error similar, sin embargo el método GLA obtiene los mejores resultados en media. Este comportamiento es debido al ruido y a la fuerte no-linealidad que presenta este problema, lo que hace muy difícil eludir el sobreaprendizaje, especialmente cuando estamos logrando reducir más y más el error en entrenamiento.

- Independientemente, en los conjuntos de entrenamiento, el ajuste-LA local (LLA) obtiene mejores resultados que el resto de aproximaciones. La ventaja real del ajuste-LA local es que el error de prueba no incrementa respecto al resto de aproximaciones, a pesar de los buenos resultados que obtiene en entrenamiento.
- Respecto al enfoque global, solo se obtienen pequeñas mejoras respecto al ajuste lateral global. El número de variables consideradas en el problema hace que el espacio de búsqueda no sea lo suficiente complejo para el enfoque global y, por lo tanto, no hay diferencias significativas en los métodos de ajuste.
- Si nos fijamos en la combinación de los métodos de ajuste y la selección de reglas, GLA+S y LLA+S respectivamente, muestran más o menos la misma precisión que GLA y LLA pero considerando un número menor de reglas, por lo que consiguen alcanzar un mejor equilibrio entre interpretabilidad y precisión. Hay que destacar que la combinación de varias técnicas (especialmente cuando actúan sobre partes distintas del SBRD) incrementa la complejidad del espacio de búsqueda. Sin embargo, en el caso del ajuste-LA global la precisión mejora cuanto es combinado con la selección de reglas, lo que podría indicar un mejor comportamiento del enfoque cuando el espacio de búsqueda crece.

La Figura 3.4 presenta las particiones lingüísticas difusas y la BR obtenida por el método GLA+S en una de sus 30 ejecuciones. Por otro lado, la Figura 3.5 muestra la BR obtenida en una de las 30 ejecuciones del método LLA+S, donde podemos ver como es el ajuste local de cada etiqueta en cada una de las reglas. Las funciones de pertenencia iniciales consideradas por este método son las mismas que las mostradas en color gris en la Figura 3.4.

Las particiones lingüísticas consideradas se componen de *cinco términos lingüísticos* con forma triangular. Los términos lingüísticos se etiquetan desde l_1 hasta l_5 y se corresponden con las funciones de pertenencia de menor a mayor valor modal (Figura D.1 del Apéndice D).

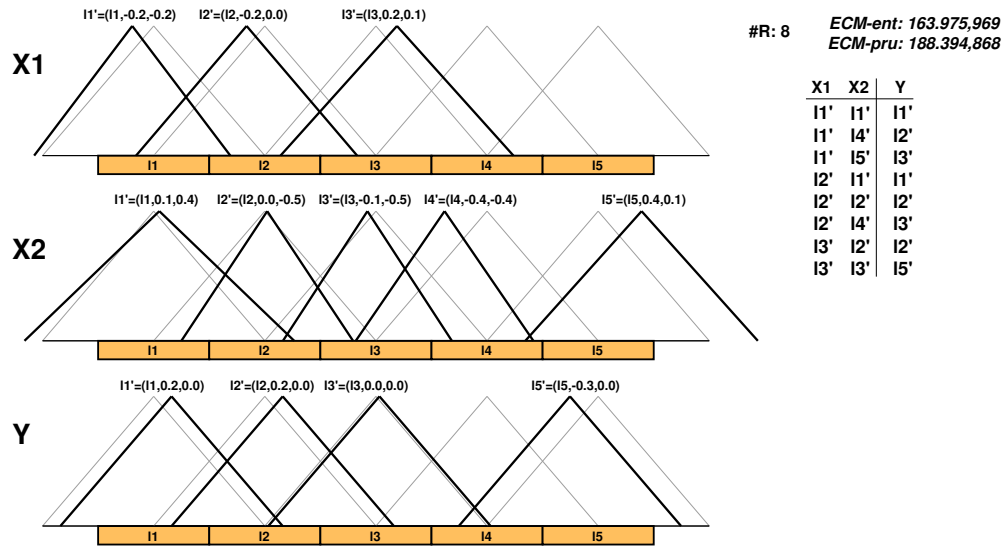


Figura 3.4: BD inicial, BD ajustada y BR de un modelo obtenido con el método GLA+S en el problema de estimación de la longitud de las líneas de baja tensión (las etiquetas no utilizadas han sido eliminadas en la figura)

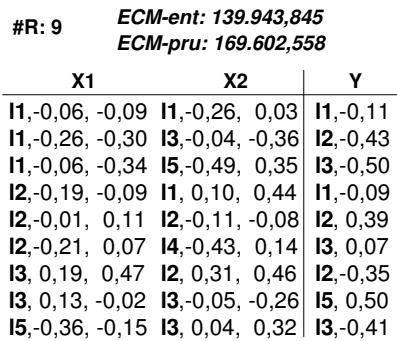


Figura 3.5: BR y desplazamientos laterales de un modelo obtenido con el método LLA+S en el problema de estimación de la longitud de las líneas de baja tensión

Para facilitar la representación gráfica, en todas las figuras las etiquetas son nombradas desde 'I1' hasta 'ILⁱ' (donde Lⁱ es el número de etiquetas para la i-ésima variable). Sin embargo, estas etiquetas deberían tener asociado un sig-

nificado lingüístico determinado por el experto. En este sentido, si la etiqueta '11' de la variable 'X1' representa 'BAJO', '11+0'11' podría ser interpretada como 'un poco más pequeño que bajo' (basado en la opinión del experto) o, como en el caso del ajuste clásico, esta podría ser interpretada manteniendo el significado original de la etiqueta. Este es el caso de la Figura 3.4, donde prácticamente todas las etiquetas nuevas podrían mantener su significado original o ser fácilmente renombradas.

Estas figuras muestran como variaciones pequeñas en las funciones de pertenencia provocan importantes mejoras en el comportamiento de los SBRDs obtenidos. Además, la dificultad de conseguir el equilibrio entre precisión y complejidad se pone de manifiesto si observamos los resultados obtenidos, la BR global y la BR local (Figuras 3.4 y 3.5). La precisión puede ser mejorada cuando consideramos un enfoque local pero la pérdida de interpretabilidad es algo mayor.

3.2.3.3. Estimación del Costo de Mantenimiento de la Línea de Media Tensión

En La Tabla 3.3 se muestran los resultados obtenidos por los métodos analizados (este tipo de tabla fue descrita en la subsección anterior).

Analizando los resultados presentados en la Tabla 3.3 podemos resaltar:

- Las BRs iniciales están también compuestas por pocas reglas, 65 reglas de 625 posibles, que es razonable para este tipo de problema.
- Los métodos de ajuste-LA muestran una importante reducción del ECM respecto a los métodos clásicos (especialmente el método LLA) y una mejora razonable respecto al ajuste lateral.
- Los mejores resultados son obtenidos por el enfoque local, presentando una buena relación entre la complejidad del espacio de búsqueda y los resultados obtenidos, consiguiendo un buen equilibrio entre precisión e interpretabilidad local. Además, teniendo en cuenta que las variaciones laterales y de amplitud son relativas a las etiquetas globales originales, se podría hacer una interpretabilidad global desde este punto de vista.
- Los métodos GLA+S y LLA+S respectivamente muestran más o menos la misma precisión que los métodos GLA y LLA considerando un número

Tabla 3.3: Resultados obtenidos en el problema de estimación del costo de mantenimiento de la línea de media tensión

Método	#R	ECM _{ent}	σ_{ent}	t-test	ECM _{pru}	σ_{pru}	t-test
WM	65,0	57.605	2.841	+	57.934	4.733	+
S	40,8	41.086	1.322	+	59.942	4.931	+
T	65,0	18.602	1.211	+	22.666	3.386	+
PAL	65,0	10.545	279	+	13.973	1.688	+
T+S	41,9	14.987	391	+	18.973	3.772	+
PAL+S	57,4	12.851	362	+	16.854	1.463	+
GL	65,0	23.064	1.479	+	25.654	2.611	+
LL	65,0	3.664	390	+	5.858	1.798	+
GL+S	49,1	18.801	2.669	+	22.586	3.550	+
LL+S	58,0	3.821	385	+	6.339	2.164	+
GLA	65,0	17.950	1.889	+	21.212	2.686	+
LLA	65,0	2.747	282	*	4.540	788	*
GLA+S	49,4	17.538	2.391	+	21.491	4.168	+
LLA+S	47,5	3.404	433	+	5.633	1.452	+

reducido de reglas. No obstante, el método LLAT+S elimina 10 reglas más que el método LL+S.

Las Figuras 3.6 y 3.7 muestran la BC de una de las 30 BCs obtenidas por los métodos GLA y GLA-S respectivamente. Analizando los dos modelos lingüísticos podemos ver que en la mayoría de los casos, las funciones de pertenencia ajustadas son muy similares a las originales, preservando probablemente su significado original desde el punto de vista del experto.

En la Figura 3.6, solo la segunda variable presenta dos etiquetas cuyo significado podría cambiar. Esto no ocurre en el caso del método GLA-S, Figura 3.7, donde solo la primera variable presenta algunas funciones de pertenencia con variaciones significativas respecto a las originales. Sin embargo, las etiquetas correspondientes podrían mantener su significado original. Esto puede ser debido a

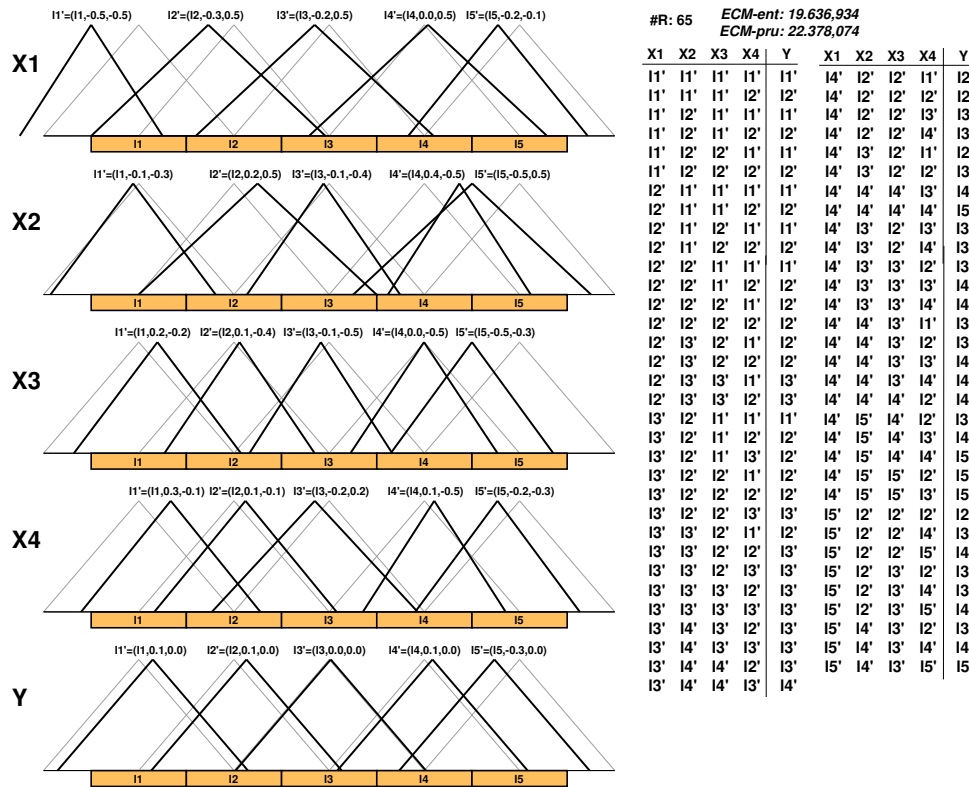


Figura 3.6: BD inicial, BD ajustada y BR de un modelo obtenido con el método GLA en el problema de estimación del costo de mantenimiento de la línea de media tensión

la existencia de reglas malas en la BR, lo que puede provocar fuertes variaciones en las etiquetas con el objetivo de mejorar el comportamiento de estas reglas. Eliminando este tipo de reglas se resuelve este problema.

3.2.3.4. Análisis sobre el Uso del Nuevo Sistema de Inferencia

En esta subsección presentamos un estudio comparativo de los métodos propuestos con y sin considerar el nuevo sistema de inferencia para trabajar con datos no cubiertos. Este estudio ha sido realizado solo sobre el primer proble-

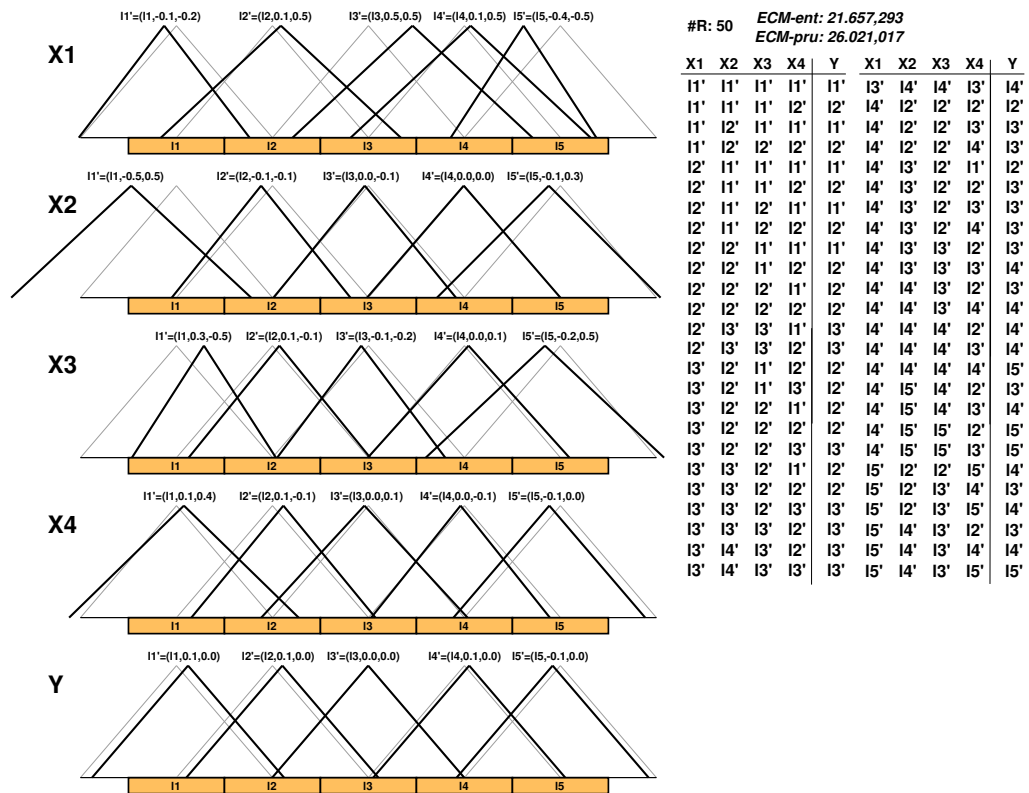


Figura 3.7: BD inicial, BD ajustada y BR de un modelo obtenido con el método GLA+S en el problema de estimación del costo de mantenimiento de la línea de media tensión

ma presentado en esta sección (estimación de la longitud de las líneas de bajo voltaje). Este problema presenta una fuerte no-linealidad, 2 puntos de ruido y un número bajo de ejemplos, lo que fuerza que en algunas ocasiones los modelos ajustados no cubran alguno de los datos de prueba (se da un caso en una de las cinco particiones de datos para prueba) o incluso algún dato de entrenamiento (se da un caso en una de las cinco particiones de datos para entrenamiento). Los dos ejemplos no cubiertos son los dos puntos de ruido presentados en la Tabla 3.4.

En el segundo problema, los ejemplos son suficientes, bien distribuidos y no presentan puntos de ruido. Además, en este problema todos los ejemplos (en entrenamiento y prueba) son finalmente cubiertos por los modelos obtenidos. Por

Tabla 3.4: Datos no cubiertos

X_1	X_2	Y
320	880	3.090
29	1.200	3.985

esta razón no tiene sentido analizar este problema. En la Tabla 3.5 podemos ver los resultados obtenidos.

Clásicamente, cuando un patrón de entrada no es cubierto por una regla el sistema da una salida por defecto. Las dos más utilizadas son: considerar cero como salida por defecto (típico en problemas de control) o considerar la media del dominio de salida como salida por defecto. Ambos son considerados en nuestro estudio, mostrando los resultados obtenidos en la tabla. En este problema, el valor medio del dominio de salida es exactamente 3.797,5, que es muy cercano a la salida deseada para los puntos no cubiertos presentados en la Tabla 3.4. Esto proporciona una mayor capacidad de generalización a los métodos cuando se utiliza como salida por defecto la media del dominio de salida. Sin embargo, esta postura no es razonable para presentar nuestros resultados, por eso desarrollamos el nuevo sistema de inferencia presentado en la Sección 3.1.1 que nos permite tener mejor en cuenta los datos no cubiertos. Así, los métodos propuestos pueden también trabajar con otros problemas en los que la media del dominio de salida no coincida con la salida deseada para los datos no cubiertos.

Como podemos ver en los resultados de la Tabla 3.5, la salida por defecto considerada tiene una gran influencia en la capacidad de generalización de los modelos obtenidos. El nuevo sistema de inferencia parece ser una forma efectiva para determinar una buena salida basada en la información contenida en las reglas ajustadas finales. Desde este punto de vista, debemos apuntar que todos los métodos considerados en las comparaciones de las Secciones 3.2.3.2 y 3.2.3.3 consideran la media del dominio de salida como salida por defecto (lo que les ayuda a obtener un mejor error en prueba) puesto que prácticamente todos ellos fueron propuestos siguiendo este enfoque.

Tabla 3.5: Resultados obtenidos en el problema de estimación de la longitud de las líneas eléctricas de baja tensión

Método	#R	ECM _{ent}	σ_{ent}	t-test	ECM _{pru}	σ_{pru}	t-test
Cero como Salida por Defecto (0)							
GLA	12,4	158.611	9.091	+	217.953	38.780	+
LLA	12,4	132.815	5.715	*	201.400	25.034	+
GLA+S	11,0	157.068	8.253	+	228.922	44.799	+
LLA+S	11,1	136.493	5.133	=	212.655	31.089	+
La media del Dominio de Salida como Salida por Defecto (3.797,5)							
GLA	12,4	158.284	9.540	+	187.879	18.408	=
LLA	12,4	134.003	3.815	=	187.568	15.828	=
GLA+S	9,9	155.927	6.328	+	187.820	20.074	=
LLA+S	9,8	139.047	6.288	+	190.678	20.892	=
Con el Nuevo Sistema de Inferencia							
GLA	12,4	157.604	9.158	+	185.810	18.812	*
LLA	12,4	133.076	4.330	=	191.945	16.456	=
GLA+S	10,2	155.404	9.264	+	189.472	20.393	=
LLA+S	10,4	134.541	5.752	=	189.057	20.106	=

3.3. Enfoque Aplicado Sobre Reglas Proporcionadas por un Experto

En la sección anterior, se ha visto como el ajuste-LA y el ajuste-LA junto con la selección de reglas pueden aplicarse sobre BRs previamente obtenidas para mejorar el rendimiento del modelo o controlador lingüístico. Así, el controlador podría obtenerse a partir de un experto y después ajustarse de forma similar a las BRs obtenidas a partir de ejemplos. En esta sección veremos el uso del ajuste-LA y del ajuste-LA junto con la selección de reglas para el desarrollo de controladores difusos dedicados al control de sistemas de climatización HVAC donde el principal

objetivo es reducir el consumo energético manteniendo el confort deseado. Debido a que el tiempo de evaluación de una solución para este tipo de problemas es elevado, modificaremos el modelo evolutivo del algoritmo básico para el ajuste-LA y para el ajuste-LA junto con la selección de reglas, propuestos en la Sección 3.2.1 y Sección 3.2.2.2 respectivamente, con el objetivo de hacerlos más eficientes.

Esta sección se organiza como sigue. En la Sección 3.3, se presentan los aspectos básicos del problema de control de los sistemas HVAC (para una descripción más detallada ver la Sección D.2.3 del Apéndice D). En las Secciones 3.3.2 y 3.3.2.3 se describe en detalle, respectivamente, la adaptación del algoritmo para el ajuste-LA y el ajuste-LA con selección de reglas para aplicarlos al problema de control del HVAC. Finalmente, en la Sección 3.3.3 se muestran los resultados obtenidos.

3.3.1. El Problema de Control de Sistemas HVAC

En los países de la UE, el consumo de energía primaria en grandes edificios representa aproximadamente el 40 % del consumo total de energía, y dependiendo del país, más de la mitad de esta energía se destina a climatizar el interior de dichos edificios. Desde un punto de vista tecnológico, se estima que el empleo de tecnologías tales como los Sistemas de Administración de Energía en Edificios — *Building Energy Management Systems* (BEMSs)— puede ahorrar hasta un 20 % del consumo de energía del sector inmobiliario, es decir, un 8 % del consumo total de la CEE [DPE96]. Con dicha intención, los BEMSs se aplican solamente al control de sistemas activos, es decir, sistemas de climatización HVAC.

Un sistema HVAC está compuesto por todas las partes electromecánicas del edificio usadas para acondicionar el aire interior del mismo. El sistema HVAC es necesario para proporcionar a los trabajadores un entorno de trabajo cómodo y productivo que satisfaga sus necesidades fisiológicas. En la Figura 3.8, se presenta la estructura de un sistema HVAC típico para oficinas.

Los sistemas HVAC son equipos usualmente desarrollados para mantener un confort satisfactorio en grandes edificios. Tanto el consumo de energía como los aspectos relativos al confort en el interior de edificios en lo que se refiere a la ventilación y al aire acondicionado, son altamente dependientes del diseño, rendimiento y control de sus equipos y sistemas HVAC. Por lo tanto, el uso de estrategias de control automático adecuadas para el control de sistemas HVAC, como los *controladores difusos* [DHR93, Mam74, MA75], podría resul-

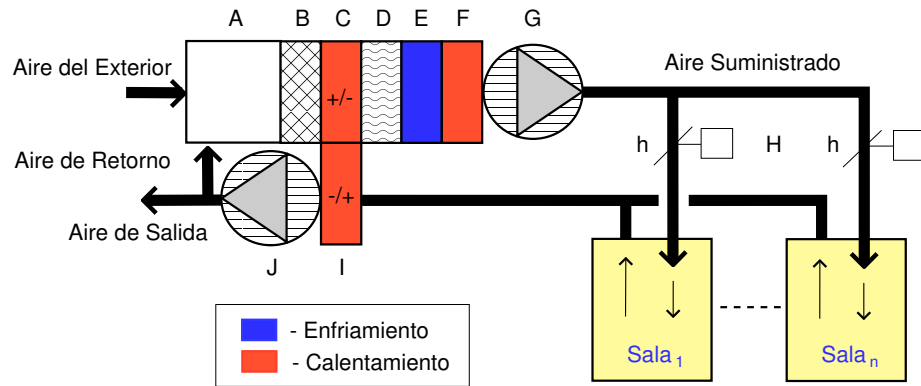


Figura 3.8: Estructura genérica de un sistema HVAC para un edificio de oficinas

tar en importantes ahorros de energía cuando se compara con el control manual [ACG99, ACC⁺01, ABC⁺03b, AHK95, HN94].

En el caso específico de los sistemas de climatización HVAC, muchas de las técnicas de la inteligencia artificial podrían ser aplicadas para mejorar el funcionamiento de los sistemas HVAC [AHK95, FCS04, Glo91, HN94]. Sin embargo, la mayoría de los trabajos aplican los controladores difusos para resolver problemas simples, como puede ser el de regulación térmica, manteniendo la temperatura en un punto concreto. Por otro lado, la construcción de la BC se basa en la información recopilada a partir de la experiencia de los operadores, lo que no siempre consigue resultados satisfactorios con un conjunto inicial de reglas definido a partir de expertos [HN94]. Además, en nuestro caso se deben optimizar varios criterios para conseguir un buen funcionamiento del sistema HVAC. Usualmente, el objetivo principal es de reducir el consumo de energía manteniendo el nivel de confort deseado.

En nuestro caso, deberemos optimizar cinco criterios para mejorar el comportamiento del controlador difuso inicial obtenido a partir de la experiencia humana (involucrando 17 variables) mediante la aplicación del ajuste-LA y el ajuste-LA con selección de reglas. Para ello, consideraremos los modelos validados y calibrados de un edificio de pruebas real. Tanto el controlador difuso inicial como el modelo de simulación fueron desarrollados dentro del marco de trabajo del programa *JOULE-THERMIE* bajo el proyecto GENESYS². A partir de ahora,

²GENESYS: Fuzzy controllers and smart tuning techniques for energy efficiency and overall

este lugar será llamado módulo de pruebas de GENESYS (ver la Sección D.2.3.3 del Apéndice D).

En las subsecciones siguientes presentamos los cinco objetivos y la función de adaptación que será optimizada junto con las variables y la arquitectura del controlador difuso inicial.

3.3.1.1. Objetivos y Función de Adaptación

Como se ha dicho, **el principal objetivo es mejorar el rendimiento energético manteniendo el nivel de confort deseado**. Para ello, el objetivo global es **minimizar** los siguientes cinco criterios:

- O₁** Límite de confort térmico superior: *si* $PMV > 0'5$, $O_1 = O_1 + (PMV - 0'5)$, donde PMV (del inglés *Predicted Mean Vote*) es el índice global de confort térmico ISO 7730 ³, que incorpora la humedad relativa y la temperatura media irradiada.
- O₂** Límite de confort térmico inferior: *si* $PMV < -0'5$, $O_2 = O_2 + (-PMV - 0'5)$.
- O₃** Requisito de calidad del aire: *si* $CO_2 \text{ conc.} > 800ppm$, $O_3 = O_3 + (CO_2 - 800)$.
- O₄** Consumo de energía: $O_4 = O_4 + \text{potencia en el tiempo } t$.
- O₅** Estabilidad del sistema: $O_5 = O_5 + \text{cambios del sistema desde el tiempo } t \text{ al } (t-1)$, donde 'cambio del sistema' representa un cambio de funcionamiento, por ejemplo, un cambio de la velocidad del ventilador o de la posición de una válvula.

En nuestro caso, los cinco criterios se combinan en una función objetivo global por medio de un vector de pesos. Esta técnica (suma ponderada de objetivos) tiene mucha sensibilidad y dependencia respecto a los pesos. Sin embargo, cuando se dispone de pesos fiables, este enfoque reduce el espacio de búsqueda proporcionando la dirección de búsqueda más adecuada y, por lo tanto, es recomendable.

performance of HVAC systems in buildings, European Commission, Directorate-General XII for Energy (contract JOE-CT98-0090).

³<http://www.iso.org/iso/en/ISOOnline.frontpage>

En nuestro caso, los expertos proporcionaron pesos fiables para la función de evaluación del módulo de pruebas de GENESYS: $w_1^O = 0'0083022$, $w_2^O = 0'0083022$, $w_3^O = 0'00000456662$, $w_4^O = 0'0000017832$ y $w_5^O = 0'000761667$. Finalmente, la función objetivo a minimizar se calcula como:

$$F = \sum_{i=1}^5 w_i^O \cdot O_i .$$

3.3.1.2. Variables y Arquitectura del Controlador Difuso para el sistema HVAC

El experto en BEMS propuso una arquitectura jerárquica de controlador difuso que consideraba el PMV, la concentración de CO_2 , el estado previo del sistema HVAC y la temperatura exterior. Dicha arquitectura, las variables y la BR inicial se muestran en la Figura 3.9.

La BD está compuesta por particiones difusas simétricas con funciones de pertenencia triangulares etiquetadas desde $L1$ a Ll_i , siendo l_i el número de funciones de pertenencia de la i -ésima variable. La Figura 3.10 representa dicha BD. Véase que la Figura 3.9 representa las tablas de decisión de cada módulo del controlador jerárquico en términos de dichas etiquetas. cada celda de la tabla representa un subespacio/regla difuso/a y contiene el consecuente(s) de salida asociados, es decir, la etiqueta(s) correspondientes. Las variables de salida se indican en la esquina superior derecha para cada módulo. Tanto la BR inicial como la BD fueron suministradas por el experto en BEMS.

3.3.2. Aplicación del Ajuste Lateral y de Amplitud al Controlador Difuso del Sistema HVAC

Como hemos visto a lo largo de esta memoria, la aplicación del ajuste-LA podría mejorar el rendimiento de las BCs obtenidas a partir de métodos simples de aprendizaje de reglas e incluso a partir de expertos, sin perder en exceso su interpretabilidad. De esta manera, una vez obtenido el controlador difuso inicial (a partir de expertos), aplicaremos una adaptación del Algoritmo Evolutivo propuesto en la Sección 3.2.1 para el ajuste-LA. Centrándonos en el problema específico que nos atañe, en el que el tiempo de evaluación de una solución considerando el modelo del sistema es elevado, necesitaremos de una modificación

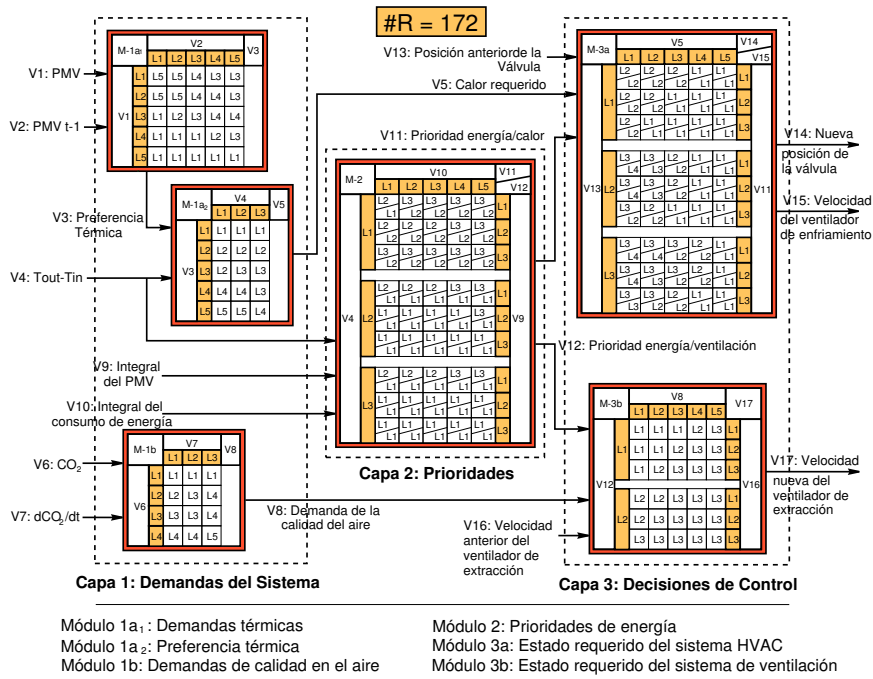


Figura 3.9: BR inicial y estructura del controlador difuso de verano de GENESYS

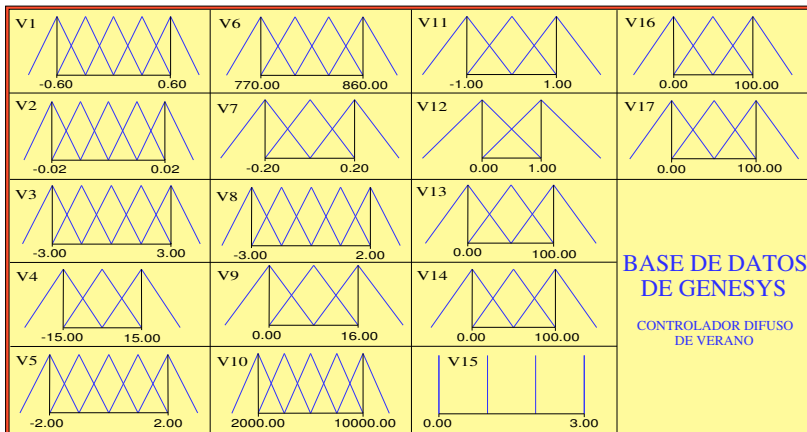


Figura 3.10: BD del controlador difuso de verano de GENESYS

del algoritmo básico para hacerlo más eficiente.

Teniendo en cuenta la existencia de pesos fiables para el cálculo de la función de adaptación y con la intención de sacar provecho de los mismos (véase la Sección 3.3.1.1), se propone un AG estacionario con una función de adaptación basada en la suma ponderada de los objetivos. En las siguientes secciones, se introduce brevemente el enfoque estacionario para, posteriormente, presentar la adaptación del proceso genético para el ajuste-LA.

3.3.2.1. Modelo Evolutivo: Enfoque Estacionario

El enfoque estacionario [WK88] consiste en seleccionar un número pequeño de individuos de la población, habitualmente dos, y combinarlos para obtener una batería de hijos menor que la del enfoque clásico generacional. Los nuevos individuos reemplazarán en la población a los peores de la misma siempre que estén mejor adaptados que ellos. De esta manera, las mejores soluciones son utilizadas pronto en el proceso genético y, por lo tanto, se acelera la convergencia y disminuye el número de evaluaciones. En nuestro caso será crucial, ya que los modelos tardan varios minutos en ser evaluados.

3.3.2.2. Algoritmo Genético para el Ajuste Lateral y de Amplitud

El AG propuesto está basado en el enfoque estacionario [WK88] y emplea una función de adaptación que considera la suma ponderada de objetivos. Sin embargo, para hacer el método más robusto e independiente de la selección de los pesos para la función de adaptación, consideraremos metas difusas que adaptarán la dirección de búsqueda del algoritmo. Por otro lado, el esquema de selección está basado en el muestreo universal estocástico de Baker [Bak87], seleccionando aleatoriamente dos individuos de entre todos los que se obtendrían por dicho método (así, los mejores individuos tienen mayor probabilidad de ser seleccionados). La Figura 3.11 presenta el organigrama del método propuesto, mientras que sus principales componentes se introducen a continuación.

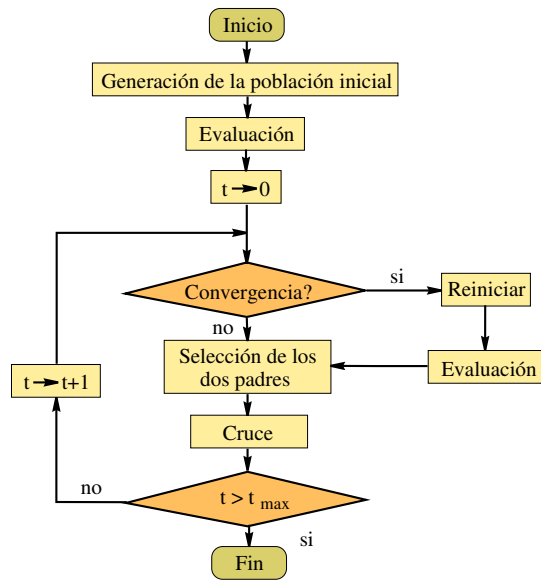


Figura 3.11: Organigrama del algoritmo genético propuesto

3.3.2.2.1. Esquema de Codificación y Población Inicial

En este caso se utilizará el mismo esquema de codificación presentado en la Sección 3.2.1.2 para el algoritmo de ajuste-LA, considerando los dos enfoques, global o local. Supongamos el siguiente número de etiquetas por variable: (m^1, \dots, m^n) , donde n es el número de variables del sistema ($n - 1$ variables de entrada y 1 variable de salida), y que N es el número de reglas inicial:

- Enfoque Global:

$$C_T = (C^L + C^A),$$

$$C^L = (c_{11}^L, \dots, c_{1m^1}^L, \dots, c_{n1}^L, \dots, c_{nm^n}^L),$$

$$C^A = (c_{11}^A, \dots, c_{1m^1}^A, \dots, c_{(n-1)1}^A, \dots, c_{(n-1)m^n}^A).$$

- Enfoque Local:

$$C_T = (C^L + C^A),$$

$$C^L = (c_{11}^L, \dots, c_{1n}^L, \dots, c_{N1}^L, \dots, c_{Nn}^L),$$

$$C^A = (c_{11}^A, \dots, c_{1n-1}^A, \dots, c_{N1}^A, \dots, c_{Nn-1}^A).$$

Al igual que con el algoritmo original, la población inicial se obtiene incluyendo el controlador inicial (todos los genes a 0) y generando aleatoriamente el resto de la población dentro del intervalo $[-0.5, 0.5)$.

3.3.2.2.2. Operadores Genéticos

El único operador genético considerado es el cruce. El operador de mutación no es considerado con el objetivo de acelerar la convergencia del algoritmo.

El operador de cruce que se considera es el cruce BLX- α [ES93] y un híbrido entre un BLX- α y el cruce aritmético [HLV97]. De esta manera, si se cruzan dos padres, $C_T^v = (c_{T1}^v, \dots, c_{Tk}^v, \dots, c_{Tm}^v)$ y $C_T^w = (c_{T1}^w, \dots, c_{Tk}^w, \dots, c_{Tm}^w)$ se consideran dos operadores:

1. Se genera un descendiente (cruce BLX- α , siendo α un parámetro constante elegido por el diseñador del AG) $C_T^h = (c_{T1}^h, \dots, c_{Tk}^h, \dots, c_{Tm}^h)$ donde c_{Tk}^h es generado aleatoriamente dentro del intervalo $[I_{L_k}, I_{R_k}] = [c_{min} - I \cdot \alpha, c_{max} + I \cdot \alpha]$, $c_{min} = \min(c_{Tk}^v, c_{Tk}^w)$, $c_{max} = \max(c_{Tk}^v, c_{Tk}^w)$ y $I = c_{max} - c_{min}$.
2. A continuación se aplica el cruce aritmético sobre el intervalo considerado por el cruce BLX- α , $[I_{L_k}, I_{R_k}]$, generando el individuo C_T^h :

$$C_T^h \text{ con } c_{Tk}^h = aI_{L_k} + (1 - a)I_{R_k},$$

donde $a \in [0, 1]$ es un parámetro aleatorio generado cada vez que este operador es aplicado. De esta manera, se puede realizar una misma adaptación relativa en cada gen, que es una característica buena.

Tras la aplicación de estos operadores se generan dos descendientes, que serán los candidatos a sustituir a los dos peores individuos de la población.

3.3.2.2.3. Evaluación de los Cromosomas

La función de evaluación (véase la Sección 3.3.1.1) ha sido modificada para considerar el uso de metas difusas que adaptarán la dirección de búsqueda del algoritmo. Para ello, se incluye una función que modifica el valor de adaptación para cada objetivo individual, $\delta_i(x)$, y que se utiliza por los dos motivos siguientes:

- para disminuir la importancia de cada objetivo cuanto más cerca esté de satisfacer su meta (tomando valores cercanos a 0), o
- para penalizar cada objetivo cuando su valor empeora respecto al controlador inicial considerado (tomando valores superiores a 1).

En la función $\delta_i(x)$, se ha incluido un factor de penalización, p_i , que permite al usuario dar una cierta prioridad a los objetivos. Dicho factor de penalización es un número real desde 0,7 hasta prácticamente 1 para cada objetivo O_i , aunque el usuario especifica esta penalización desde 0 hasta 1 (menor y mayor prioridad, respectivamente), lo que es más intuitivo. De esta manera, la función de adaptación global finalmente considerada es:

$$F' = \sum_{i=1}^5 w_i^O \cdot \delta_i(O_i) \cdot O_i \text{ ,}$$

donde O_i representa cada criterio (objetivo) considerado y w_i^O es el peso correspondiente.

Dependiendo de los valores de las metas, g_i , y del valor de la solución inicial para cada objetivo, i_i , pueden presentarse dos situaciones diferentes en la evaluación del individuo correspondiente. Por ello, se aplicará una función δ diferente en cada caso:

- El primer caso ocurre cuando el valor de g_i es menor que el valor de i_i , presentando el siguiente comportamiento (Figura 3.12):

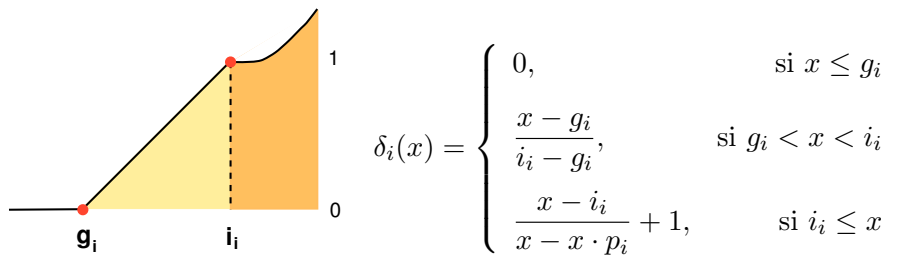


Figura 3.12: $\delta_i(x)$ cuando $g_i \leq i_i$

En este caso, el objetivo se anula si se consigue la meta y se penaliza si se empeora el valor inicial.

- El segundo caso ocurre cuando el valor inicial i_i es menor que la meta g_i (Figura 3.13):

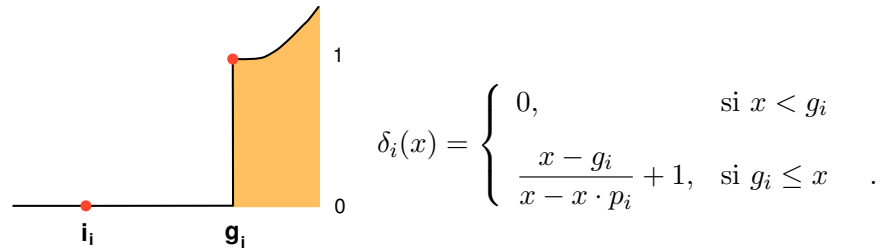


Figura 3.13: $\delta_i(x)$ cuando $g_i > i_i$

Ahora, el valor inicial puede empeorarse mientras que la meta se cumpla, y se penaliza en caso contrario.

La función de penalización permite al proceso de búsqueda empeorar ligeramente una meta específica, mejorando otros objetivos, para posteriormente conseguir dicha meta de nuevo. De este modo, la dirección de la búsqueda en el espacio de las soluciones se adapta dinámicamente.

3.3.2.3. Algoritmo Genético para el Ajuste Lateral y de Amplitud junto con la Selección de Reglas

Como hemos visto a lo largo de esta memoria, la aplicación del ajuste-LA junto con la selección de reglas mejora el rendimiento de las BCs obtenidas a partir de métodos simples de aprendizaje de reglas. En este caso, también aplicaremos la combinación del del ajuste-LA con la selección de reglas al problema del HVAC.

Para realizar el ajuste-LA junto con la selección de reglas hemos modificado el esquema de codificación y los operadores genéticos respecto al algoritmo presentado en la Sección 3.3.2.2. En las siguientes subsecciones describimos en detalle en que consisten estas modificaciones.

3.3.2.3.1. Esquema de Codificación y Población Inicial

Para combinar el ajuste-LA y la selección de reglas hemos usado un esquema de codificación doble, *selección de reglas* (C_S) y *ajuste-LA* (C_T):

$$C^p = C_S^p C_T^p .$$

En este caso, el enfoque previo (parte C_T) se combina con la selección de reglas mediante la unión del vector binario C_S que determina qué regla está seleccionada y cual no. De esta manera, para la parte C_S , el esquema de codificación genera cadenas binarias de longitud m (donde m es el número de reglas difusas existentes en el SBRD obtenido a partir de conocimiento experto). Dependiendo de si una regla está o no seleccionada, se le asignará el alelo '1' ó '0' respectivamente al gen correspondiente. Así, la parte correspondiente a C_S^p para el cromosoma p -ésimo es,

$$C_S^p = (c_{S1}^p, \dots, c_{Sm}^p) \mid c_{Si}^p \in \{0, 1\} .$$

Para hacer uso de la información disponible, el SBRD inicial obtenido a partir de conocimiento experto es incluido en la población como una solución inicial. Para ello, se introduce en la población un individuo con todos los genes de la parte C_S a '1' (todas las reglas seleccionadas) y con todos los genes de la parte C_T a '0,0' (sin desplazamientos ni variaciones en la amplitud). El resto de los individuos de la población son generados aleatoriamente dentro de los intervalos correspondientes.

3.3.2.3.2. Operadores Genéticos

El operador de cruce utilizado es el operador de cruce presentado en la Sección en la parte C_T combinado con el operador de cruce estándar en dos puntos en la parte C_S (ver la Sección B.4 del Apéndice B).

Finalmente, se generan dos descendientes para la parte C_S y otros dos para la parte C_T del cromosoma, obteniendo 4 descendientes como resultado de combinar los dos descendientes de cada parte. A continuación, aplicamos el operador de mutación clásicamente empleado en los AG (ver la Sección B.5 del Apéndice B) sobre los genes de la parte C_S , mientras que la parte C_T de los cromosomas permanece igual. El objetivo de aplicar el operador de mutación solo sobre la parte C_S de los cromosomas es mejorar la convergencia del algoritmo, debido a

que la parte C_S de los cromosomas puede presentar una convergencia prematura en la búsqueda. Así, una vez que el operador de mutación ha sido aplicado sobre cada uno de los 4 descendientes generados por el operador de cruce, seleccionamos los dos mejores individuos para considerarlos como los dos descendientes finales.

3.3.2.3.3. Mecanismo de Reinicialización

Finalmente, para no caer en óptimos locales, el algoritmo usa un mecanismo de reinicialización [Esh91]. Así, cuando la población de soluciones converge a soluciones muy similares (si la diferencia entre el valor de la función de adaptación para el mejor y el peor individuo es menor de un 1% del valor de la función de adaptación para la solución inicial), se reinicia la búsqueda manteniendo el mejor individuo en la población y generando el resto aleatoriamente dentro de los intervalos de variación correspondientes. Este esquema permite al algoritmo una mejor exploración del espacio de búsqueda, escapando de óptimos locales.

3.3.3. Experimentos y Análisis de Resultados

Para evaluar la bondad de la técnica propuesta, se han llevado a cabo diferentes experimentos considerando el módulo de pruebas de GENESYS. Las principales características, objetivos y el controlador difuso inicial para GENESYS se presentan en la Sección 3.3.1. En esta sección, se describen los experimentos realizados sobre el controlador difuso inicial aplicando en los algoritmos propuestos y usando el simulador del módulo de pruebas. Para ello se han realizado dos estudios diferentes:

1. *Un estudio sobre los métodos basados en el ajuste-LA.* En este caso, los resultados obtenidos por los métodos basados en el ajuste-LA se analizan por separado. Los métodos son:
 - a) Ajuste-LA (global y local).
 - b) Ajuste-LA + Selección de Reglas (global y local).
2. *Un análisis considerando también los resultados obtenidos por otros métodos.* Se realiza una comparativa de los resultados medios obtenidos por los métodos propuestos en este capítulo con un método de ajuste clásico y los métodos para el ajuste lateral.

Como ya hemos comentado, para facilitar el cálculo de la función de adaptación de las técnicas propuestas, los expertos obtuvieron modelos precisos del edificio controlado, además de aportar el controlador difuso inicial. Para las evaluaciones se realizaron simulaciones de 10 días con las condiciones climáticas deseadas (con un tiempo de cómputo aproximado de 1 ó 2 minutos).

Los controladores obtenidos se compararán tanto con el controlador difuso inicial como con un controlador clásico *On-Off*. *Las metas y mejoras se calcularán respecto a éste último*. La intención de los expertos fue obtener un 10 % de ahorro de energía, O_4 , junto con una mejora global en el comportamiento del sistema comparado al *On-Off*. Las medidas del confort, O_1 y O_2 , podían incrementarse, pero no deben tomar valores mayores a 1,0.

En la Tabla 3.6 (donde % representa el tanto por ciento de mejora respecto al controlador On-Off para cada criterio y #R el número de reglas difusas), se muestran los resultados obtenidos con el controlador On-Off y el controlador difuso inicial junto con los parámetros considerados para el cómputo de la función de adaptación en el AG (F'), las metas difusas y los factores de penalización (los pesos utilizados para la suma ponderada de los objetivos ya fueron introducidos en la Sección 3.3.1.1). Véase que, para intentar obtener aún mejores resultados, las metas impuestas al algoritmo son mayores que las requeridas inicialmente por los expertos. En la tabla no se han representado porcentajes de mejora para $O_1 \dots O_3$, puesto que estos objetivos se satisfacen siempre y el controlador *On-Off* presenta valor cero en ellos.

Tabla 3.6: Controlador inicial y parámetros usados en la función de adaptación (F')

MODELO	#R	F. Adapt.		PMV			CO ₂		Energía		Estabilidad	
		F	%	O_1	O_2	O_3	O_4	%	O_5	%		
ON-OFF	—	6,58	—	0,0	0	0	3.206.400	—	1.136	—		
C. DIFUSO	172	6,32	4	0,0	0	0	2.901.686	9,50	1.505	-32,48		
Metas (g_i)	—	—	—	1,0	1	7	1.500.000	—	650	—		
Penaliz. (p_i)	—	—	—	1	1	1	0,9	—	0,97	—		

Finalmente, los valores considerados para los parámetros utilizados en los experimentos son: 31 individuos, probabilidad de mutación 0,2; 0,3 como factor α del

operador BLX y 0,35 para el factor a del operador de cruce max-min-aritmético. La condición de parada es la realización de un número fijo de evaluaciones, 2.000 en todos los casos, permitiendo un estudio comparativo razonable. Para evaluar la buena convergencia del AG hemos realizado 3 ejecuciones de los métodos utilizando en cada una de ellas una semilla distinta para el generador de números aleatorios.

3.3.3.1. Estudio sobre los Métodos Basados en el Ajuste Lateral y de Amplitud

En la Tabla 3.7 se muestran los métodos considerados en este estudio. Los controladores presentados en la Tabla 3.8, donde el % representa la mejora respecto a cada criterio del controlador On-Off y #R el número de reglas difusas, se corresponden con los mejores individuos de la población tras 2.000 evaluaciones considerando las tres ejecuciones realizadas. Además, en la Tabla también se muestra la media de dichos resultados para cada criterio. El tiempo requerido para cada evaluación del modelo es de aproximadamente 60 segundos. Por lo tanto, el tiempo de ejecución estimado para 2.000 evaluaciones es de 1 día, calculado como el producto del número de evaluaciones y el tiempo de evaluación.

Tabla 3.7: Métodos considerados en la comparación

Método	Descripción
GLA	Ajuste-LA Global
LLA	Ajuste-LA Local
GLA+S	Ajuste-LA Global + Selección de Reglas
LLA+S	Ajuste-LA Local + Selección de Reglas

A la vista de los resultados, podemos resaltar que todos los controladores difusos obtenidos por los métodos propuestos consiguen importantes mejoras sobre los dos controladores originales, el *On-Off* y el controlador difuso inicial. En este caso, se consiguieron todas las metas impuestas por los expertos, superando con creces los resultados esperados.

Respecto a la combinación de los métodos de ajuste y selección de reglas,

Tabla 3.8: Resultados obtenidos por los métodos

Método	Modelo	#R	PMV			CO ₂		Energía		Estabilidad	
			O ₁	O ₂	O ₃	O ₄	%	O ₅	%		
Controladores	ON-OFF	—	0,0	0	0	3.206.400	—	1.136	—		
Originales	C. DIFUSO	172	0,0	0	0	2.901.686	9,50	1.505	-32,48		
Ajuste-LA Global	GLA1	172	0,92	0	0	2.267.300	29,29	938	17,43		
	GLA2	172	0,81	0	0	2.180.391	31,70	717	36,88		
	GLA3	172	0,82	0	0	2.289.745	28,59	736	35,21		
	\overline{GLA}	172	0,9	0	0	2.245.812	29,96	797	29,84		
Ajuste-LA Local	LLA1	172	0,54	0	1	2.230.260	30,44	722	36,44		
	LLA2	172	0,81	0	0	2.236.515	30,25	758	33,27		
	LLA3	172	0,94	0	0	2.215.529	30,90	812	28,52		
	\overline{LLA}	172	0,8	0	0	2.227.435	30,53	764	32,75		
Ajuste-LA Global + Selección de Reglas	GLA+S1	100	0,66	0	0	2.229.108	30,48	625	44,98		
	GLA+S2	108	1,00	0	0	2.303.994	28,14	577	49,20		
	GLA+S3	105	0,60	0	0	2.228.887	30,49	701	38,29		
	$\overline{GLA + S}$	104	0,8	0	0	2.253.996	29,70	634	44,19		
Ajuste-LA Local + Selección de Reglas	LLA+S1	98	0,79	0	0	2.184.928	31,86	706	37,85		
	LLA+S2	114	0,32	0	2	2.241.309	30,10	761	33,01		
	LLA+S3	127	0,93	0	0	2.177.610	32,08	696	38,73		
	$\overline{LLA + S}$	113	0,7	0	1	2.201.282	31,35	721	36,53		

GLA+S y LLA+S respectivamente, obtienen mayor precisión que si los consideramos por separado, además de reducir el número de reglas, mejorando el equilibrio entre interpretabilidad y precisión. Así, la selección de reglas nos ayuda a obtener controladores difusos más simples y que al mismo tiempo mejoran su precisión.

Podemos resaltar que, el método LLA+S con la semilla 1 presenta una mejora del 32 % en energía, del 38 % en estabilidad y todo ello con solo 98 reglas de las 172 reglas posibles, es decir, existen soluciones óptimas más simples con más o menos

la misma o mejor precisión (ver la semilla 2) que otras con un número mayor de reglas. Este hecho también se pone de manifiesto en el algoritmo GLA+S, que presenta mejoras de alrededor del 30% en energía y de un 45% en estabilidad con tan solo 100 reglas.

Por los resultados obtenidos, parece ser que el enfoque global del ajuste-LA junto con la selección de reglas es el mejor enfoque, ya que es el más interpretable y presenta un ahorro de energía similar y mejor estabilidad que el enfoque local. Por otro lado, el algoritmo propuesto parece presentar una buena convergencia y ser independiente de factores aleatorios.

La Figura 3.14 representa la base de datos inicial y final del controlador difuso obtenido por el método GLA+S1 en la Tabla 3.8. Ésta muestra como pequeñas modificaciones en las funciones de pertenencia provocan grandes mejoras en el comportamiento del controlador difuso. La Figura 3.15 representa las tablas de decisión del controlador difuso obtenido con el método GLA+S1. En este caso se han eliminado un elevado número de reglas con respecto al controlador inicial, obteniendo modelos más simples (se eliminaron 72 reglas).

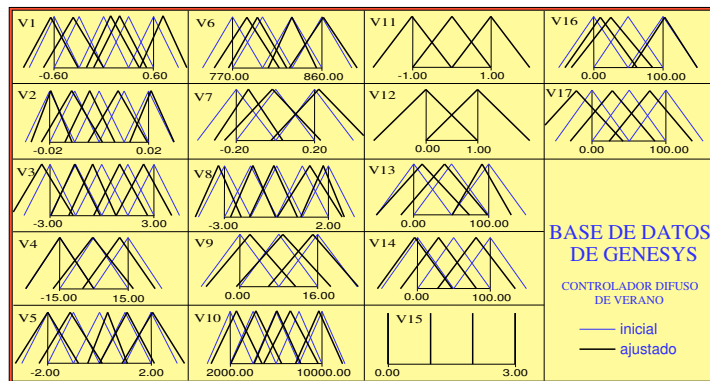


Figura 3.14: BD inicial y ajustada del modelo obtenido con GLA+S (semilla1)

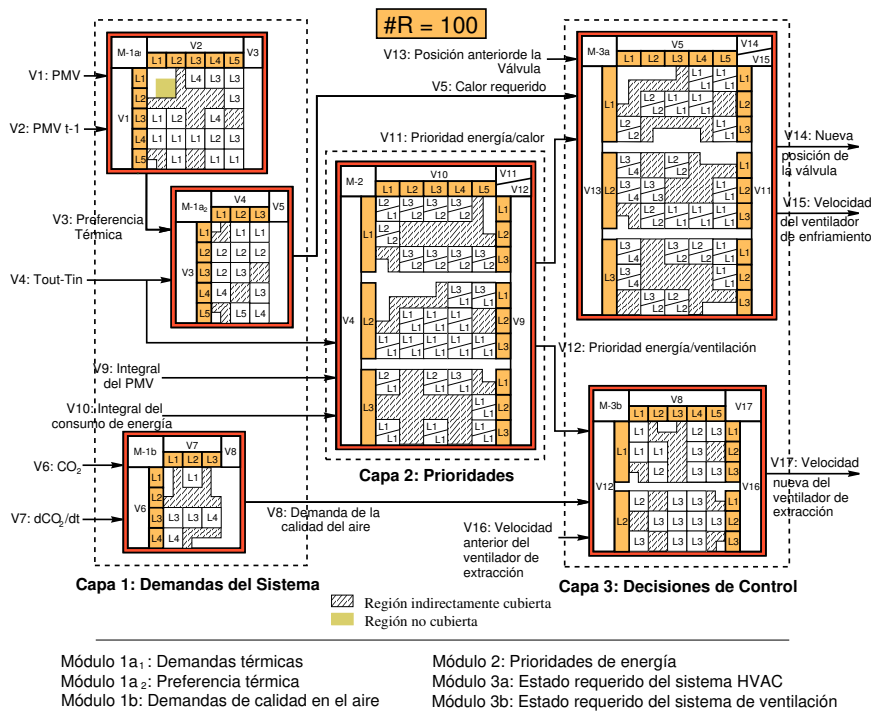


Figura 3.15: BR y estructura final del modelo obtenido con GLA+S (semilla1)

3.3.3.2. Análisis Considerando los Resultados Obtenidos por otros Métodos

Con el objetivo de ver el buen comportamiento de los métodos propuestos en este capítulo se realiza un estudio comparativo con los resultados medios obtenidos por los métodos mostrados en la Tabla 3.9.

La Tabla 3.10 presenta una comparativa entre los resultados medios de las 3 ejecuciones realizadas para cada uno de los métodos indicados en la Tabla 3.9. Como podemos ver, los métodos basados en el ajuste-LA muestran mejoras importantes en estabilidad con respecto al resto de los métodos. Además, presentan una importante mejora en energía con respecto al ajuste clásico, y una mejora razonable de la energía respecto al ajuste lateral.

Los métodos que combinan la selección de reglas con el ajuste han conseguido mucho mejores resultados que si consideramos cada una de las técnicas por se-

Tabla 3.9: Métodos considerados para la comparación

Ref., Año	Método	Descripción
[ACC ⁺ 05], 2005	S	Selección de Reglas (parte C_S de GLA-S)
[ABC ⁺ 03b], 2003	T	Ajuste Clásico
[AAFB ⁺ 06], 2006	GL	Ajuste Lateral Global (parte C^L de GLA)
[AAFB ⁺ 06] ¹ , 2006	LL	Ajuste Lateral Local (parte C^L de LLA)
—	GLA	Ajuste-LA Global
—	LLA	Ajuste-LA Local
—	T+S	Ajuste Clásico (T) + Selección de Reglas (S)
[AAFB ⁺ 06], 2006	GL+S	Ajuste Lateral Global (GL) + Selección de Reglas (S)
[AAFB ⁺ 06] ¹ , 2006	LL+S	Ajuste Lateral Local (LL) + Selección de Reglas (S)
—	GLA+S	Ajuste-LA Global (GLA) + Selección de Reglas (S)
—	LLA+S	Ajuste-LA Local (LLA) + Selección de Reglas (S)

¹Adaptación al enfoque local del método presentado en la referencia

parado, especialmente en la combinación del ajuste-LA global con la selección de reglas. Además, en el caso del GLA+S, el nivel de interpretabilidad es muy cercano al original, ya que las reglas iniciales y la forma de las funciones de pertenencia permanecen fijas (triangulares y simétricas). Cabe destacar el hecho de que, en general, la simplificación de los controladores difusos solo mantiene un 60,4 % de las reglas iniciales. Además, considerado la selección de reglas se ayuda a reducir el espacio de búsqueda y favorece la habilidad de la técnica de ajuste para obtener buenas soluciones.

En todos los modelos obtenidos con el método GLA+S se ha obtenido un buen equilibrio entre energía y estabilidad, manteniendo el resto de criterios dentro de los valores óptimos. El método GLA+S presenta mejoras de alrededor del 29.7 % en energía y de un 44.2 % en estabilidad, al mismo tiempo que mantiene los criterios de confort y de la calidad del aire dentro de los niveles requeridos. Además, los algoritmos propuestos presentan una buena convergencia y parecen ser independientes de factores aleatorios.

Tabla 3.10: Comparación entre los diferentes métodos

Método	#R	PMV		CO ₂	Energía		Estabilidad	
		O ₁	O ₂	O ₃	O ₄	%	O ₅	%
Resultados Medios								
\bar{S}	160	0,1	0	0	2.886.422	9,98	1.312	-15,52
\bar{T}	172	0,0	0	0	2.586.717	19,33	1.081	4,84
\overline{GL}	172	0,9	0	0	2.325.093	27,49	1.072	5,66
\overline{LL}	172	0,7	0	0	2.369.013	26,12	926	18,52
$\overline{T+S}$	109	0,1	0	0	2.536.849	20,88	1.057	6,98
$\overline{GL+S}$	113	0,7	0	0	2.287.993	28,64	800	29,58
$\overline{LL+S}$	110	0,5	0	0	2.326.088	27,46	804	29,26
\overline{GLA}	172	0,9	0	0	2.245.812	29,96	797	29,84
\overline{LLA}	172	0,8	0	0	2.227.435	30,53	764	32,75
$\overline{GLA+S}$	104	0,8	0	0	2.253.996	29,70	634	44,19
$\overline{LLA+S}$	113	0,7	0	1	2.201.282	31,35	721	36,53

3.4. Sumario

En este capítulo se ha realizado una extensión del método de post-procesamiento para el ajuste lateral de funciones de pertenencia, proponiendo un nuevo método para realizar un ajuste lateral y de amplitud de las funciones de pertenencia. Este enfoque propone una nueva representación simbólica con dos parámetros (α , β), que representan el desplazamiento lateral y la variación de la amplitud del soporte de una etiqueta respectivamente. Además, hemos realizado un análisis de cómo interactúa el ajuste-LA con la selección de reglas, comprobando que obtiene mejores resultados que el resto de técnicas, sobre todo en problemas complejos como el problema del sistema HVAC. Del estudio realizado podemos sacar las siguientes conclusiones:

- La representación de reglas basada en las 3-tuplas lingüísticas junto con el algoritmo evolutivo de ajuste propuesto, representa un buen mecanismo

para obtener modelos más precisos, aunque esto provoque la pérdida de interpretabilidad en un cierto grado.

- El ajuste-LA realiza una reducción del espacio de búsqueda (mayor cuanto más complejo es el problema) que nos permite obtener modelos más óptimos.
- En muchas ocasiones, solo se realizan pequeñas modificaciones respecto a las funciones de pertenencia originales, por lo que se mantiene la interpretabilidad a un nivel razonable.
- Las características complementarias que presentan el ajuste y la selección de reglas permiten obtener buenas soluciones. La habilidad de la selección de reglas para reducir el número de reglas, seleccionando únicamente aquellas con buena cooperación, es combinada con la mejora en la precisión provocada por el ajuste, obteniendo modelos precisos y compactos (sobre todo en problemas complejos). Esta combinación incrementa el espacio de búsqueda (ajuste de parámetros + selección de reglas), pero sigue siendo fácil de manejar gracias a la nueva representación de regla.

Finalmente, queremos destacar que, después de realizar diferentes estudios sobre la combinación del ajuste-LA con la selección de reglas, existe un gran conjunto de óptimos locales que presentan un número diferente de reglas y que más o menos tienen la misma (o incluso mejor en algunos casos) precisión. Esto es de especial interés porque podríamos buscar este tipo de soluciones, obteniendo modelos más simples y comprensibles con la misma o similar precisión a los modelos obtenidos con las técnicas propuestas en este capítulo.

Comentarios Finales

Dedicaremos esta sección a resumir brevemente las conclusiones que esta memoria puede aportar. Además, presentaremos las publicaciones asociadas a esta tesis y comentaremos algunos aspectos sobre trabajos futuros que siguen la línea aquí expuesta y sobre otras líneas de investigación que se pueden derivar.

A. Resumen y Conclusiones

Hemos presentado diferentes métodos para obtener modelos lingüísticos en distintos problemas. Esto se ha realizado con la intención de obtener modelos precisos manteniendo en la medida de lo posible un alto grado de comprensibilidad. Los siguientes apartados resumen brevemente los resultados obtenidos y presentan algunas conclusiones sobre los mismos.

A.1 Uso de la Representación de 2-Tuplas para el Aprendizaje A Priori de la Base de Datos

En esta memoria se ha propuesto un nuevo método de aprendizaje de la BC por medio de un aprendizaje evolutivo *a priori* de la BD (granularidad y parámetros de translación) basado en el modelo de representación de reglas de 2-tuplas lingüísticas y considerando un nuevo sistema de inferencia. Además, se han propuesto dos métodos nuevos *ad-hoc* guiados por ejemplos para analizar su influencia y la de otros métodos de generación de reglas en el enfoque de aprendizaje propuesto.

La utilización del esquema de aprendizaje junto con el modelo de repre-

sentación de reglas de 2-tuplas lingüísticas y el nuevo sistema de inferencia permite realizar una importante reducción del espacio de búsqueda, utilizando un único parámetro por etiqueta en lugar de los tres parámetros por etiqueta del aprendizaje a priori clásico (en el caso de funciones de pertenencia triangulares). Esta reducción provoca que para cada combinación del número de etiquetas se consiga un mejor ajuste de los parámetros de las funciones de pertenencia, facilitando la obtención de modelos lingüísticos más precisos y compactos.

Para este método, la selección del proceso de generación de reglas en el aprendizaje *a priori* de la BD es un aspecto importante. El uso de un método de generación de reglas simple para la búsqueda de las reglas básicas que mejor cubran los datos de ejemplo favorece el aprendizaje de una BD mejor y la generación de BRs con un número menor de reglas. Al influir más el aprendizaje de la BD que la composición de la BR en el comportamiento del sistema, este tipo de métodos facilitan la obtención de modelos más precisos y compactos.

Además, al considerar un enfoque global y al mantener la forma de las funciones de pertenencia iniciales, la interpretabilidad de los modelos obtenidos se ha mantenido a un alto nivel respecto al aprendizaje clásico de sistemas difusos.

Después de realizar diferentes estudios sobre el método de aprendizaje propuesto, se ha comprobado que existen modelos óptimos con diferente nivel de precisión y simplicidad en función de lo que penalicemos el número de reglas, dando lugar a distintos grados de equilibrio entre interpretabilidad y precisión. Este aspecto muestra la fuerte multi-modalidad del problema de aprendizaje de la BD a priori.

A.2 Ajuste Lateral y de Amplitud de SBRDs

El ajuste lateral de funciones de pertenencia basado en la representación de 2-tuplas lingüísticas permite la translación simbólica de una etiqueta considerando un único parámetro por etiqueta. Sin embargo, durante todo el proceso de ajuste la amplitud del soporte de las funciones de pertenencia permanece fija. La amplitud determina la especificidad de una etiqueta y permite mejorar potencialmente la precisión del modelo, ya que permite determinar la región de mejor cubrimiento de la etiqueta.

En esta memoria se ha propuesto una nueva representación simbólica con dos parámetros (α, β) , que representan el desplazamiento lateral y la variación de la amplitud del soporte de una etiqueta respectivamente. Basándonos en esta nueva

representación, se ha realizado una extensión del método de post-procesamiento para el ajuste lateral de funciones de pertenencia, proponiendo un nuevo método de ajuste para realizar un ajuste lateral y de amplitud de las funciones de pertenencia.

Del estudio realizado podemos concluir que la nueva representación de reglas basada en 3-tuplas lingüísticas junto con el algoritmo evolutivo de ajuste propuesto representa un buen mecanismo para obtener modelos más precisos, aunque este provoque la pérdida de interpretabilidad en un cierto grado. Por un lado, esto se debe a que el ajuste-LA realiza una reducción del espacio de búsqueda (mayor cuanto más complejo es el problema) respecto al ajuste clásico. Por otro lado, el poder modificar la amplitud del soporte de las funciones de pertenencia nos proporciona una mayor flexibilidad respecto al ajuste lateral.

Aunque la nueva representación provoca la pérdida de interpretabilidad en un cierto grado, en muchas ocasiones solo se realizan pequeñas modificaciones con respecto a las funciones de pertenencia originales, por lo que se mantiene la interpretabilidad a un nivel razonable.

Además, se ha realizado un análisis de como interactúa el ajuste-LA con la selección de reglas, comprobando que se obtienen mejores resultados que con el resto de técnicas, sobre todo en problemas complejos como el problema del sistema HVAC. Las características complementarias que presentan el ajuste y la selección de reglas permiten obtener buenas soluciones. La habilidad de la selección de reglas para reducir el número de reglas, seleccionando únicamente aquellas con buena cooperación, es combinada con la mejora en la precisión provocada por el ajuste, obteniendo modelos precisos y compactos (sobre todo en problemas complejos). Esta combinación incrementa el espacio de búsqueda (ajuste de parámetros + selección de reglas), pero sigue siendo fácil de manejar gracias a la nueva representación de regla.

Finalmente, queremos destacar que, después de realizar diferentes estudios sobre la combinación del ajuste-LA con la selección de reglas, existe un conjunto de óptimos locales con una alta precisión y que presentan un número diferente de reglas. Esto sugiere que el problema de ajuste-selección es fuertemente multi-modal, y es de especial interés porque dentro de este conjunto de óptimos podríamos buscar las soluciones con menor número de reglas, obteniendo modelos más simples con la misma o similar precisión.

B. Publicaciones Asociadas a la Tesis

A continuación se presenta un listado de las publicaciones asociadas a la tesis.

- Publicaciones en revistas internacionales:
 1. Alcalá R., Alcalá-Fdez J., Herrera F. y Otero J. (2006) Genetic learning of accurate and compact fuzzy rule based systems based on the 2-tuples linguistic representation. *International Journal of Approximate Reasoning*. Por aparecer.
 2. Alcalá R., Alcalá-Fdez J., Gacto M. J. y Herrera F. (2006) Rule base reduction and genetic tuning of fuzzy systems based on the linguistic 3-tuples representation. *Soft Computing*. Por aparecer.
- Publicaciones en congresos internacionales:
 1. Alcalá R., Alcalá-Fdez J., Herrera F. y Otero J. (2005) Genetic learning of the knowledge base of a fuzzy system by using the linguistic 2-tuples representation. En *Proceedings of the IEEE International Conference on Fuzzy Systems*, páginas 797-802. Florida, EE.UU.
 2. Alcalá R., Alcalá-Fdez J., Herrera F. y Otero J. (2005) A New Genetic Fuzzy System Based on Linguistic 2-Tuples to Learn Knowledge Bases. *I International Workshop on Genetic Fuzzy Systems*, páginas 107-112. Granada, Spain.
 3. Alcalá R., Alcalá-Fdez J., Gacto M. J. y Herrera F. (2005) Genetic lateral and amplitude tuning of membership functions for fuzzy systems. En *Proceedings of the 2nd International Conference on Machine Intelligence*, páginas 589-595. Tozeur, Túnez.
 4. Alcalá R., Alcalá-Fdez J., Berlanga F., Gacto M. J. y Herrera F. (2006) Genetic lateral and amplitude tuning with rule selection for fuzzy control of heating, ventilating and air conditioning systems. En Ali M. y Dapoigny R. (Eds.) *Proc. 19th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE06)*, number 4031 in *Lecture Notes in Computer Science*, páginas 127-131. Springer-Verlag, Berlin.

- Publicaciones en congresos nacionales:

1. Alcalá R., Alcalá-Fdez J., Gacto M. J. y Herrera F. (2005) Ajuste Evolutivo Lateral y de Amplitud de etiquetas para Sistemas Basados en Reglas Difusas. Simposio de Inteligencia Computacional, páginas 481-488. Granada, España.

C. Líneas de Investigación Futuras

A continuación, consideraremos algunas líneas de trabajo futuras a partir de las conclusiones obtenidas en esta memoria.

C.1 Uso de Algoritmos Genéticos Multiobjetivo para la Búsqueda de la Precisión/Interpretabilidad mediante la Selección de Reglas y Ajuste Lateral y de Amplitud o Aprendizaje de la Base de Conocimiento completa

Las conclusiones presentadas nos hacen pensar que siguen quedando espacios no explorados en los que podríamos encontrar soluciones más interpretables con una precisión parecida. Es por ello que dos trabajos futuros se enfocarán hacia este objetivo:

1. *Mejora en el Balance entre Precisión e Interpretabilidad mediante la Reducción del Número de Reglas.*

En el Capítulo 3 hemos visto cómo al combinar el ajuste-LA con la selección de reglas se provocó la eliminación de muchas reglas sin perder (o incluso mejorándola) precisión. Esto significa que un número excesivo de reglas podría ir en detrimento de la capacidad de ajuste de las funciones de pertenencia, debido al elevado espacio de búsqueda. Se debe tener en cuenta que, a medida que el proceso de ajuste avanza en la búsqueda, los ajustes van mejorando la interacción de las reglas y por lo tanto se hace más difícil la eliminación de éstas, aún cuando dichas reglas entorpezcan (provocando una adaptación de las reglas vecinas a la regla correspondiente). De esta manera, fomentar la eliminación de reglas desde un principio, puede ser apropiado para:

- Favorecer la simplicidad de los modelos. Un sistema compacto con pocas reglas requiere un menor esfuerzo para ser interpretado.
- Eliminar reglas redundantes, inconsistentes o contradictorias. Una vez que este tipo de reglas y sus vecinas se ajustan para mejorar el modelado de la zona, se hace mucho más difícil su eliminación. Si eliminamos de la población una regla mala después de que sus reglas vecinas hayan adaptado sus funciones de pertenencia para mejorar el comportamiento global del sistema, se puede empeorar la precisión del modelo, haciendo difícil la eliminación de dichas reglas, puesto que con alta probabilidad la solución sería eliminada de la población antes de que las funciones de pertenencia de las reglas vecinas fuesen readaptadas.
- Ayudar al ajuste de las funciones de pertenencia, dándole mayor protagonismo. Con un número menor de reglas, el proceso de ajuste maneja un espacio de búsqueda menor, compensando la pérdida de precisión impuesta por la eliminación de reglas.

Por lo tanto, se propone considerar dos objetivos: precisión y simplicidad. Para ello, proponemos utilizar un algoritmo evolutivo multiobjetivo [CVL02, Deb01] con dichos objetivos. En cualquier problema con múltiples objetivos, siempre hay un conjunto de soluciones que son superiores a las demás en el espacio de búsqueda cuando se consideran todos los objetivos. Dichas soluciones se conocen como soluciones no dominadas (conjunto Pareto). Ninguna de las soluciones contenidas en el conjunto Pareto es absolutamente mejor que el resto de las no dominadas. Se podría obtener por tanto, un conjunto de soluciones que abarcaría desde los modelos más precisos hasta los más simples, pasando por distintos niveles de equilibrio entre ambos criterios. De esta forma podríamos escoger la solución que más nos interesara en cada momento. Además, esta propuesta se podría aplicar a todos los métodos presentados en esta memoria.

2. *Diseño de Distintas Medidas de Calidad de Regla.*

Hemos comprobado que existen muchas configuraciones diferentes de las funciones de pertenencia que nos permiten obtener modelos con una precisión similar. Un aspecto interesante sería centrarse en aquellas soluciones que teniendo una precisión similar al resto sean más interpretables.

Una forma de conseguirlo es diseñar distintas medidas de calidad de regla que, junto con el objetivo de minimizar el número de reglas, se puedan

optimizar mediante un algoritmo multiobjetivo, dando lugar a modelos con una información más útil. Algunas posibilidades podrían ser:

- Minimizar los desplazamientos laterales. De esta forma el significado original de las etiquetas se mantendrá.
- En el mismo sentido, minimizar las variaciones de amplitud.
- Maximizar distintas medidas de cubrimiento.
- Verificar propiedades de consistencia.

En definitiva, mejorar la calidad de las reglas para que representen de manera más fiel los datos que cubren, llegando a tener un mayor significado por sí solas. Este enfoque puede ser de gran utilidad en problemas de minería de datos, en los que se desea obtener reglas significativas y se manejan espacios de búsqueda muy grandes.

C.2 Aplicación de las Técnicas Propuestas a la Determinación de Particiones Difusas para la Extracción de Reglas de Asociación

Los métodos propuestos en esta memoria nos permiten realizar una adaptación al contexto. De esta forma, podríamos adaptarlos para aprender las particiones difusas para extraer reglas de asociación con alto grado de cubrimiento de los ejemplos y de gran interés que nos aporten conocimiento.

Para ello utilizaremos como referencia los trabajos realizados por el equipo de investigación de T.P. Hong [HCWL06, HCWL04] sobre la extracción de reglas de asociación, el cual sigue de cerca nuestros trabajos.

Apéndices

Apéndice A

Introducción a los Sistemas Basados en Reglas Difusas

Este apéndice está dedicado a introducir los distintos tipos de SBRDs existentes. En concreto, describiremos los SBRDs lingüísticos, empleados habitualmente en MDL, y los SBRDs aproximativos y TSK, considerados en MDP. Para describirlos, partiremos del concepto de grafo difuso propio de la Teoría de Conjuntos Difusos, que puede considerarse como el origen de dichos sistemas. Realizaremos un análisis más profundo de los SBRDs lingüísticos, ya que la presente memoria está principalmente centrada en ellos, al estar encuadrada en el área del MDLE.

A.1. Grafos Difusos para Modelado de Sistemas

Los *grafos difusos* fueron originalmente introducidos por Zadeh en [Zad71] y desarrollados por él mismo en [Zad74, Zad76] o, más recientemente, en [Zad96], donde ilustra el concepto de grafos difusos de la siguiente manera: «*La función principal de un grafo difuso es servir como una representación de una dependencia definida de forma imprecisa*». Así, este concepto resulta muy apropiado para el Modelado de Sistemas ya que definir un modelo no es más que establecer la relación existente entre las diferentes dependencias funcionales del sistema que se desea modelar.

Un grafo difuso se compone de un conjunto de *puntos difusos* del tipo

SI X_1 es A_1 y ... y X_n es A_n **ENTONCES** Y es B ,

siendo $X = (X_1, \dots, X_n)$ las variables difusas de entrada, Y la variable difusa de salida, y siendo A_1, \dots, A_n y B las funciones de pertenencia asociadas a las variables de entrada y salida respectivamente.

Simplificando esta expresión con $\mathbf{A} = A_1 \times \dots \times A_n$ (donde el símbolo ‘ \times ’ denota el producto cartesiano) tenemos

SI X es \mathbf{A} **ENTONCES** Y es B ,

que se puede expresar como una restricción difusa de una variable combinada (X, Y) , es decir,

$$(X, Y) \text{ es } \mathbf{A} \times B.$$

La función de pertenencia de $\mathbf{A} \times B$ se define empleando un operador de conjunción ‘ \wedge ’ (generalmente la t-norma del *mínimo*) como

$$\mu_{\mathbf{A} \times B}(\mathbf{x}, y) = \mu_{\mathbf{A}}(\mathbf{x}) \wedge \mu_B(y) = \mu_{A_1}(x_1) \wedge \dots \wedge \mu_{A_n}(x_n) \wedge \mu_B(y).$$

Por tanto, un grafo difuso se construye con una colección de r puntos difusos que representa una dependencia funcional f^* de Y sobre X y se puede definir como

$$f^* = \mathbf{A}_1 \times B_1 + \dots + \mathbf{A}_r \times B_r,$$

o, de forma más compacta

$$f^* = \sum_{i=1}^r \mathbf{A}_i \times B_i.$$

Tradicionalmente, en el campo del Modelado de Sistemas, a estos puntos difusos se les conoce como *reglas difusas*. En general, se suelen imponer ciertas restricciones para hacer a estas reglas más interpretables. La más común consiste en emplear variables lingüísticas [Zad75] en lugar de difusas, es decir, forzar a que, para cada variable, las funciones de pertenencia consideradas en cada regla difusa pertenezcan a un conjunto común de funciones de pertenencia asociadas a la misma. En este caso, las relaciones funcionales se denominan *reglas lingüísticas difusas* o, simplemente, reglas lingüísticas. Esta restricción hace que dicha estructura sólo sea un subconjunto de la posible colección de dependencias funcionales

cubiertas por los grafos difusos. La figura A.1(a) ilustra una representación aproximada de relaciones con un grafo difuso restringido de esta forma.

Por contra, si consideramos toda la potencialidad de los grafos difusos haciendo que cada punto difuso se describa independientemente a través de las funciones de pertenencia contenidas en él, la capacidad de aproximación será mayor pero las dependencias funcionales serán más difíciles de comprender. En la figura A.1(b) se muestra la representación gráfica de un grafo difuso sin este tipo de restricciones.

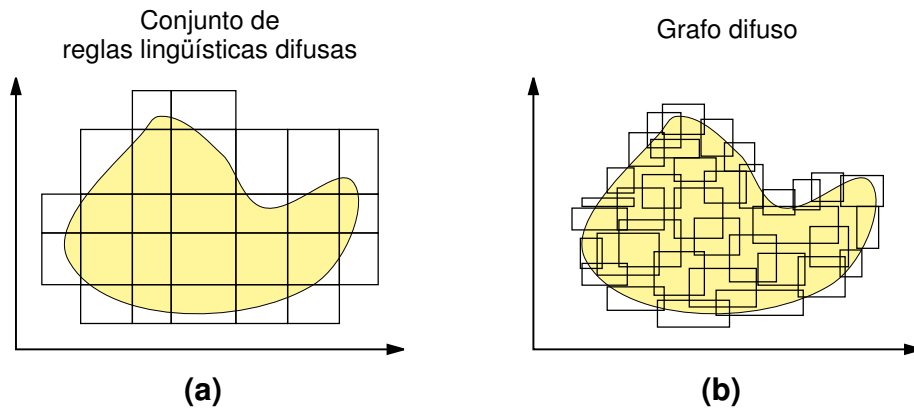


Figura A.1: Representación aproximada de relaciones: (a) considerando grafos difusos restringidos, (b) considerando grafos difusos sin restricciones

Los grafos difusos empleando variables lingüísticas son equivalentes a los *SBRDs lingüísticos*, mientras que los grafos difusos en general se denominan *SBRDs aproximativos*. Como hemos comentado, el objetivo principal de los primeros es obtener relaciones claramente interpretables, de forma que son ideales para MDL. Los segundos, por contra, consideran como principal requisito la aproximación al modelo, de forma que son más adecuados para MDP. En la figura A.2 podemos observar el paralelismo existente entre los SBRDs lingüísticos y aproximativos.

En la literatura especializada podemos encontrar otras designaciones equivalentes para distinguir entre estos dos tipos de sistemas. Entre otras, tenemos las siguientes:

- SBRDs con *conjuntos difusos globales o locales* [CFM96], porque en los

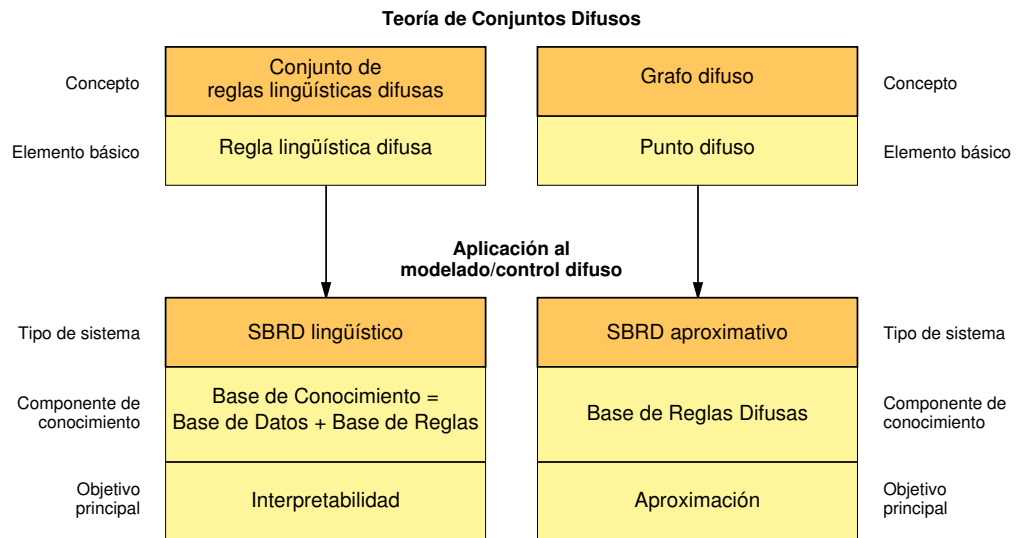


Figura A.2: Paralelismo entre SBRDs lingüísticos (obtenidos a partir de un conjunto de reglas lingüísticas difusas) y SBRDs aproximativos (obtenidos a partir de un grafo difuso sin restricciones)

SBRDs lingüísticos la semántica (es decir, las definiciones de las funciones de pertenencia) es común para el conjunto completo de reglas difusas mientras que en los SBRDs aproximativos se considera una semántica particular para cada regla,

- SBRDs *basados en dominios* o *basados en reglas* [CV94], porque en los SBRDs lingüísticos se define un dominio y partición comunes para todas las reglas mientras que en los SBRDs aproximativos las funciones de pertenencia son particulares a cada regla, o
- SBRDs con *partición de rejilla* o *partición dispersa* [Fri97], ya que en los SBRDs lingüísticos se consideran subespacios de entrada difusos inscritos en la rejilla difusa formada por la intersección de las diferentes semánticas de las variables de entrada mientras que en los SBRDs aproximativos no se considera una partición homogénea.

En las secciones siguientes introduciremos más detenidamente estos dos tipos de SBRDs.

A.2. Sistemas Basados en Reglas Difusas Lingüísticos

Los SBRDs lingüísticos fueron inicialmente propuestos por Mamdani y Assilian [Mam74, MA75], 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* (nombre que ya acuñaron Mamdani y Assilian en sus primeros trabajos [MA75]), ya que su aplicación principal ha sido históricamente el control de sistemas.

Los SBRDs lingüísticos son una translación directa de los SBRDs puros al campo de la ingeniería. De este modo, mantienen tanto el formato de regla difusa como la estructura básica de los sistemas puros, con lo que siguen empleando un sistema de inferencia que efectúa el razonamiento difuso, teniendo en cuenta la información contenida en una base de conocimiento. La única novedad que presentan 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*¹.

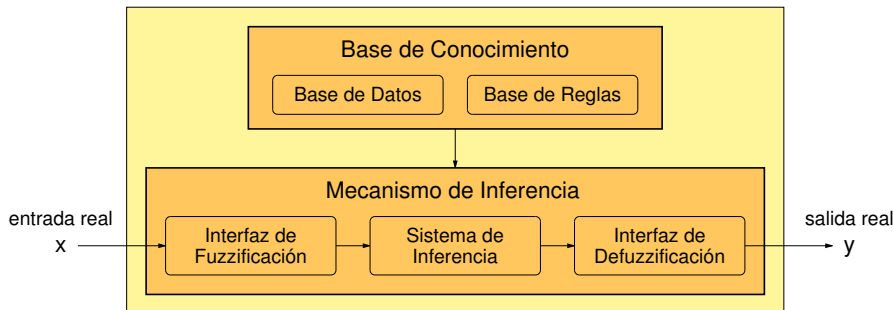


Figura A.3: Estructura básica de un SBRD lingüístico

La figura A.3 muestra la estructura general de los SBRDs lingüísticos. Tal como comentaremos en la sección A.5, en la que se estudiará detenidamente

¹ Los términos 'fuzzificación' y 'defuzzificación' son un calco directo de los vocablos ingleses *fuzzification* y *defuzzification*. Aunque estos términos no existen en la lengua española, reflejan claramente la actuación de ambos interfaces (convertir un número real a difuso o un número difuso a real, respectivamente) y su uso en la comunidad científica está ampliamente extendido

cada uno de los componentes de este tipo de sistemas, la interfaz de fuzzificación establece una aplicación entre valores precisos en el dominio \mathbf{U} de las entradas del sistema y conjuntos difusos definidos sobre el mismo universo de discurso. La interfaz de defuzzificación, en cambio, realiza la operación inversa estableciendo una aplicación entre conjuntos difusos definidos en el dominio \mathbf{V} de las salidas y valores precisos definidos en el mismo universo.

El SBRD lingüístico presenta 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 conocimiento experto en forma de reglas lingüísticas y permite combinar este de un modo muy sencillo con 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 sistema de inferencia, de modo que permite diseñar el SBRD más adecuado para un problema concreto.

Con respecto al tipo de modelado que llevan a cabo, los SBRDs lingüísticos fueron inicialmente pensados para realizar un MDL. Las reglas difusas están formadas por variables lingüísticas de entrada y salida que toman valores dentro de un conjunto de términos con un significado en el mundo real. De este modo, cada regla es una descripción de una condición-acción que puede ser interpretada por un ser humano.

A.3. Sistemas Basados en Reglas Difusas Aproximativos

Como vimos en la sección 1.2 del capítulo 1, los SBRDs lingüísticos presentan ciertos problemas relacionados con la estructura de las reglas difusas que emplean. Si se analizan detenidamente estos problemas, se llega a la conclusión de que la estructura de regla difusa de tipo SI-ENTONCES basada en el manejo de variables lingüísticas posee las siguientes limitaciones [Bas94, CFM96]: partición rígida de los espacios de entrada y salida, dificultad para obtener una partición difusa adecuada cuando las variables presentan una fuerte dependencia, falta de adaptación a espacios altamente dimensionales y necesidad del aumento de la granularidad para ganar precisión con el consecuente crecimiento del número de

reglas.

Debido a esta serie de problemas, que provocan que el SBRD obtenido no presente el grado de exactitud deseado, existe otra variante de los SBRDs lingüísticos, que antepone la exactitud del sistema obtenido a su interpretabilidad. Este tipo de sistemas se denominan habitualmente SBRDs aproximativos. Obviamente, la aplicación principal de los SBRDs aproximativos es el MDP de sistemas.

La estructura de un SBRD aproximativo es prácticamente la misma que la de un lingüístico. La única diferencia radica en el tipo de reglas que se emplean en la BC. En este caso, dichas reglas no manejan variables lingüísticas sino, directamente, variables difusas. Así, la estructura de las reglas difusas empleadas por los sistemas aproximativos es la siguiente:

$$\text{SI } X_1 \text{ es } \hat{A}_1 \text{ y } \dots \text{ y } X_n \text{ es } \hat{A}_n \text{ ENTONCES } Y \text{ es } \hat{B},$$

donde \hat{A}_i y \hat{B} son conjuntos difusos sin una interpretación lingüística directa en lugar de etiquetas lingüísticas.

De este modo, este tipo de sistemas no necesita del empleo de una BD que almacene los términos lingüísticos existentes y los conjuntos difusos que determinan la semántica asociada a los mismos. En este caso, la BC empleada en los SBRDs lingüísticos, que estaba compuesta por la BD comentada y por una BR, queda reducida a una BRD compuesta por un conjunto de reglas que presentan la última estructura mostrada en la cual cada regla individual contiene la semántica que la describe. La figura A.4 muestra gráficamente la diferencia entre la BC de los SBRDs lingüísticos y la BRD de los SBRDs aproximativos.

A.4. Sistemas Basados en Reglas Difusas de Tipo Takagi-Sugeno-Kang

En lugar de trabajar con reglas difusas del tipo introducido en la sección anterior, Takagi y Sugeno [SK88, TS85] propusieron un nuevo modelo de SBRD que empleaba reglas en las que el antecedente estaba constituido por variables lingüísticas o difusas y el consecuente representaba una función de las variables de entrada. La forma más general de este tipo de reglas es la que mostramos a

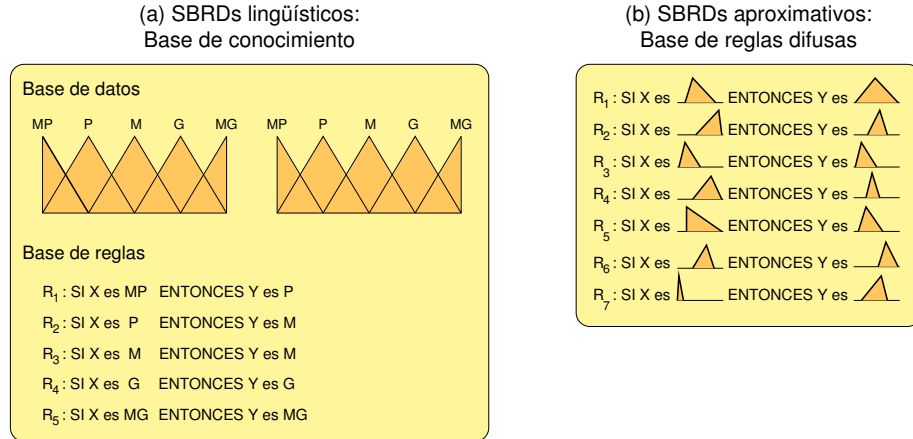


Figura A.4: Comparación gráfica entre una BC lingüística y una BRD aproximativa

continuación, en la que el consecuente constituye una combinación lineal de las variables contenidas en el antecedente:

$$\text{Si } X_1 \text{ es } A_1 \text{ y } \dots \text{ y } X_n \text{ es } A_n \text{ entonces } Y = p_1 \cdot X_1 + \dots + p_n \cdot X_n + p_0,$$

donde X_i 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_i , pueden ser bien etiquetas lingüísticas asociadas con conjuntos difusos en el caso en que las X_i sean variables lingüísticas, o bien conjuntos difusos en el caso en que éstas sean directamente variables difusas. Este tipo de reglas suelen denominarse *reglas difusas de tipo TSK*, en alusión a sus creadores [TS85].

La salida de un SBRD TSK que emplee una Base de Conocimiento formada por m reglas de este tipo se obtiene como la media ponderada de las salidas individuales aportadas por cada regla, Y_i , $i = 1 \dots m$, del siguiente modo:

$$\frac{\sum_{i=1}^m h_i \cdot Y_i}{\sum_{i=1}^m Y_i},$$

donde $h_i = T(A_1(x_1), \dots, A_n(x_n))$ es el grado de emparejamiento entre la parte antecedente de la regla i y las entradas actuales al sistema, $x = (x_1, \dots, x_n)$. T es un operador de conjunción que se modela mediante una t-norma.

De este modo, tal y como enuncian sus creadores en [TS85], este SBRD se basa en la división del espacio de entrada (uni o multidimensional) en varios subespacios difusos (uni o multidimensionales) y en definir una relación de entrada-salida lineal en cada uno de dichos subespacios. En el proceso de inferencia, se combinan estas relaciones parciales en el modo comentado para así obtener la relación global de entrada-salida, teniendo en cuenta la dominancia de las relaciones parciales en sus respectivas áreas de aplicación y el conflicto que se presenta en las zonas en las que existe solapamiento.

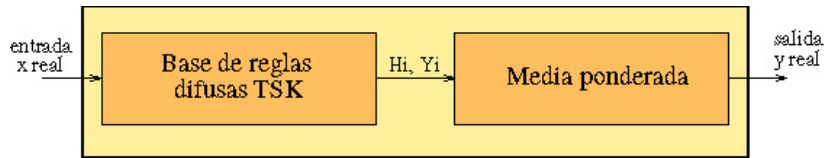


Figura A.5: Estructura básica de un sistema basado en reglas difusas TSK

En la figura A.5 se muestra una representación gráfica de este segundo tipo de SBRDs. Los SBRDs TSK han sido aplicados con éxito a una gran cantidad de problemas prácticos. La ventaja principal de estos sistemas es el hecho de presentar una ecuación del sistema compacta que permite estimar los parámetros p_i empleando métodos clásicos, lo que facilita su diseño. Sin embargo, el mayor inconveniente que tienen asociado está también relacionado con la forma de los consecuentes de las reglas, que al no ser difusos, provocan que el sistema no constituya un marco de trabajo natural para representar el conocimiento experto. Es posible integrar conocimiento experto en estos SBRDs efectuando una pequeña modificación sobre el consecuente de la regla: cuando un experto aporta una regla lingüística con consecuente Y es B , se sustituye dicho consecuente por $Y = p_0$, donde p_0 toma como valor el punto modal del conjunto difuso asociado a la etiqueta B . Este tipo de reglas se suelen denominar *reglas TSK simplificadas* o *de orden cero*. Aun así, no se pueden emplear muchos de los principios de la Lógica Difusa en este tipo de sistemas (por ejemplo, emplear la potencialidad del Razonamiento Aproximado).

En lo que respecta al tipo de modelado que es posible llevar a cabo, los SBRDs TSK constituyen un punto intermedio entre el cualitativo y el difuso. En el caso en que los antecedentes de las reglas estén compuestos por variables lingüísticas, los SBRDs TSK combinan un grado de descripción aceptable (aunque no sea posible interpretar la acción que se lleva a cabo, es posible determinar lingüísticamente de un modo claro cuando dicha acción es aplicada) con un grado de precisión superior

al que se puede obtener con un consecuente difuso. En cambio, en el caso en que las variables que conforman los antecedentes sean difusas y no lingüísticas, es claro que el SBRD se reduce a un modelo difuso que persigue la precisión como característica primordial.

A.5. Análisis de los Componentes de los Sistemas Basados en Reglas Difusas Lingüísticas

Recordando los conceptos introducidos en la sección A.2, un SBRD lingüístico está formado por los siguientes componentes:

- Una BC que contiene las reglas difusas que guían el comportamiento del mismo.
- Una interfaz de fuzzificación, 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 sistema de inferencia, que emplea estos valores y la información contenida en la base para llevar a cabo dicho proceso
- Una interfaz de defuzzificación, que transforma la acción difusa resultante del proceso de inferencia en una acción precisa que constituye la salida global del SBRD.

A continuación, analizaremos en detalle cada una de estas componentes.

A.5.1. La Base de Conocimiento

La BC es la parte esencial del SBRD debido a que los tres componentes restantes del sistema se ocupan de interpretar las reglas contenidas en ella y de posibilitar su utilización en problemas concretos. La componente que almacena estas reglas se denomina BR y la que almacena la estructura de los conjuntos difusos que describen a las variables lingüísticas contenidas en dichas reglas (particiones lingüísticas) se denomina BD.

- La BR está formada por un conjunto de reglas lingüísticas de tipo SI-ENTONCES que, en el caso de los SBRDs con múltiples entradas y una única salida, presentan la siguiente estructura:

R_1 : SI X_1 es A_{11} y ... y X_n es A_{1n} ENTONCES Y es B_1
ADEMÁS

R_2 : SI X_1 es A_{21} y ... y X_n es A_{2n} ENTONCES Y es B_2
ADEMÁS

...

ADEMÁS

R_m : SI X_1 es A_{m1} y ... y X_n es A_{mn} ENTONCES Y es B_m

donde X_i e Y son variables lingüísticas de entrada y salida respectivamente, y A_{ij} y B_i son etiquetas lingüísticas asociadas con conjuntos difusos que determinan su semántica en cada una de las reglas. La BR está compuesta por una serie de reglas de este tipo unidas por el operador ADEMÁS, lo que indica —tal como se verá más adelante— que todas las reglas pueden dispararse ante una entrada concreta.

- La BD contiene la definición de los conjuntos difusos asociados a los términos lingüísticos empleados en las reglas de la BR, así como los valores de los factores de escala que efectúan las transformaciones necesarias para trasladar los universos de discurso en los que están definidos dichos conjuntos a aquellos en que se definen las variables de entrada y salida del sistema.

A.5.2. La Interfaz de Fuzzificación

La *interfaz de fuzzificación* es una de las componentes que permite al SBRD lingüístico trabajar con entradas y salidas reales. Su tarea es la de establecer una correspondencia entre cada valor preciso del espacio de entrada y un conjunto difuso definido en el universo de discurso de dicha entrada. Así, la *interfaz de fuzzificación* trabaja del siguiente modo:

$$A' = F(x_0),$$

donde x_0 es un valor preciso de entrada al SBRD definido en el universo de discurso U , A' es un conjunto difuso definido sobre el mismo dominio y F es un operador de fuzzificación.

El conjunto difuso A' puede construirse al menos de dos maneras diferentes [DHR93]. En el primero de los casos, A' se construye como un conjunto difuso puntual con soporte en x_0 , es decir, con la siguiente función de pertenencia:

$$\mu_{A'}(x) \begin{cases} 1, & \text{si } x = x_0 \\ 0, & \text{en otro caso,} \end{cases}$$

La otra posibilidad comprende la fuzzificación no puntual o aproximada. En ella, $A'(x_0) = 1$ y el grado de pertenencia de los valores restantes de U va disminuyendo según se alejan de x_0 . Este tipo de operador de fuzzificación permite el uso de distintos tipos de funciones de pertenencia, por ejemplo, en el caso de una función de pertenencia triangular:

$$\mu_{A'}(x) \begin{cases} 1 - \frac{|x-x_0|}{\varepsilon}, & \text{si } |x-x_0| \leq \varepsilon \\ 0, & \text{en otro caso.} \end{cases}$$

A.5.3. El Sistema de Inferencia

El sistema de inferencia es la componente encargada de llevar a cabo el proceso de inferencia difuso. Para ello, se hace uso de los principios de la Lógica Difusa para establecer una relación entre los conjuntos difusos definidos en $U = U_1 \times U_2 \times \dots \times U_m$ y conjuntos difusos definidos en V , correspondientes a los dominios de las variables de entrada y salida, respectivamente.

El proceso de inferencia difusa está basado en la aplicación del *modus ponens generalizado*, extensión del modus ponens de la lógica clásica, propuesto por Zadeh según la siguiente expresión [Zad73]:

$$\frac{\begin{array}{l} \text{SI } X \text{ es } A \text{ ENTONCES } Y \text{ es } B \\ X \text{ es } A' \end{array}}{Y \text{ es } B'}$$

Para llevar a la práctica esta expresión, primero es necesario interpretar el tipo de regla que emplea el SBRD. Una regla con la forma: SI X es A ENTONCES Y es B , puede ser representada como una relación difusa entre A y B definida en $U \times V$. Dicha relación se expresa mediante el conjunto difuso R cuya función de pertenencia $\mu_R(x, y)$ presenta la siguiente forma:

$$\mu_R(x, y) = I(\mu_A(x), \mu_B(y)), \quad \forall x \in U, y \in V,$$

donde $\mu_A(x)$ y $\mu_B(y)$ son funciones de pertenencia de los conjuntos difusos A y B , respectivamente, e I es un operador de implicación difuso que modela la relación difusa existente.

Dada una relación R , la función de pertenencia del conjunto difuso B' , resultante de la aplicación del modus ponens generalizado, se obtiene a partir de la *regla composicional de inferencia* introducida por Zadeh en [Zad73] del siguiente modo: «Si R es una relación difusa definida de U a V y A' es un conjunto difuso definido en U , entonces el conjunto difuso B' , inducido por A' , viene dado por la composición de R y A' , esto es:

$$B' = A' \circ R,$$

donde A' juega el papel de una relación unaria» [DHR93].

De este modo, la aplicación de la regla composicional de inferencia sobre reglas que implícitamente denotan una relación difusa, toma la forma de la siguiente expresión:

$$\mu_{B'}(y) = \text{Sup}_{x \in U} \{T(\mu_{A'}(x), I(\mu_A(x), \mu_B(y)))\},$$

donde T es un operador de conjunción de la familia de las *t-normas* asociado al operador de implicación I .

El diseño de un mecanismo de inferencia requiere de la elección de los distintos operadores que intervendrán en dicho proceso, esto es, definir:

- *el operador que permite realizar la intersección* de conjuntos difusos propios de las premisas de las reglas difusas. Para ello se dispone de distintos operadores pertenecientes a la familia de las funciones denominadas *t-normas* [GQ91, TV85].
- *la implicación difusa*, correspondiente al condicional de las reglas ENTONCES. En su inicio, Mamdani empleó la t-norma *mínimo* [Mam74], y en consecuencia varios operadores de esa familia han sido aplicados *a posteriori* [GQ91]. Por otro lado, la familia de funciones de implicación difusa ofrece una amplia variedad de operadores clasificados en distintos grupos, dependiendo del modo en que interpretan la implicación difusa [TV85]. Además de las mencionadas funciones, otros autores sugieren el uso de *t-conormas* y operadores externos a las anteriores definiciones [CK89, CHP97, KKS85] se pueden encontrar estudios específicos de la implicación difusa.

- la *agregación ADEMÁS*, que se emplea para combinar las distintas salidas individuales en una final en conjunción con un método de defuzzificación. La composición de este operador depende del tipo de defuzzificación que el SBRD emplee. En el próximo subapartado, se describirán con mayor detalle tanto este operador como el método de defuzzificación asociado.

A.5.4. La Interfaz de Defuzzificación

Del modo de trabajo del sistema de inferencia del SBRD lingüístico descrito en la sección anterior puede extraerse claramente el hecho de que el proceso de inferencia difusa se aplica a nivel de reglas individuales. De este modo, una vez aplicada la inferencia sobre las m reglas que componen la BR, se obtienen m conjuntos difusos B'_i que representan las acciones difusas que ha deducido el SBRD a partir de las entradas que recibió.

Puesto que el sistema debe devolver una salida precisa, la interfaz de defuzzificación debe asumir la tarea de agregar la información aportada por cada uno de los conjuntos difusos individuales y transformarla en un valor preciso. Existen dos formas de trabajo diferentes para efectuar esta agregación [BD95, CHP97, Wan94]:

1. *Modo A: agregar primero, defuzzificar después.* En este primer caso, la interfaz de defuzzificación lleva a cabo las siguientes tareas:
 - Agrega los conjuntos difusos individuales inferidos B'_i , para obtener un conjunto difuso final B' , empleando para ello un *operador de agregación difuso* G que, modelando el operador *ADEMÁS*, relaciona las reglas de la base:

$$\mu_{B'}(y) = G \left\{ \mu_{B'_1}(y), \dots, \mu_{B'_m}(y) \right\}.$$

- Mediante un *método de defuzzificación* D , transforma el conjunto difuso B' obtenido en un valor preciso y_0 , que será proporcionado como salida global del sistema:

$$y_0 = D(\mu_{B'}(y)).$$

2. *Modo B: defuzzificar primero, agregar después.* Este segundo modo de trabajo considera individualmente la contribución de cada conjunto difuso inferido y el valor preciso final se obtiene mediante una operación (una media,

una suma ponderada o la selección de uno de ellos, entre otras) sobre un valor preciso característico de cada uno de los conjuntos difusos individuales.

De este modo, se evita el cálculo del conjunto difuso final B' , hecho que ahorra una gran cantidad de tiempo computacional. Este modo de operación supone una aproximación distinta al concepto representado por el operador *ADEMÁS*.

Inicialmente fue propuesto el *Modo A*, el cual fue empleado por Mamdani en su primera aproximación al control difuso [Mam74]. En los últimos años, la modalidad *B* ha sido muy empleada [CHP97, DHR93, SY93], sobre todo en sistemas de tiempo real, donde se requieren tiempos de respuesta rápidos.

Cuando se trabaja en *Modo A*, la función del operador de agregación *ADEMÁS* es unir todos los conjuntos difusos, resultantes de la inferencia de cada regla, en un único conjunto difuso global. Para definir matemáticamente este operador, se emplean distintos operadores, principalmente t-normas y t-conormas, los cuales están descritos en [BD95], donde también se analizan sus propiedades en detalle.

En cuanto a la definición matemática del método de defuzzificación a emplear para transformar el conjunto difuso global resultante del proceso de inferencia en un valor preciso de salida, encontramos que los más habituales cuando se trabaja en *Modo A* son: el *centro de gravedad*, el *centro de sumas* (aproximación al centro de gravedad computacionalmente más rápida de obtener) y la *media de los máximos* [DHR93].

Por otro lado, en caso de emplear el *Modo B*, los *operadores de agregación* más utilizados son la *media*, la *media ponderada* o la selección de algún *valor característico* de los conjuntos difusos en función del grado de importancia de la regla que los ha generado en el proceso de inferencia [CHP97]. Como métodos para extraer valores representativos se suelen emplear el *centro de gravedad* y el *punto de máximo criterio*; 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* de la misma con la entrada al sistema. 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, SY93].

Apéndice B

Algoritmos Genéticos

Puesto que todos los métodos de MDLE propuestos en esta memoria están basados en el uso de los Algoritmos Genéticos como técnica de optimización y búsqueda, dedicaremos este segundo Apéndice a describir las líneas generales de este tipo de algoritmos.

B.1. Introducción

Los Algoritmos Genéticos 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 Algoritmos Genéticos han demostrado ser, tanto desde un punto de vista teórico como práctico, una herramienta óptima para proporcionar una búsqueda robusta en espacios complejos, ofreciendo un enfoque válido para solucionar problemas que requieran una búsqueda eficiente y eficaz.

Los Algoritmos Genéticos 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 pos-

teriormente 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. 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*. Como en todos los Algoritmos Evolutivos, es necesario diseñar una *función de adaptación* 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 Algoritmo Genético 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.

La figura B.1, en la que $P(t)$ denota la población en la generación t , muestra la estructura general de un Algoritmo Genético básico.

A continuación, comentaremos brevemente los aspectos básicos relacionados con los Algoritmos Genéticos, la representación de las soluciones, el mecanismo de selección y los operadores de cruce y mutación.

```
Procedimiento Algoritmo Genetico  
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  $P'(t)$ ;  
             $P(t) = P'(t)$ ;  
            evaluar  $P(t)$ ;  
        FIN  
    FIN
```

Figura B.1: Estructura básica de un algoritmo genético

B.2. Representación de las Soluciones

El esquema de representación o codificación es un factor clave en la aplicación de los Algoritmos Genéticos, ya que éstos manipulan directamente una representación codificada del problema y, en consecuencia, el esquema escogido puede limitar de una forma muy severa la ventana desde la cual el Algoritmos Genético 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 [Hol75, Gol89]. 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*: La codificación binaria presenta una serie de inconvenientes importantes cuando se trabaja con problemas que incluyen variables

definidas sobre dominios continuos: excesiva longitud de los cromosomas, falta de precisión, etc. En los últimos años, se ha estudiado ampliamente la codificación real [HLV98c], más adecuada para este tipo de problemas. 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.

Además de estos esquemas generales de representación, se pueden emplear muchos otros particulares al tipo de problema que se esté resolviendo, siempre que sea necesario.

B.3. 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 Algoritmos Genético en la siguiente generación. De este modo, si notamos por P la población actual formada por n cromosomas, C_1, \dots, 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 B.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' .

Existen diferentes formas de poner en práctica la selección [BS91]. Por ejemplo, puede establecerse un paralelismo entre la población y una ruleta, en la que 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. Este mecanismo de selección es uno de los

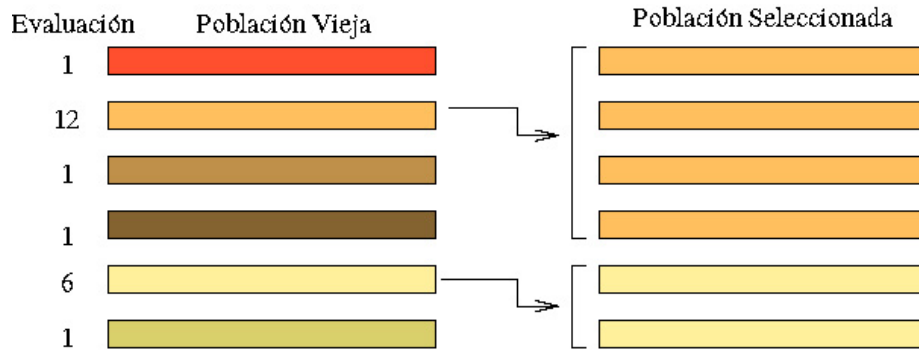


Figura B.2: Ejemplo de aplicación del mecanismo de selección

más conocidos y se denomina *muestreo estocástico con reemplazamiento*. Uno de los más eficientes es el *muestreo universal estocástico*, propuesto por Baker en [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].

B.4. 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*, P_c .

El operador de cruce juega un papel fundamental en el Algoritmos Genético. Su tarea es la de *explorar el espacio de búsqueda* refinando las soluciones obtenidas hasta el momento mediante la combinación de las buenas características que pre-

senten. 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 la figura B.3), o el *cruce multipunto*, que procede como el anterior pero trabajando sobre dos o más puntos de cruce.

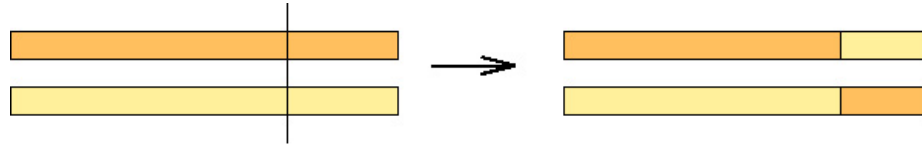


Figura B.3: Ejemplo de aplicación del operador de cruce simple en un punto

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 [HLV98c]. 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*, el cual será usado en capítulos posteriores. Dados dos cromosomas de la población $P(t)$, $C_v^t = (c_1, \dots, c_k, \dots, c_H)$ y $C_w^t = (c'_1, \dots, c'_k, \dots, c'_H)$, que van a ser cruzados, este operador genera los cuatro descendientes siguientes:

$$\begin{aligned} C_1^{t+1} &= aC_w^t + (1-a)C_v^t \\ C_2^{t+1} &= aC_v^t + (1-a)C_w^t \\ C_3^{t+1} \text{ con } c_{3k}^{t+1} &= \min\{c_k, c'_k\} \\ C_4^{t+1} \text{ con } c_{4k}^{t+1} &= \max\{c_k, c'_k\} \end{aligned}$$

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 Algoritmos Genético o variable dependiendo de la edad de la población.

B.5. 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. Todos los genes de los cromosomas existentes están sujetos a la posibilidad de mutar de acuerdo a una probabilidad de mutación P_m .

En este caso, 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 Algoritmos Genéticos 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 B.4.

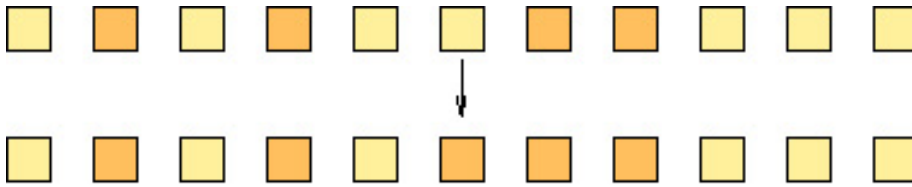


Figura B.4: Ejemplo de aplicación del operador de mutación

Este operador puede trasladarse al campo de los Algoritmos Genéticos 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 [HLV98c, Mic96]. De entre ellos destacaremos el operador de mutación *no uniforme* propuesto por Michalewicz en esta última referencia, el cual ha demostrado buen comportamiento en numerosas aplicaciones basadas en la codificación real. A continuación, describimos su modo de trabajo.

Dado un cromosoma de la población $P(t)$, $C_v^t = (c_1, \dots, c_k, \dots, c_H)$, y uno de sus genes, c_k , $k \in 1, \dots, H$, definido en $[c_{ki}, c_{kd}]$, seleccionado para ser mutado, el cromosoma obtenido tras la mutación presenta la forma $C_v^{t+1} = (c_1, \dots, c'_k, \dots, c_H)$, con

$$c'_k = \begin{cases} c_k + \Delta(t, c_{kd} - c_k), & \text{si } p = 0 \\ c_k - \Delta(t, c_k - c_{ki}), & \text{si } p = 1 \end{cases}$$

donde p es un número aleatorio generado en $\{0, 1\}$ y la función $\Delta(t, y)$ devuelve un valor en el intervalo $[0, y]$, de modo que la probabilidad de que $\Delta(t, y)$ sea cercana a 0 aumenta cuando lo hace el contador de generaciones t :

$$\Delta(t, y) = y(1 - r^{(1 - \frac{t}{T})^b})$$

donde, a su vez, r es un número aleatorio generado en $[0, 1]$, T es el número de generaciones durante las que se ejecutará el Algoritmos Genético y b es un parámetro escogido por el usuario que determina el grado de dependencia existente con respecto al número de generaciones. Esta propiedad da lugar a que el operador lleve a cabo una búsqueda uniforme en el espacio cuando t es pequeña, es decir, en las primeras iteraciones, y una mucho más localizada en generaciones posteriores.

Apéndice C

Sistemas Basados en Reglas Difusas Evolutivos

En los últimos años, la investigación en el campo de los Sistemas Difusos ha evolucionado hasta dar lugar a un marco de trabajo más general que contempla la integración de la Lógica Difusa con otras técnicas tales como los Algoritmos Evolutivos (AEs) [Bäc96], las Redes Neuronales y el Razonamiento Probabilístico. Esta nueva área recibe el nombre de *Computación Flexible*¹ [Bon97].

En concreto, la combinación de la Lógica Difusa y los AEs ha obtenido unos resultados muy prometedores en los últimos años [CHL96, CHL97a, CHL97b, CHHM01, HV96, Ped97, SSZ97, Tet95]. La definición automática de un Sistema Difuso se puede afrontar como un proceso de optimización o búsqueda, y los AEs, en particular los Algoritmos Genéticos, están considerados en la actualidad como la técnica de búsqueda global más conocida y empleada. Además, la codificación genética que emplean les permite incorporar conocimiento a priori de una forma muy sencilla y aprovecharlo para guiar la búsqueda. Todas estas razones han incrementado el empleo de los AEs para el diseño de Sistemas Difusos a lo largo de los últimos años, lo que ha dado lugar a la creación de los *Sistemas Difusos Evolutivos* [CH95, CHHM01, HM97].

Un Sistema Difuso Evolutivo es un método de diseño automático de Sistemas

¹El término castellano *Computación Flexible* es una traducción del término inglés *Soft Computing*

Difusos basado en AEs. Dentro de este gran grupo, los más conocidos y estudiados son los denominados SBRDEs [CH95, CHHM01, GH97, HM97], en los que se hace uso de los AEs para obtener de un modo automático la totalidad o una parte de la base de reglas difusas del SBRD (la figura C.1 ilustra esta idea). Existen distintos tipos según que la tarea de diseño que lleven a cabo sea la definición de las funciones de pertenencia de los conjuntos difusos de las reglas (la BD, en el caso de los SBRDs lingüísticos), la obtención de las propias reglas (la BR lingüística, en el caso de los SBRDs lingüísticos) o ambas a la vez (la BC completa en el caso de los SBRDs lingüísticos o la BRD en el caso de los SBRDs aproximativos).

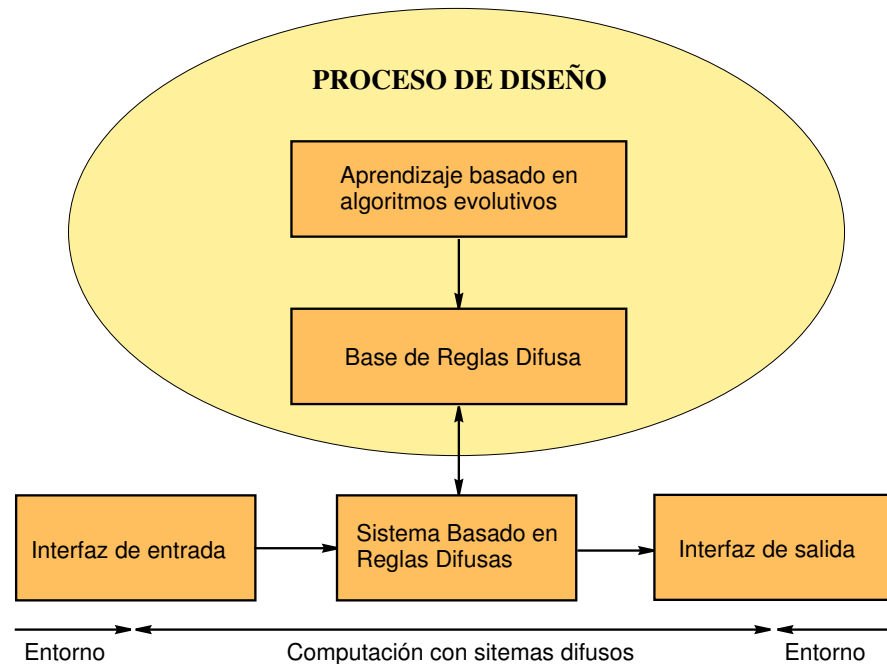


Figura C.1: Sistemas basados en reglas difusas evolutivos

Aunque los AEs no son algoritmos de aprendizaje, ofrecen una metodología de búsqueda potente e independiente del dominio que puede ser aplicada a gran cantidad de tareas de aprendizaje. De hecho, los AEs han sido muy utilizados en problemas de aprendizaje automático de reglas de producción a partir de conjuntos de ejemplos [Gre94]. Esto se debe a que los individuos, cuando representan conocimiento, se pueden tratar al mismo tiempo como datos que son manipulados

por el AE y como código ejecutable que lleva a cabo cierta tarea [Mic96].

Los AEs se han utilizado fundamentalmente para el aprendizaje de reglas, existiendo tres enfoques de aplicación. De ellos, dos son clásicos y han sido muy usados a lo largo de los últimos años, mientras que el tercero ha aparecido recientemente tratando de combinar las mejores características y de resolver los distintos problemas de los dos anteriores. Los dos primeros enfoques, *Michigan* y *Pittsburgh*, propuestos respectivamente en [Boo82, HR78] y [Smi80], se diferencian principalmente en el esquema de representación. Mientras que en el enfoque Michigan cada cromosoma codifica una única regla y la solución final al problema está formada por la última población obtenida por el AE al completo, en el enfoque Pittsburgh, los cromosomas corresponden a bases de reglas completas, por lo que la solución al problema de aprendizaje está compuesta finalmente por el mejor cromosoma de la última población. En cambio, en el tercero de los enfoques, el de *Aprendizaje Iterativo de Reglas* [CH97, GH97, Ven92], cambia un poco la filosofía de trabajo. Al igual que en el enfoque Michigan, cada cromosoma codifica una regla individual, con lo que el proceso evolutivo se hace a nivel de regla y no de base de reglas, pero el único individuo considerado para formar parte de la solución final es el mejor adaptado de la última población.

Para el caso de la BD de un SBRDE, los AEs se han utilizado fundamentalmente para adaptar las definiciones de los factores de escala y de las funciones de pertenencia, considerando definiciones parametrizadas de los mismos, además de para adaptar los contextos. Estos métodos parten de una BR lingüística ya derivada que no van a modificar. En este caso, no tiene sentido hablar de los enfoques anteriores ya que lo que se codifica en cada individuo del AE es la definición completa de la BD. Ejemplos de SBRDEs que ajustan la BD son [BKC96, GGP98, Gür99, HLV95, HS96, Kar91a, Kar91b, KG93, KKK94, LM94, Mag97, NL94].

El aprendizaje de la BR mediante SBRDEs asume la existencia de un conjunto de funciones de pertenencia en la BD al que las reglas se refieren mediante etiquetas lingüísticas. Algunos ejemplos de SBRDEs que aprenden la BR son [Bon96, CdJHL99, GIP99, HP97, INM99, PK91, Thr91, VR91].

Existen también otras propuestas de SBRDEs que aprenden la BC en su totalidad [BAB01, CFM96, CH97, CdH01, MMH97, MV97, PB93, PKL94, Vel98].

Aparte de los procesos evolutivos anteriores, existen algunas propuestas de SBRDEs para selección de características [CCdJH01, GP01], selección genética [CH97, GJ99, INYT95, IMT97, RS00], obtención de la BC mediante el aprendizaje evolutivo de la BD [Glo96, FJ96, IM96], aprendizaje genético considerando

estructuras del modelo diferentes (reglas de doble consecuente [CH00], reglas ponderadas [IT01], BCs jerárquicas [CHZ02], ...), etc.

Apéndice D

Problemas de Modelado Considerados

Este último apéndice contiene una descripción de los aspectos de experimentación considerados y de las aplicaciones de modelado empleadas en los diferentes estudios experimentales desarrollados a lo largo de esta memoria.

D.1. Aspectos Generales de la Experimentación

Con la intención de realizar un análisis comparativo objetivo hemos considerado un marco común en toda la experimentación realizada, definido por los siguientes aspectos:

- *Función objetivo* — En todos los métodos de búsqueda analizados hemos empleado el mismo criterio para valorar la calidad de las soluciones generadas, exceptuando aquellas situaciones donde el algoritmo considerado añade explícitamente penalizaciones a dicho criterio y el enfoque propuesto en la Sección 3.3 del Capítulo 3. El criterio seguido es minimizar el *error*

cuadrático medio sobre el conjunto de entrenamiento:

$$\text{ECM} = \frac{1}{2 \cdot N} \sum_{l=1}^N (F(x_1^l, \dots, x_n^l) - y^l)^2,$$

donde $F(x_1^l, \dots, x_n^l)$ es la salida obtenida a partir del SBRD cuando se considera x^l como entrada e y^l es la salida deseada conocida.

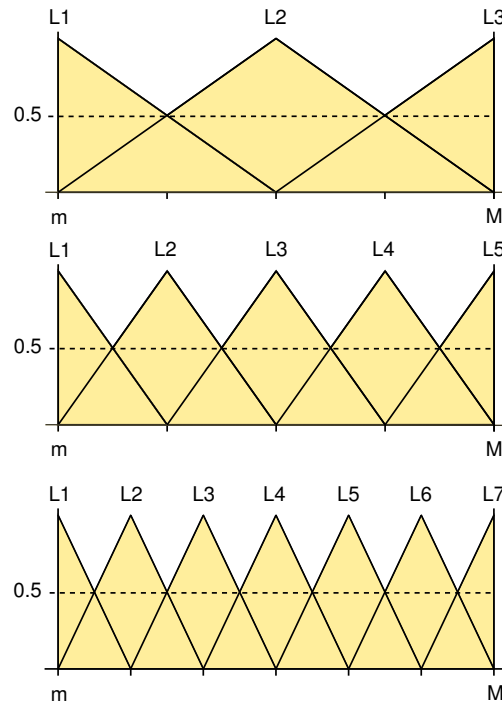


Figura D.1: Representación gráfica de posibles particiones difusas con 3, 5 o 7 etiquetas lingüísticas

- *BD inicial* — En todos los métodos de aprendizaje que precisen la definición de una BD inicial (por ejemplo, porque deriven la BR) se empleará la misma BD. Estará constituida por particiones difusas uniformemente distribuidas en cada variable. Dichas particiones se caracterizan por emplear funciones de pertenencia triangulares, tener la misma distancia entre los vértices de dos funciones de pertenencia consecutivas y ser la suma de los

grados de pertenencia a cada conjunto difuso igual a uno para cualquier valor del universo de discurso. El número de términos lingüísticos (etiquetas lingüísticas) y el universo de discurso de cada variable dependerá del problema y la técnica de aprendizaje empleados. En la figura D.1 se muestran tres posibles particiones difusas con diferente número de etiquetas lingüísticas.

- *Método de razonamiento de los SBRDs* — Emplearemos la t-norma del *mínimo* como operadores de conjunción e implicación, y la estrategia de defuzzificación en *Modo B* del *centro de gravedad ponderado por el grado de emparejamiento* (véase la sección A.5.4) [CHP97].

La única excepción la conforma el método de aprendizaje propuesto en la Sección 3.3 del Capítulo 3, que precisa de un método de razonamiento particular para acelerar su cálculo.

- *Modelo de experimentación* — Los experimentos se han realizado considerando un *modelo de validación cruzada de 5 particiones*, es decir, se han generado 5 particiones aleatorias a partir del conjunto de datos al 20% y se combinan cuatro de ellas (80%) para formar el conjunto de entrenamiento y la restante se utiliza para prueba, obteniéndose 5 particiones distintas. Para cada una de las 5 particiones cada método se ha ejecutado 6 veces, lo que supone 30 ejecuciones de cada método en total. En las tablas de resultados se muestran los resultados medios de las 30 ejecuciones de cada método. Además, se aplica un test *t-student* (con 95% de confianza) para comprobar si son significativas las diferencias que hay entre los resultados mostrados en las tablas.

D.2. Descripción de las Aplicaciones Consideradas

En esta memoria abordaremos dos problemas de distribución de energía eléctrica: la relación de algunas características de ciertos pueblos con la extensión actual de la línea de baja tensión contenidas en ellos y la relación del costo de mantenimiento de la red de media tensión instalada en ciertas ciudades con sus características. En ambos casos, es necesario que, además de que los modelos sean numéricamente precisos en el problema a resolver, sean capaces de explicar también cómo se calcula un valor específico para cierto pueblo o ciudad. Esto es, resulta interesante que estas soluciones sean interpretables por seres humanos en

Tabla D.1: Principales características de las aplicaciones consideradas en esta memoria

Aplicación	Complejidad	#V	#Entr	#Prue	#TL	#P
Longitud de líneas de baja tensión	alta	2	396	99	3/5/7/9	5
Costo de líneas de media tensión	media-alta	4	847	212	3/5/7/9	5

#V = número de variables de entrada, #Entr = tamaño del conjunto de entrenamiento, #Prue = tamaño del conjunto de prueba, #TL = número de términos lingüísticos considerados para cada variable (depende de la técnica aplicada), #P = número de particiones de ejemplos consideradas

cierto grado. La tabla D.1 recoge un resumen de las principales características de los dos problemas de distribución de energía eléctrica.

Además, trataremos la optimización de controladores difusos dedicados al control de sistemas HVAC (mejora del rendimiento energético manteniendo el confort). Este problema constituye un ejemplo de aplicación específico del ajuste de las funciones de pertenencia y de su combinación con la selección de reglas y, por lo tanto, sólo es usado en el Capítulo 3.

A continuación, describiremos todos tres problemas.

D.2.1. Estimación de la Longitud de las Líneas de Baja Tensión

Algunas veces existe la necesidad de medir la longitud de las líneas de electricidad que tiene una compañía eléctrica. Esta medida puede ser muy útil para varios aspectos tales como la estimación del costo de mantenimiento de la red, el cual constituyó el principal objetivo del problema presentado en España [CHS99, SSZ97]. Las líneas de alta y media tensión pueden ser fácilmente medidas, pero las de baja tensión —contenidas en ciudades o pueblos— resultan ser muy costosas de medir. Esta clase de línea es muy intrincada y en algunos casos una compañía puede servir más de 10.000 pequeños núcleos. En consecuencia, es necesario determinar un método indirecto para calcular la longitud de dichas líneas.

Por esta razón, se tuvo que encontrar una relación entre algunas características de la población y la longitud de la línea instalada en ella, haciendo uso de algunos

Tabla D.2: Notación y universos de discurso de las variables consideradas para el problema de la estimación de la longitud de las líneas de baja tensión

<i>Símbolo</i>	<i>Significado</i>	<i>Universo de discurso</i>
X_1	Número de clientes en la población i	[1, 320]
X_2	Radio de la población i	[60, 1673,329956]
Y	Longitud de la línea en la población i	[80, 7675]

datos existentes, que pudiera ser utilizada para predecir la longitud real de la línea en cualquier otro pueblo. En esta memoria trataremos de solucionar este problema generando diferentes clases de modelos lingüísticos y difusos para determinar la relación desconocida. Para ello, se cuenta con la longitud de la línea medida, el número de habitantes y la distancia media desde el centro de la ciudad hasta los tres clientes más lejanos, considerada como el radio de la población en una muestra de 495 núcleos rurales [SSZ97, Sán00]. En la tabla D.2 se indica el nombre y universo de discurso de cada variable.

Para comparar los distintos métodos de derivación considerados en la memoria, hemos dividido la muestra en dos conjuntos compuestos por 396 y 99 ejemplos, denominados de entrenamiento y de prueba, respectivamente. La dependencia que existe entre las dos variables de entrada y la de salida en ambos conjuntos se muestra en la figura D.2.

D.2.2. Estimación del Costo de Mantenimiento de la Línea de Media Tensión

Estimar los costos de mantenimiento de la red eléctrica de media tensión en un ciudad [CHS99] es un problema complejo pero, a la vez, muy interesante. Dado que es muy difícil obtener una medición real, la consideración de modelos resulta tremendamente útil. Estas estimaciones permiten a las compañías eléctricas justificar sus gastos. Además, el modelo debe poder explicar cómo se calcula un valor específico para una determinada ciudad.

Nuestro objetivo será el de relacionar el *costo de mantenimiento* con cuatro características determinadas aplicando las técnicas de modelado consideradas en esta memoria (la denominación y universo de discurso de cada variable se muestra en la tabla D.3). Para ello disponemos de estimaciones de costos de mantenimiento

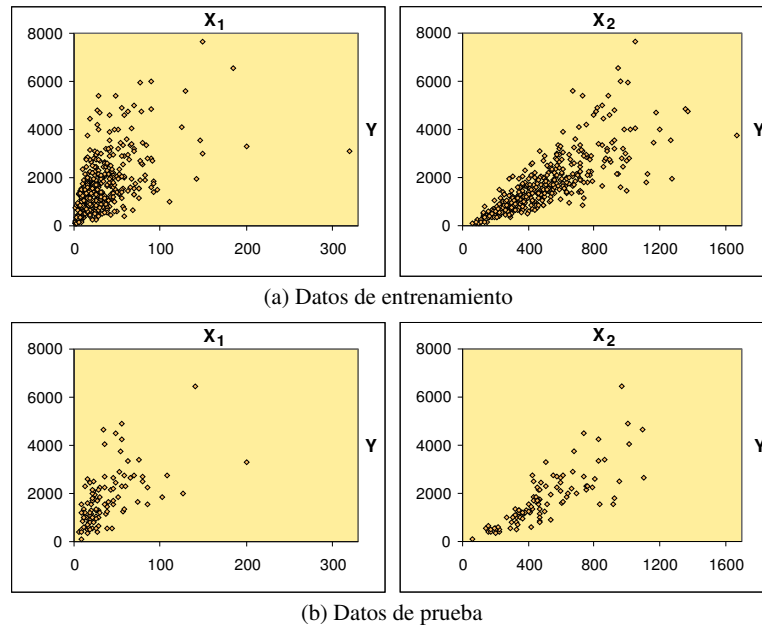


Figura D.2: (a) Dependencia (X_1, Y) y (X_2, Y) en el conjunto de entrenamiento; (b) Dependencia (X_1, Y) y (X_2, Y) en el conjunto de prueba

Tabla D.3: Notación y universos de discurso de las variables consideradas para el problema de mantenimiento de las líneas de media tensión

<i>Símbolo</i>	<i>Significado</i>	<i>Universo de discurso</i>
X_1	Suma de las longitudes de todas las calles de la ciudad	$[0, 11]$
X_2	Área total de la ciudad	$[0,15, 8,55]$
X_3	Área ocupada por edificios	$[1,64, 142,5]$
X_4	Energía suministrada a la ciudad	$[1, 165]$
Y	Costo de mantenimiento de la línea de media tensión	$[0, 8546,030273]$

basadas en un modelo de una red eléctrica óptima para cada ciudad en una muestra de 1.059 ciudades.

La muestra se ha dividido al azar en dos conjuntos de 847 y 212 ejemplos, 80 y 20 por ciento del conjunto de datos completo, denominados conjunto de entrenamiento y de prueba, respectivamente.

D.2.3. Sistemas de Ventilación, Calefacción y Aire Acondicionado (*HVAC Systems*)

Un sistema HVAC está compuesto por todas las partes electromecánicas del edificio usadas para acondicionar el aire interior del mismo. El sistema HVAC es necesario para proporcionar a los trabajadores un entorno de trabajo cómodo y productivo que satisfaga sus necesidades fisiológicas. Por lo tanto, un sistema de climatización HVAC debería realizar dos tareas principales:

- diluir y eliminar las emisiones de personas, equipos y actividades, suministrando aire limpio (Calidad del Aire Interior); y
- mantener la calidad térmica en verano e invierno (Clima Térmico),

de una manera eficiente (mínimo consumo de energía), silenciosa y con un bajo coste del ciclo de vida.

No hay datos estadísticos recopilados sobre tipos y tamaños de sistemas HVAC relativos a cada tipo de edificio en los diferentes países europeos. Por lo tanto, proporcionar un sistema HVAC compatible con el entorno es una tarea del diseñador de BEMS basada en su propia experiencia. En la Figura D.3, se presenta la estructura de un sistema HVAC típico para oficinas. Este sistema consta del siguiente conjunto de componentes capaces de aumentar y disminuir la temperatura y la humedad relativa del aire suministrado:

- A. Este modulo mezcla el aire del exterior con el aire de retorno para obtener el aire suministrado y también cierra el regulador del aire exterior cuando se para el ventilador.
- B. Un filtro para reducir la emisión del aire del exterior al aire suministrado.
- C. Un precalentador que calienta el aire suministrado y recupera energía del aire de salida.

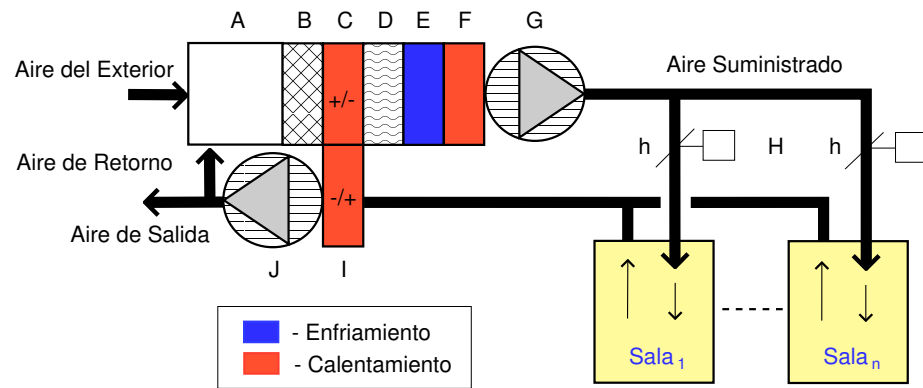


Figura D.3: Estructura genérica de un sistema HVAC para un edificio de oficinas

- D. Un humidificador para aumentar la humedad relativa en invierno.
- E. Un refrigerador para reducir la temperatura del aire suministrado y/o la humedad.
- F. Una unidad de post-calentamiento para aumentar la temperatura del aire después de humidificarlo o después del enfriamiento colateral (dehumidificador).
- G. El ventilador del aire suministrado.
- H. Los reguladores para demandar un flujo de aire controlado a las habitaciones.
- I. Una unidad de recuperación de calor para la obtención de energía a partir del aire de salida.
- J. El ventilador del aire de salida.

D.2.3.1. El Problema del Control de Sistemas HVAC

La temperatura y la humedad relativa son factores esenciales para satisfacer las necesidades fisiológicas de los ocupantes del edificio (trabajadores). Cuando la temperatura está por encima o por debajo de la zona de confort, el entorno trastorna los procesos metabólicos de las personas e interrumpe sus actividades.

Por lo tanto, el diseño correcto y el control adecuado del sistema de climatización HVAC es esencial para mantener un entorno funcional y habitable, mejorando el confort de los ocupantes de un edificio. Los sistemas anticuados, inadecuados o mal utilizados dan lugar a reclamaciones por falta de confort, problemas con la calidad del aire, dificultad en el control y costos excesivos en el uso. Además, muchos sistemas de climatización HVAC no mantienen una temperatura uniforme en todo el edificio, debido al empleo de algoritmos de control poco sofisticados. En un edificio inteligente moderno, un control sofisticado del sistema debería proporcionar un excelente control ambiental [AHK95, HN94].

Dentro de este marco de trabajo (automatización de edificios), el objetivo de un controlador global es mantener el ambiente dentro de los límites deseados (o estipulados). En nuestro caso, mantener las condiciones ambientales dentro de la zona de confort y controlar la calidad del aire interior. Además, se requieren otros objetivos importantes, por ejemplo, ahorro de energía (nuestro principal objetivo), estabilidad del sistema, etc. En cualquier caso, deben ser considerados numerosos factores para conseguir dichos objetivos. Esto hace que el sistema controlado sea muy complejo y provoca una fuerte no linealidad.

Para obtener un controlador óptimo, los parámetros de control y los parámetros controlados ¹ deben seleccionarse de acuerdo a la estrategia de control implementada, a su viabilidad técnica y de acuerdo a consideraciones económicas. Afortunadamente, el diseñador del BEMS normalmente es capaz de determinar dichos parámetros.

En las siguientes secciones, se presentan los parámetros de control y controlados más comúnmente usados. Los parámetros específicos considerados en el módulo de pruebas presentado en esta memoria se seleccionarán de entre éstos en la Sección D.2.3.3, donde se introduce dicho módulo.

D.2.3.1.1. Parámetros de Control o Explícitos: Variables del Controlador Dependiendo del tipo de sistema de climatización HVAC, de los sensores y de los actuadores utilizados, se pueden considerar distintos parámetros (de control o explícitos) para identificar las variables del controlador difuso. Usualmente, estos parámetros se seleccionan de entre los siguientes:

¹Los parámetros de control o explícitos son variables que podrían usarse como entradas o salidas de una estrategia de control (variables del controlador), mientras que los parámetros controlados o implícitos son las variables afectadas por la acción de un dispositivo controlado y podrían considerarse para evaluar el rendimiento de tal controlador (objetivos del problema)

- *Índice de confort térmico PMV*²: En vez de considerar únicamente la temperatura del aire como índice de confort térmico, podríamos considerar el índice global de confort térmico ISO 7730³, más conocido como índice de confort térmico PMV, que incorpora la humedad relativa y la temperatura media irradiada.
- *Diferencia entre la temperatura suministrada y la existente*: Algunas perturbaciones pueden estar relacionadas con la diferencia de temperatura entre el aire suministrado y el aire existente. Cuando los sistemas de ventilación se usan para acondicionar el aire, tal criterio puede ser importante.
- *Concentración de CO₂*: Se ha descubierto que la calidad del aire interior es un factor crítico a tener en cuenta. Puesto que la concentración de CO₂ es una medida segura de la contaminación emitida por los ocupantes de un edificio, puede seleccionarse como índice de calidad del aire. Esto sólo es cierto si suponemos que tanto el edificio como el sistema de climatización HVAC han sido correctamente diseñados y que los ocupantes son realmente la principal fuente de contaminación.
- *Temperatura exterior*: La temperatura exterior también debe ser considerada, puesto que durante la primavera y el otoño (o incluso durante las mañanas de verano), el potencial de enfriamiento (o calentamiento) mediante la ventilación puede ser importante y, por lo tanto, puede reducir la necesidad de aplicar un enfriamiento (o calentamiento) artificial.
- *Actuadores del sistema HVAC*: Dependen directamente del sistema de climatización HVAC, por ejemplo, posiciones de válvulas, modos de operación, velocidades del ventilador, etc.

D.2.3.1.2. Parámetros Controlados o Implícitos: Objetivos del Problema

Dependiendo de los objetivos de la estrategia de control, se pueden considerar distintos parámetros (controlados o implícitos) para identificar los índices globales que evalúan el ambiente creado en el edificio. En este tipo de problemas, dichos parámetros podrían seleccionarse de entre los siguientes:

²Del inglés, *Predicted Mean Vote*

³<http://www.iso.org/iso/en/ISOOnline.frontpage>

- **Confort térmico:** El control de las condiciones climáticas del interior del edificio es uno de los objetivos más importantes de los edificios inteligentes. Entre las principales características del clima interior de un edificio, el confort térmico es la de mayor importancia. Este índice podría incluir parámetros de confort global y local.
- **Calidad del aire:** La calidad del aire también tiene gran interés en los edificios modernos. Este factor puede controlarse ya sea en la fase de diseño mediante la reducción de posibles contaminantes, o en la fase de funcionamiento gracias a los sistemas de ventilación. Como nuestro trabajo está dedicado a los sistemas de climatización HVAC, la calidad del aire también será un parámetro importante a tener en cuenta.
- **Consumo de energía:** Aunque la calidad del aire y los niveles de confort térmico deben estar garantizados, ésto se debe conseguir con un mínimo coste de energía. Por lo tanto, debería considerarse algún parámetro para la medida del consumo de energía.
- **Estado del sistema de climatización HVAC:** Para aumentar el tiempo de vida y reducir los costos de mantenimiento de los equipos controlados, se necesita un funcionamiento estable de los mismos. Por lo tanto, debe considerarse la información del estado de los equipos en el momento de tomar una decisión o durante un periodo de tiempo mayor.
- **Clima exterior:** Las condiciones climáticas en el interior de un edificio son influenciadas por las condiciones climáticas del exterior (temperatura del aire, radiación solar, viento, etc.). Además, en los sistemas HVAC de distribución de aire, la energía requerida para aumentar o disminuir la temperatura suministrada es una función de la temperatura y la humedad exteriores. Así, algunos de estos parámetros necesitarían ser seleccionados.

D.2.3.2. Control Difuso de Sistemas HVAC

Hoy en día, los controladores difusos se han empleado con éxito en multitud de aplicaciones del mundo real, como por ejemplo, sistemas de suspensión inteligente, navegación de robots móviles, control de transformadores de energía eólica, control de aparatos de aire acondicionado, enfoque automático y estabilizador de imagen de cámaras de fotografía y video, control de anti-balanceo de grúas y muchas aplicaciones industriales de automatización [Hir93]. En la Figura D.4 se

muestra la estructura genérica de este tipo de controladores (véase la Sección A.5 del Apéndice A para obtener una explicación detallada de los componentes de este tipo de SBRDs).

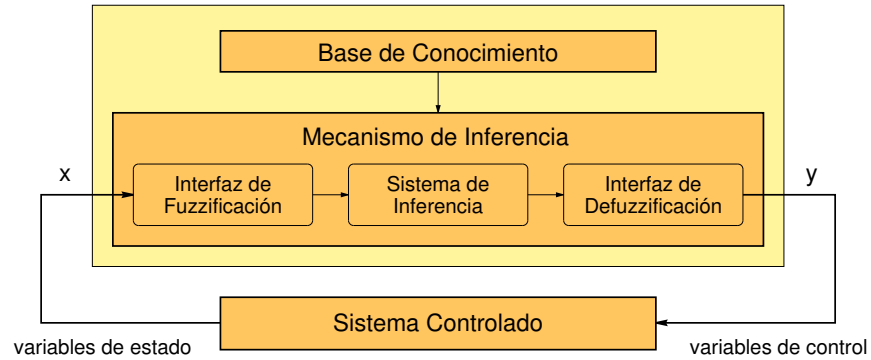


Figura D.4: Estructura genérica de un controlador difuso

En el caso específico de los sistemas de climatización HVAC, la mayoría de los trabajos aplican los controladores difusos para resolver problemas simples, como puede ser el de regulación térmica, manteniendo la temperatura en un punto concreto [AHK95, Glo91, HN94]. Sin embargo, en nuestro caso se deben considerar varios criterios con el objetivo de reducir el consumo de energía manteniendo el nivel de confort deseado. Por lo tanto, deberemos considerar muchas variables del sistema a ser controlado, lo que complica mucho el problema.

Usualmente, en este tipo de problemas (diseño de controladores para sistemas de climatización HVAC), la construcción de la BC se basa en la información recopilada a partir de la experiencia de los operadores. Sin embargo, el controlador difuso no siempre consigue resultados satisfactorios con un conjunto inicial de reglas definido a partir de expertos [HN94]. En nuestro caso, el sistema controlado es demasiado complejo y necesita controladores difusos óptimos que mejoren el rendimiento del sistema. Por lo tanto, estos enfoques necesitarían de una modificación de la BC inicial para obtener un controlador óptimo.

Como se dijo en el Capítulo 1, una forma de aumentar el rendimiento del controlador difuso sin perder su interpretabilidad en un alto grado es mejorar la forma en que el modelo difuso lingüístico realiza el razonamiento interpolativo, induciendo una mejor cooperación entre las reglas de su base. En la literatura especializada se pueden encontrar diferentes posibilidades para mejorar el MDL [CCHM03b]. Estos enfoques también pueden aplicarse en el campo

de trabajo del control difuso (por ejemplo, modificando los componentes de la BD [ACG99, ACC⁺01, ?, ABC⁺03a], puede realizarse un ajuste de la semántica de un controlador difuso inicial obtenido a partir de la experiencia). En nuestro caso, se propone mejorar un controlador difuso inicial obtenido a partir de expertos considerando el ajuste-LA y la combinación del ajuste-LA con la selección del subconjunto de reglas que presente la mejor cooperación, mediante la aplicación de técnicas de aprendizaje automático.

Por otro lado, para evaluar el rendimiento del controlador difuso, usualmente es necesaria una modelización física de los edificios controlados. Estos modelos han sido desarrollados por los diseñadores de BEMSs utilizando herramientas para la simulación de edificios y están capacitados para tener en cuenta todos los parámetros considerados en el proceso de control. Así, utilizando el sistema simulado, tendremos la posibilidad de evaluar el controlador diseñado con las condiciones ambientales deseadas. De la misma forma, el sistema simulado podrá ser utilizado por los expertos para validar la BC inicial antes de aplicar el proceso de optimización automático. Además, el simulador obtenido es de suma importancia para calcular el valor de la función de adaptación en dicho proceso.

D.2.3.3. El Módulo de Pruebas de ATC

Dentro del marco de trabajo del programa *JOULE-THERMIE*, bajo el proyecto GENESYS antes mencionado, una empresa privada cuyo nombre debe permanecer anónimo proporcionó un módulo de pruebas. A partir de ahora, este lugar será llamado módulo de pruebas de ATC —del inglés, *Anonymous Test Cell*—.

Situado en Francia, este entorno de prueba consta de siete módulos alrededor de los cuales se puede recrear un clima artificial en cualquier momento (las condiciones ambientales de invierno pueden simularse en verano y *viceversa*). La Figura D.5 ilustra este entorno y presenta sus principales características. Se dispuso de dos módulos gemelos adyacentes para los experimentos, el número 4 y el 5. Ambos módulos fueron equipados con todos los sensores y actuadores necesarios de acuerdo a los parámetros de control y controlados. El sistema de climatización HVAC utilizado era una unidad de aire refrigerado provista de una bomba de calor de inversión de ciclo (para la recuperación de calor) y un ventilador con un sistema mecánico de variación de velocidad (para el sistema de ventilación/extracción).

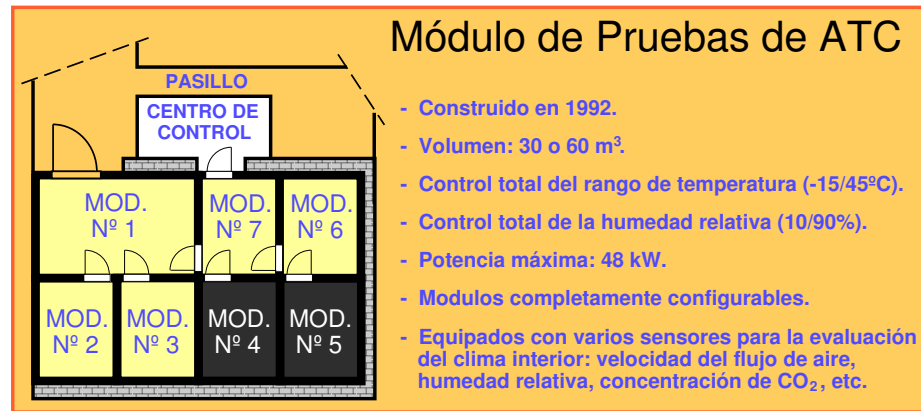


Figura D.5: Representación y principales características de los módulos de ATC

Lo primero fue desarrollar el modelo térmico de los módulos de prueba. El mayor logro fue el desarrollo de un modelo de edificio monozona completo. El modelo fue construido a partir de cero en el entorno de simulación *Matlab-Simulink* y desarrollado como un modelo de propósito general que pudiera usarse con diferentes condiciones climáticas, proyectos o aplicaciones en el futuro. Sin embargo, después fue adaptado para ajustarse a los módulos de pruebas de ATC para mejorar su funcionamiento. La simulación térmica se basó en métodos de diferencias finitas para el modelo de conductividad. El valor máximo para el tiempo de paso de la simulación se calculó usando la condición de estabilidad de acuerdo al esquema de discretización. El tiempo de paso de simulación podía reducirse hasta 60 segundos. Debido al pequeño grosor y a la gran conductividad térmica de las ventanas, el modelo de conductividad del calor para las ventanas se consideró constante. Los cambios de calor por convección se basaron en coeficientes constantes de convección del calor. La temperatura irradiada se calculó como una función de su superficie, ponderada por su área relativa. El modelo del sistema de climatización HVAC se basó en los datos de los fabricantes y los módulos desarrollados en el campo de la *tarea 22 de la IEA (International Energy Agency)*. Dicha información fué suministrada por el *Royal Technical Institute* de Estocolmo.

Los datos disponibles fueron usados para calibrar el modelo. Los principales problemas en el calibrado fueron relativos al modelado del equipo HVAC así como a los efectos de la radiación solar en la ganancia del calor interior. *La experimentación de este trabajo ha sido realizada considerando el modelo de simulación*

de ATC calibrado y validado. Concretamente, el modelo de verano de ATC.

D.2.3.3.1. Objetivos y Función de Adaptación

Como se ha dicho, **el principal objetivo que nos planteamos es mejorar el rendimiento energético manteniendo el nivel de confort deseado**. Por lo tanto, deberíamos considerar el desarrollo de una función de adaptación que represente el rendimiento de cada controlador mediante el confort térmico, la calidad del aire interior, el consumo de energía y la estabilidad del sistema. Para ello, el objetivo global es **minimizar** los siguientes cinco criterios:

O₁: Límite de confort térmico superior:

$$\text{si } PMV > 0,5, O_1 = O_1 + (PMV - 0,5) .$$

O₂: Límite de confort térmico inferior:

$$\text{si } PMV < -0,5, O_2 = O_2 + (-PMV - 0,5) .$$

O₃: Requisito de calidad del aire:

$$\text{si } CO_2 \text{ conc. } > 800ppm, O_3 = O_3 + (CO_2 - 800) .$$

O₄: Consumo de energía:

$$O_4 = O_4 + \text{potencia en el tiempo } t .$$

O₅: Estabilidad del sistema:

$$O_5 = O_5 + \text{'cambios del sistema' desde el tiempo } t \text{ al } (t - 1) ,$$

donde 'cambio del sistema' representa un cambio de funcionamiento, por ejemplo, un cambio de la velocidad del ventilador o de la posición de una válvula.

En nuestro caso, los cinco criterios se combinan en una función objetivo global por medio de un vector de pesos. Esta técnica (suma ponderada de objetivos) tiene mucha sensibilidad y dependencia respecto a los pesos. Sin embargo, cuando

se dispone de pesos fiables, este enfoque reduce el espacio de búsqueda proporcionando la dirección de búsqueda más adecuada y, por lo tanto, es recomendable. Puesto que los expertos obtuvieron pesos fiables en el marco del proyecto GENESYS, este será el enfoque que seguiremos en esta memoria.

Por lo tanto, asignar los pesos apropiados para cada criterio en la función de adaptación supuso un logro importante. La idea básica de esta definición de pesos fue la de encontrar equivalencias económicas para todos ellos. Dichas equivalencias son difíciles de establecer debido, entre otras cosas, a la falta de datos de confianza en este tópico. Mientras que el coste del consumo de energía es fácil de determinar, los criterios de confort son más difíciles. Estudios recientes han mostrado que un 18% de mejora en la satisfacción de las personas respecto al clima interior se corresponde con un 3% de mejora en la productividad de los trabajadores de una oficina. Basándonos en el salario típico y debido al hecho de que el *PMV* y la concentración de *CO₂* están relacionados con la satisfacción de las personas, podemos establecer dichas equivalencias. La misma estrategia se puede aplicar al criterio de la estabilidad del sistema, teniendo en cuenta el tiempo de vida de diferentes sistemas relacionado con el número de operaciones. De esta manera, se pueden obtener los pesos para cada edificio específico (o módulo de prueba).

En nuestro caso, los expertos proporcionaron pesos fiables para la función de evaluación del módulo de pruebas de ATC aplicando las equivalencias anteriores: $w_1^O = 0.0083022$, $w_2^O = 0.0083022$, $w_3^O = 0.00000456662$, $w_4^O = 0.0000017832$ y $w_5^O = 0.000761667$. Finalmente, la función objetivo a minimizar se calcula como:

$$F = \sum_{i=1}^5 w_i^O \cdot O_i .$$

D.2.3.3.2. Variables y Arquitectura del Controlador Difuso

El experto en BEMS propuso una arquitectura jerárquica de controlador difuso que consideraba el *PMV*, la concentración de *CO₂*, el estado previo del sistema HVAC y la temperatura exterior. Dicha arquitectura, las variables y la BR inicial se muestran en la Figura D.6.

La BD está compuesta por particiones difusas simétricas con funciones de pertenencia triangulares etiquetadas desde *L1* a *Ll_i*, siendo *l_i* el número de funciones de pertenencia de la *i*-ésima variable. La Figura D.7 representa dicha BD. Tanto la BR inicial como la BD fueron suministradas por el experto en BEMS.

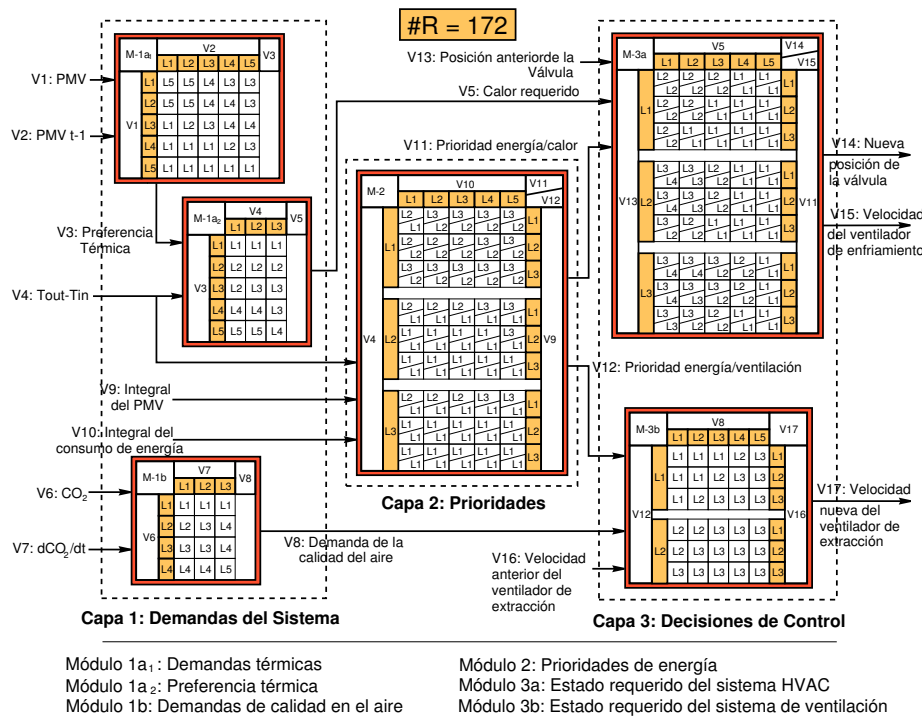


Figura D.6: BR inicial y estructura del controlador difuso de verano de ATC

Véase que la Figura D.6 representa las tablas de decisión de cada módulo del controlador jerárquico en términos de dichas etiquetas. Cuando la BR consta de más de dos variables de entrada (como en el caso de los módulos M-2, M-3a y M-3b que consideran tres variables de entrada), la tabla tridimensional se descompone en tres tablas de decisión bidimensionales (una para cada posible etiqueta de la primera variable) con el principal objetivo de mostrar más claramente su composición. Por lo tanto, cada celda de la tabla representa un subespacio/regla difuso/a y contiene el consecuente(s) de salida asociado, es decir, la etiqueta(s) correspondiente. Las variables de salida se indican en la esquina superior derecha para cada módulo. Téngase en cuenta que, cuando hay dos consecuentes, aparecen en la misma celda separados por una línea diagonal.

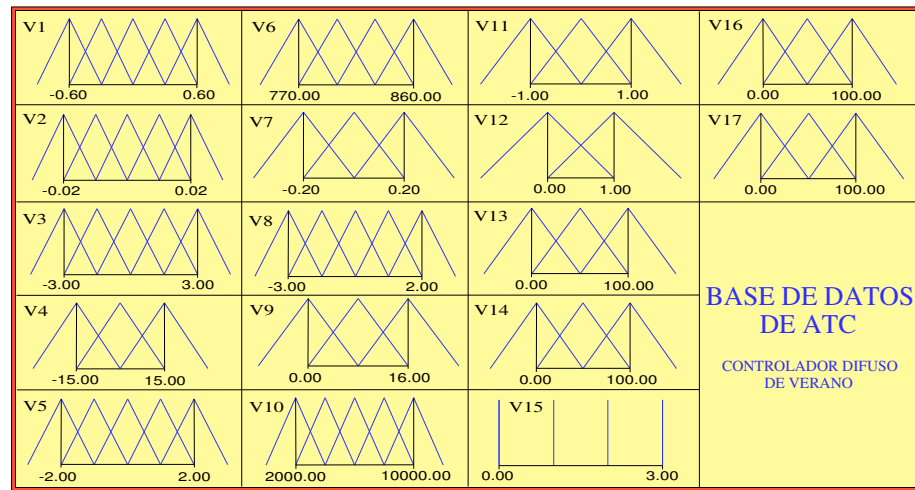


Figura D.7: BD del controlador difuso de verano de ATC

Bibliografía

- [AAFB⁺06] Alcalá R., Alcalá-Fdez J., Berlanga F., Gacto M. J. y Herrera F. (2006) Improving fuzzy rule-based decision models by means of a genetic 2-tuples based tuning and the rule selection. En Torra V., Narukawa Y., Valls A. y Domingo-Ferrer J. (Eds.) *Proc. Third International Conference on Modeling Decisions for Artificial Intelligence (MDAI06)*, number 3885 in Lecture Notes in Computer Science, páginas 317–328. Springer-Verlag, Berlin.
- [AAFGH06] Alcalá R., Alcalá-Fdez J., Gacto M. J. y Herrera F. (2006) Fuzzy rule reduction and tuning of fuzzy logic controllers for a hvac system. En Kahraman C. (Ed.) *Fuzzy Set Techniques in Industrial Engineering*, International Series Studies in Fuzziness and Soft Computing. Springer-Verlag, Berlin. Por aparecer.
- [ABC⁺03a] Alcalá R., Benítez J. M., Casillas J., Castro J. L., Cordon O., González A., Herrera F. y Pérez R. (2003) Multicriteria genetic tuning for the optimization and control of HVAC systems. En Yu X., Kacprzyk J. y Carlsson C. (Eds.) *Applied Decision Support with Soft Computing*, Studies in Fuzziness and Soft Computing, páginas 308–345. Springer-Verlag, Heidelberg, Alemania.
- [ABC⁺03b] Alcalá R., Benítez J. M., Casillas J., Cordon O. y Pérez R. (2003) Fuzzy control of HVAC systems optimized by genetic algorithms. *Applied Intelligence* 18: 155–177.
- [ACC⁺00] Alcalá R., Casillas J., Cordon O., Herrera F. y Zwir I. (2000) Learning and tuning fuzzy rule-based systems for linguistic modeling. En Leondes C. T. (Ed.) *Knowledge-Based Systems: Techniques*

- and Applications*, volumen 3, páginas 889–941. Academic Press, San Diego/Londres, EE.UU./Reino Unido.
- [ACC⁺01] Alcalá R., Casillas J., Castro J. L., González A. y Herrera F. (2001) A multicriteria genetic tuning for fuzzy logic controllers. *Mathware and Soft Computing* 8(2): 179–201.
- [ACC⁺05] Alcalá R., Casillas J., Cordon O., González A. y Herrera F. (2005) A genetic weighted rule and rule selection process for fuzzy control of heating, ventilating and air conditioning systems. *Engineering Applications of Artificial Intelligence* 18(3): 279–296.
- [ACCH01a] Alcalá R., Casillas J., Cordon 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.
- [ACCH01b] Alcalá R., Casillas J., Cordon O. y Herrera F. (2001) Improvement to the cooperative rules methodology by using the ant colony system algorithm. *Mathware & Soft Computing* 8(3): 321–335.
- [ACG99] Alcalá R., Casillas J. y González A. (1999) Tuning fuzzy logic controllers for energy efficiency consumption in buildings. En *Proceedings of the 1999 Eusflat-Estylf Joint Conference*, páginas 103–106. Palma de Mallorca, España.
- [AH04] Alcalá R. y Herrera F. (2004) Genetic tuning on fuzzy systems based on the linguistic 2-tuples representation. En *Proceedings of the 13th IEEE International Conference on Fuzzy Systems*, volumen 1, páginas 233–238. Budapest, Hungría.
- [AHK95] Arima M., Hara E. H. y Katzberg J. D. (1995) A fuzzy logic and rough sets controller for HVAC systems. En *Proceedings of the IEEE (WESCANEX'95)*, volumen 1, páginas 133–138. Nueva York, NY, EE.UU.
- [Bab98] Babuška R. (1998) *Fuzzy modeling for control*. Kluwer Academic, Norwell, MA, 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.
- [Bas94] Bastian A. (1994) How to handle the flexibility of linguistic variables with applications. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 3(4): 463–484.
- [BB00] Bodenhofer U. y Bauer P. (2000) Towards an axiomatic approach to interpretability. En *Proceedings of the 6th International Conference on Soft Computing*, páginas 334–339. Iizuka, Japón.
- [BB03] Bodenhofer U. y Bauer P. (2003) A formal model of interpretability of linguistic variables. En Casillas J., Cordon O., Herrera F. y Magdalena L. (Eds.) *Interpretability issues in fuzzy modeling*, páginas 524–545. Springer-Verlag.
- [BBM93] Beasley D., Bull D. R. y Martin R. R. (1993) A sequential niche technique for multimodal function optimization. *Evolutionary Computation* 1(2): 101–125.
- [BD95] Bardossy A. y Duckstein L. (1995) *Fuzzy rule-based modeling with application to geophysical, biological and engineering systems*. CRC Press.
- [BH99] Berthold M. R. y Huber K. P. (1999) Constructing fuzzy graphs from examples. *Intelligent Data Analysis* 3: 37–53.
- [BKC96] Bonissone P. P., Khedar P. S. y Chen Y.-T. (1996) Genetic algorithms for automated tuning of fuzzy controllers, a transportation application. En *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'96)*, páginas 674–680. Nueva Orleans, LA, EE. UU.
- [Bod00] Bodenhofer U. (2000) A similarity-based generalization of fuzzy orderings preserving the classical axioms. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 8(5): 593–610.

- [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.
- [Bon97] Bonissone P. P. (1997) Soft computing: the convergence of emerging reasoning technologies. *Soft Computing* 1(1): 6–18.
- [Boo82] Booker L. B. (1982) *Intelligent behaviour as an adaption to the task environment*. PhD thesis, University of Michigan.
- [BOOB02] Babuška R., Oosterhoff J., Oudshoorn A. y Bruijn P. M. (2002) Fuzzy self-tuning PI control of pH in fermentation. *Engineering Applications of Artificial Intelligence* 15(1): 3–15.
- [BS91] Bäck T. y Schwefel H. P. (1991) Extended selection mechanisms in genetic algorithms. En *Proc. Fourth International Conference on Genetic Algorithms (ICGA'91)*, páginas 2–9. San Diego, EE.UU.
- [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.
- [CCdJH01] Casillas J., Cordon O., del Jesús M. J. y Herrera F. (2001) Genetic feature selection in a fuzzy rule-based classification system learning process for high dimensional problems. *Information Sciences* 136: 169–191.
- [CCdJH05] Casillas J., Cordon O., del Jesus M. J. y Herrera F. (2005) Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction. *IEEE Transactions on Fuzzy Systems* 13(1): 13–29.
- [CCdVH05] Casillas J., Cordon O., de Viana I. F. y Herrera F. (2005) Learning cooperative linguistic fuzzy rules using the best-worst ant systems algorithm. *International Journal of Intelligent Systems* 20: 433–452.
- [CCH02] Casillas J., Cordon 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., Cordon O., Herrera F. y Magdalena L. (Eds.) (2003) *Accuracy improvements in linguistic fuzzy modeling*, volumen 129 of *Studies in Fuzziness and Soft Computing*. Springer-Verlag, Heidelberg, Alemania.
- [CCHM03b] Casillas J., Cordon O., Herrera F. y Magdalena L. (2003) Accuracy improvements to find the balance interpretability-accuracy in linguistic fuzzy modeling: an overview. En Casillas J., Cordon O., Herrera F. y Magdalena L. (Eds.) *Accuracy Improvements in Linguistic Fuzzy Modeling*, volumen 129 of *Studies in Fuzziness and Soft Computing*, páginas 3–24. Springer-Verlag, Heidelberg, Alemania.
- [CCHM03c] Casillas J., Cordon O., Herrera F. y Magdalena L. (Eds.) (2003) *Interpretability issues in fuzzy modeling*, volumen 128 of *Studies in Fuzziness and Soft Computing*. Springer-Verlag, Heidelberg, Alemania.
- [CCHV04] Casillas J., Cordon O., Herrera F. y Villar P. (2004) A hybrid learning process for the knowledge base of a fuzzy rule-based system. En *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, volumen 3, páginas 2189–2196. Perugia, Italy.
- [CCZ04] Carmona P., Castro J. y Zurita J. (2004) Strategies to identify fuzzy rules directly from certainty degrees: A comparison and a proposal. *IEEE Transactions on Fuzzy Systems* 12(5): 631–640.
- [CdH01] Cordon 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.
- [CdJH98] Cordon O., del Jesus M. J. y Herrera F. (1998) Genetic learning of fuzzy rule-based classification systems cooperating with fuzzy reasoning methods. *International Journal of Intelligent Systems* 13(10–11): 1025–1053.
- [CdJHL99] Cordon 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.

- [CDM99] Castro J. L., Delgado M. y Mantas C. J. (1999) A hybrid system for approximate reasoning. Technical report, Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada.
- [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.
- [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.
- [CH95] Cordón O. y Herrera F. (1995) A general study on genetic fuzzy systems. En Periaux J., Winter G., Galán M. y Cuesta P. (Eds.) *Genetic Algorithms in Engineering and Computer Science*, páginas 33–57. John Wiley and Sons.
- [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.
- [CH00] Cordón O. y Herrera F. (2000) A proposal for improving the accuracy of linguistic modeling. *IEEE Transactions on Fuzzy Systems* 8(4): 335–344.
- [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.
- [Chi94] Chiu S. (1994) Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems* 2: 267–278.
- [CHL96] Cordón O., Herrera F. y Lozano M. (1996) On the bidirectional integration of genetic algorithms and fuzzy logic. En *Proceedings of the Second Online Workshop on Evolutionary Computation (WEC2)*, páginas 13–16. Nagoya.
- [CHL97a] Cordón O., Herrera F. y Lozano M. (1997) A classified review on the combination fuzzy logic-genetic algorithms bibliography: 1989-1995.

- En Sanchez E., Shibata T. y Zadeh L. (Eds.) *Genetic Algorithms and Fuzzy Logic Systems. Soft Computing Perspectives*, páginas 209–241. World Scientific.
- [CHL97b] Cordon O., Herrera F. y Lozano M. (1997) On the combination of fuzzy logic and evolutionary computation: a short review and bibliography. En Pedrycz W. (Ed.) *Fuzzy Evolutionary Computation*, páginas 57–77. Kluwer Academic.
- [CHMV01] Cordon O., Herrera F., Magdalena L. y Villar P. (2001) A genetic learning process for the scaling factors, granularity and contexts of the fuzzy rule-based system data base. *Information Sciences* 136(1–4): 85–107.
- [CHP97] Cordon O., Herrera F. y Peregrin A. (1997) Applicability of the fuzzy operators in the design of fuzzy logic controllers. *Fuzzy Sets and Systems* 86: 15–41.
- [CHS99] Cordon O., Herrera F. y Sánchez L. (1999) Solving electrical distribution problems using hybrid evolutionary data analysis techniques. *Applied Intelligence* 10: 5–24.
- [CHV00] Cordon 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.
- [CHV01a] Cordon 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.
- [CHV01b] Cordon O., Herrera F. y Villar P. (2001) A genetic learning process for the scaling factors, granularity and contexts of the fuzzy rule-based system data base. *Information Science* 136(1-4): 85–107.
- [CHZ02] Cordon O., Herrera F. y Zwir I. (2002) Linguistic modeling by hierarchical systems of linguistic rules. *IEEE Transactions on Fuzzy Systems* 10(1): 2–20.
- [CK89] Cao Z. y Kandel A. (1989) Applicability of some fuzzy implication operators. *Fuzzy Sets and Systems* 31: 151–186.

- [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.
- [CQ98] Chin T. C. y Qi X. M. (1998) Genetic algorithms for learning the rule base of fuzzy logic controller. *Fuzzy Sets and Systems* 97(1): 1–7.
- [CV94] Cooper M. G. y Vidal J. J. (1994) Genetic design of fuzzy controllers: the cart and jointed pole problem. En *Proceedings of the 3rd IEEE International Conference on Fuzzy Systems*, páginas 1332–1337. Piscataway, NJ, EE. UU.
- [CVL02] Coello C., Veldhuizen D. V. y Lamont G. (2002) *Evolutionary algorithms for solving multi-objective problems*. Kluwer Academic Publishers.
- [CYP96] Chi Z., Yan H. y Pham T. (1996) *Fuzzy algorithms with application to image processing and pattern recognition*. World Scientific, Singapur.
- [Deb01] Deb K. (2001) *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons.
- [DHR93] Driankov D., Hellendoorn H. y Reinfrank M. (1993) *An introduction to fuzzy control*. Springer-Verlag.
- [DPE96] Dexter A. L., Phil D. y Eng C. (1996) Intelligent buildings: fact or fiction? *HVAC&R Research* 2(2): 105–123.
- [ES93] Eshelman L. J. y Schaffer J. D. (1993) Real-coded genetic algorithms and interval-schemata. En Whitley L. (Ed.) *Foundations of genetic algorithms*, volumen 2, páginas 187–202. Morgan Kaufman.
- [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.

- [EV00] Espinosa J. y Vandewalle J. (2000) Constructing fuzzy models with linguistic integrity from numerical data-AFRELI algorithm. *IEEE Transactions on Fuzzy Systems* 8(5): 591–600.
- [FCS04] F. Calvino M.L. Gennusa G. R. y Scaccianoce G. (2004) The control of indoor thermal comfort conditions: introducing a fuzzy adaptive controller. *Energy and Buildings* 36: 97–102.
- [FJ96] Filipic B. y Juricic D. (1996) A genetic algorithm to support learning fuzzy control rules from examples. En Herrera F. y Verdegay J. L. (Eds.) *Genetic Algorithms and Soft Computing*, páginas 403–418. Physica Verlag.
- [Fri97] Fritzke B. (1997) Incremental neuro-fuzzy systems. En Bosacchi B., Bezdek J. C. y Fogel D. B. (Eds.) *Proceedings of SPIE (The International Society for Optical Engineering): Applications of Soft Computing*, volumen 3165, páginas 86–97.
- [GG94] Gudwin R. R. y Gomide F. A. (1994) Context adaptation in fuzzy processing. En *Proceedings of the 1st Brazil-Japan Joint Symposium in Fuzzy Systems*, páginas 15–20. Campinas, SP, Brasil.
- [GGP98] Gudwin R. R., Gomide F. A. y Pedrycz W. (1998) Context adaptation in fuzzy processing and genetic algorithms. *International Journal of Intelligent Systems* 13: 929–948.
- [GH97] González A. y Herrera F. (1997) Multi-stage genetic fuzzy systems based on the iterative rule learning approach. *Mathware & Soft Computing* 4: 233–249.
- [GJ99] Gómez-Skarmeta A. F. y Jiménez F. (1999) Fuzzy modeling with hybrid systems. *Fuzzy Sets and Systems* 104: 199–208.
- [Glo91] Glorennec P. Y. (1991) Application of fuzzy control for building and equipments performance evaluation. En *Proceedings of Building Simulation (BS'91): International Building Performance Simulation Association*, volumen 1, páginas 197–202. Sophia Antipolis, Francia.
- [Glo96] Glorennec P. Y. (1996) Constrained optimization of FIS using an evolutionary method. En Herrera F. y Verdegay J. L. (Eds.) *Genetic Algorithms and Soft Computing*, páginas 349–368. Physica Verlag.

- [GIP99] 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.
- [Gol89] Goldberg D. E. (1989) *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley.
- [GP97] González A. y Pérez R. (1997) Completeness and consistency conditions for learning fuzzy rules. *Fuzzy Sets and Systems* 96(1): 37–51.
- [GP98] González A. y Pérez R. (1998) A fuzzy theory refinement algorithm. *International Journal of Approximate Reasoning* 19(3–4): 193–220.
- [GP99] González A. y Pérez R. (1999) A study about the inclusion of linguistic hedges in a fuzzy rule learning algorithm. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 7(3): 257–266.
- [GP01] González A. y Pérez R. (2001) Selection of relevant features in a fuzzy genetic learning algorithm. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics* 31(3): 417–425.
- [GQ91] Gupta M. M. y Qi J. (1991) Design of fuzzy logic controllers based on generalized t-operators. *Fuzzy Sets and Systems* 40: 473–489.
- [Gre94] Grefenstette J. J. (Ed.) (1994) *Genetic algorithms for machine learning*. Kluwer Academic.
- [GT95] Goonatilake S. y Treleaven P. (Eds.) (1995) *Intelligent systems for finance and bussines*. John Willey & Sons.
- [Gür99] Gürocak H. B. (1999) A genetic-algorithm-based method for tuning fuzzy-logic controllers. *Fuzzy Sets and Systems* 108(1): 39–47.
- [Gür00] Gürocak H. B. (2000) Tuning fuzzy logic controllers using response envelope method. *Fuzzy Sets and Systems* 115(2): 287–304.
- [HCWL04] Hong T. P., Chen C. H., Wu Y. L. y Lee Y. C. (2004) Using divide-and-conquer ga strategy in fuzzy data mining. En *Proceedings of the Ninth International Symposium on Computers and Communications*, volumen 1, páginas 116–121. Alexandria, Egypt.

- [HCWL06] Hong T. P., Chen C. H., Wu Y. L. y Lee Y. C. (2006) A ga-based fuzzy mining approach to achieve a trade-off between number of rules and suitability of membership functions. *Soft Computing* Por aparecer.
- [HG94] Halgamuge S. y Glesner M. (1994) Neural networks in designing fuzzy systems for real world applications. *Fuzzy Sets and Systems* 65(1): 1–12.
- [Hir93] Hirota K. (Ed.) (1993) *Industrial applications of fuzzy technology*. Springer-Verlag.
- [HL99] Hong T. P. y Lee C. Y. (1999) Effect of merging order on performance of fuzzy induction. *Intelligent Data Analysis* 3(2): 139–151.
- [HLS03] Herrera F., Lozano M. y Sánchez A. M. (2003) A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study. *International Journal of Intelligent Systems* 18: 309–338.
- [HLV95] Herrera F., Lozano M. y Verdegay J. L. (1995) Tuning fuzzy controllers by genetic algorithms. *International Journal of Approximate Reasoning* 12: 299–315.
- [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 genetic 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 the behaviour analysis. *Artificial Intelligence Review* 12: 265–319.
- [HLV98c] Herrera F., Lozano M. y Verdegay J. L. (1998) Tackling real-coded genetic algorithms: operators and tools for behavioural 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.

- [HM97] Herrera F. y Magdalena L. (1997) Genetic fuzzy systems. *Tatra Mountains Mathematical Publications. Fuzzy Structures. Current Trends. Lecture Notes of the Tutorial: Genetic Fuzzy Systems. Seventh IFSA World Congress (IFSA'97)* 13: 93–121.
- [HM00] Herrera F. y Martínez L. (2000) A 2-tuple fuzzy linguistic representation model for computing with words. *IEEE Transactions on Fuzzy Systems* 8(6): 746–752.
- [HN94] Huang S. y Nelson R. M. (1994) Rule development and adjustment strategies of a fuzzy logic controller for an HVAC system - parts i and ii (analysis and experiment). *ASHRAE Transactions* 100(1): 841–856.
- [Hol75] Holland J. H. (1975) *Adaptation in natural and artificial systems*. Ann arbor: The University of Michigan Press.
- [HP97] Hoffmann F. y Pfister G. (1997) Evolutionary design of a fuzzy knowledge base for a mobile robot. *International Journal of Approximate Reasoning* 17(4): 447–469.
- [HR78] Holland J. H. y Reitman J. S. (1978) Cognitive systems based on adaptive algorithms. En Waterman D. A. y Hayes-Roth F. (Eds.) *Pattern-Directed Inference Systems*. Academic Press.
- [HS96] Hanebeck U. D. y Schmidt G. K. (1996) Genetic optimization of fuzzy networks. *Fuzzy Sets and Systems* 79(1): 59–68.
- [HT93] Hellendoorn H. y Thomas C. (1993) Defuzzification in fuzzy controllers. *Journal of Intelligent Fuzzy Systems* 1: 109–123.
- [HV96] Herrera F. y Verdegay J. L. (Eds.) (1996) *Genetic algorithms and soft computing*. Physica-Verlag.
- [IM96] Ishibuchi H. y Murata T. (1996) A genetic-algorithms-based fuzzy partition method for pattern classification problems. En Herrera F. y Verdegay J. (Eds.) *Genetic Algorithms and Soft Computing*, páginas 555–578. Physica Verlag.
- [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: 135–150.

- [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.
- [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* 9(3): 260–270.
- [IT01] Ishibuchi H. . y Takashima T. (2001) Effect of rule weights in fuzzy rule-based classification systems. *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., Cordon 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.
- [Jan93] Jang J. S. R. (1993) ANFIS: Adaptive network based fuzzy inference system. *IEEE Transactions on Systems, Man and Cybernetics* 23(3): 665–684.
- [JGSRB01] Jiménez F., Gómez-Skarmeta A., Roubos 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.
- [JvSS99] Jin Y., von Seelen W. y Sendhoff B. (1999) On generating FC³ fuzzy rule systems from data using evolution strategies. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics* 29(4): 829–845.
- [Kar91a] Karr C. (1991) Applying genetics to fuzzy logic. *AI Expert* páginas 38–43.
- [Kar91b] Karr C. (1991) Genetic algorithms for fuzzy controllers. *AI Expert* 6(2): 26–33.

- [KG93] Karr C. y Gentry E. J. (1993) Fuzzy control of pH using genetic algorithms. *IEEE Transactions on Fuzzy Systems* 1(1): 46–53.
- [KKK94] Kinzel J., Klawoon F. y Kruse R. (1994) Modifications of genetic algorithms for designing and optimizing fuzzy controllers. En *Proc. First IEEE Conference on Evolutionary Computation (ICEC'94)*, páginas 28–33. Orlando, FL, USA.
- [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.
- [KKS00] Krone A., Krause H. y Slawinski T. (2000) A new rule reduction method for finding interpretable and small rule bases in high dimensional search spaces. En *Proceedings of the 9th IEEE International Conference on Fuzzy Systems*, páginas 693–699. San Antonio, TX, EE.UU.
- [Kla94] Klawonn F. (1994) Fuzzy sets and vague environments. *Fuzzy Sets and Systems* 66: 207–221.
- [KNN98] Klose A., Nurnberger A. y Nauck D. (1998) Some approaches to improve the interpretability of neuro-fuzzy classifiers. En *Proc. of the 6th European Congress on Intelligent Techniques and Soft Computing*, páginas 629–633. Aachen (Germany).
- [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.
- [KS95] KrishnaKumar K. y Satyadas A. (1995) GA-optimized fuzzy controller for spacecraft attitude control. En Periaux J., Winter G., Galán M. y Cuesta P. (Eds.) *Genetic algorithms in engineering and computer science*, páginas 305–320. John Wiley & Sons, Nueva York, NY, EE. UU.
- [KT01] Krone A. y Taeger H. (2001) Data-based fuzzy rule test for fuzzy modelling. *Fuzzy Sets and Systems* 123(3): 343–358.
- [LAT96] Lotfi A., Andersen H. y Tsoi A. (1996) Interpretation preservation of adaptive fuzzy inference systems. *International Journal of Approximate Reasoning* 15: 379–394.

- [LCT01] Liu B. D., Chen C. Y. y Tsao J. Y. (2001) Design of adaptive fuzzy logic controller based on linguistic-hedge concepts and genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics* 31(1): 32–53.
- [LM94] Liska J. y Melsheimer S. S. (1994) Complete design of fuzzy logic systems using genetic algorithms. En *Proceedings of the 3rd IEEE International Conference on Fuzzy Systems*, páginas 1377–1382. Orlando, FL, EE. UU.
- [LMV99] López S., Magdalena L. y Velasco J. R. (1999) Genetic fuzzy c-means algorithm for the automatic generation of fuzzy partitions. En Bouchon-Meunier B., Yager R. R. y Zadeh L. A. (Eds.) *Information, Uncertainty, Fusion*, páginas 407–418. Kluwer Scientific, Norwell, MA, EE. UU.
- [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.
- [Mag97] Magdalena L. (1997) Adapting the gain of an FLC with genetic algorithms. *International Journal of Approximate Reasoning* 17(4): 327–349.
- [Mam74] Mamdani E. H. (1974) Applications of fuzzy algorithm for control a simple dynamic plant. *Proceedings of the IEE* 121(12): 1585–1588.
- [Mic96] Michalewicz Z. (1996) *Genetic algorithms + data structures = evolution programs*. Springer-Verlag.
- [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.
- [MP96] Mitra S. y Pal S. K. (1996) Fuzzy self organization, inferencing, and rule generation. *IEEE Transactions on Systems, Man, and Cybernetics* 26: 608–619.
- [MR90] Müller B. y Reinhardt J. (1990) *Neural networks. An introduction*. Springer-Verlag, Heidelberg, Alemania.

- [MV97] Magdalena L. y Velasco J. R. (1997) Evolutionary based learning of fuzzy controllers. En Pedrycz W. (Ed.) *Fuzzy Evolutionary Computation*, páginas 249–268. Kluwer Academic.
- [Nau00] Nauck D. (2000) Adaptive rule weights in neuro-fuzzy systems. *Neural Computing and Applications* 9: 60–70.
- [Nel01] Nelles O. (2001) *Nonlinear system identification: from classical approaches to neural networks and fuzzy models*. Springer Verlag.
- [NIT97] Nozaki K., Ishibuchi H. y Tanaka H. (1997) A simple but powerful heuristic method for generating fuzzy rules from numerical data. *Fuzzy Sets and Systems* 86: 251–270.
- [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*, páginas 1235–1240. Anchorage, AK, EE.UU.
- [NK99] Nauck D. y Kruse R. (1999) Neuro-fuzzy systems for function approximation. *Fuzzy Sets and Systems* 101(2): 261–271.
- [NL94] Ng K. C. y Li Y. (1994) Design of sophisticated fuzzy logic controllers using genetic algorithms. En *Proc. Third IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'94)*, volumen 3, páginas 1708–1712. Orlando, FL, USA.
- [PB93] Parodi A. y Bonelli P. (1993) A new approach to fuzzy classifier systems. En *Proc. Fifth International Conference on Genetic Algorithms (ICGA'93)*, páginas 223–230. Morgan Kaufmann.
- [Ped96] Pedrycz W. (Ed.) (1996) *Fuzzy modelling: paradigms and practice*. Kluwer Academic.
- [Ped97] Pedrycz W. (Ed.) (1997) *Fuzzy evolutionary computation*. Kluwer Academic.
- [Ped01] Pedrycz W. (2001) Fuzzy equalization in the construction of fuzzy sets. *Fuzzy Sets and Systems* 119(2): 329–335.
- [Ped04] Pedrycz W. (2004) Associations and rules in data mining: A link analysis. *Int. Journal of Intelligent Systems* 19(7): 653–670.

- [PGG97] Pedrycz W., Gudwin R. y Gomide F. (1997) Nonlinear context adaptation in the calibration of fuzzy sets. *Fuzzy Sets and Systems* 88: 91–97.
- [PK91] Pham D. T. y Karaboga D. (1991) Optimum design of fuzzy logic controllers using genetic algorithms. *Journal of Systems Engineering* 1: 114–118.
- [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.
- [PP99] Pal N. R. y Pal K. (1999) Handling of inconsistent rules with an extended model of fuzzy reasoning. *Journal of Intelligent Fuzzy Systems* 7: 55–73.
- [PRO⁺00] Pomares H., Rojas I., Ortega J., González J. y Prieto A. (2000) A systematic approach to a self-generating fuzzy rule-table for function approximation. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics* 30(3): 431–447.
- [PS01] Peña-Reyes C. A. y Sipper M. (2001) Fuzzy CoCo: a cooperative coevolutionary approach to fuzzy modeling. *IEEE Transactions on Fuzzy Systems* 9(5): 727–737.
- [RGB93] Rovatti R., Guerrieri R. y Baccarani G. (1993) Fuzzy rules optimization and logic synthesis. En *Proceedings of the 2nd IEEE International Conference on Fuzzy Systems*, volumen 2, páginas 1247–1252. San Francisco, EE.UU.
- [RPOP00] Rojas I., Pomares H., Ortega J. y Prieto A. (2000) Self-organized fuzzy system generation from training examples. *IEEE Transactions on Fuzzy Systems* 8(1): 23–36.
- [RS00] Roubos H. y Setnes M. (2000) Compact fuzzy models through complexity reduction and evolutionary optimization. En *Proceedings of the 9th IEEE International Conference on Fuzzy Systems*, volumen 2, páginas 762–767. San Antonio, Texas, EE.UU.
- [RS01] Roubos J. A. y Setnes M. (2001) Compact and transparent fuzzy models and classifiers through iterative complexity reduction. *IEEE Trans. Fuzzy Syst.* 9(4): 516–524.

- [Sán00] Sánchez L. (2000) Interval-valued GA-P algorithms. *IEEE Transactions on Evolutionary Computation* 4(1): 64–72.
- [SBKN98] Setnes M., Babuska R., Kaymak U. y Nauta-Lemke H. R. v. (1998) Similarity measures in fuzzy rule base simplification. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics* 28: 376–386.
- [SEC99] Shi Y., Eberhart R. y Chen Y. (1999) Implementation of evolutionary fuzzy systems. *IEEE Transactions on Fuzzy Systems* 7(2): 109–119.
- [SH00] Setnes M. y Hellendoorn H. (2000) Orthogonal transforms for ordering and reduction of fuzzy rules. En *Proceedings of the 9th IEEE International Conference on Fuzzy Systems*, volumen 2, páginas 700–705. San Antonio, Texas, EE.UU.
- [Sim02] Simon D. (2002) Sum normal optimization of fuzzy membership functions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10(4): 363–384.
- [SK88] Sugeno M. y Kang G. T. (1988) Structure identification of fuzzy model. *Fuzzy Sets and Systems* 28: 15–33.
- [SKFT01] Suzuki T., Kodama T., Furuhashi T. y Tsutsui H. (2001) Fuzzy modeling using genetic algorithms with fuzzy entropy as conciseness measure. *Information Sciences* 136(1–4): 53–67.
- [SKG93] Surmann H., Kanstein A. y Goser K. (1993) Self-organizing and genetic algorithms for an automatic design of fuzzy control and decision systems. En *Proceedings of the 1st European Congress on Fuzzy and Intelligent Technologies*, páginas 1097–1104. Aachen, Alemania.
- [Smi80] Smith S. F. (1980) *A learning system based on genetic adaptive algorithms*. PhD thesis, Universidad de Pittsburgh.
- [SSZ97] Sanchez E., Shibata T. y Zadeh L. (Eds.) (1997) *Genetic algorithms and fuzzy logic systems. Soft computing perspectives*. World Scientific.

- [SY93] Sugeno M. y Yasukawa T. (1993) A fuzzy-logic-based approach to qualitative modeling. *IEEE Transactions on Fuzzy Systems* 1(1): 7–31.
- [Tet95] Tettamanzi A. G. (1995) Evolutionary algorithms and fuzzy logic: a two-way integration. En *Proc. Second Joint Conference on Information Sciences*, páginas 464–467.
- [Thr91] Thrift P. (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 Kacprzyk J. y Yager R. (Eds.) *Management Decision Support Systems Using Fuzzy Logic and Possibility Theory*, páginas 198–212. Verlag TUV Rheinland.
- [TW04] Teng Y. y Wang W. (2004) Constructing a user-friendly ga-based fuzzy system directly from numerical data. *IEEE Trans. Syst., Man, Cybern. B* 34(5): 2060–2070.
- [Val99a] Valente de Oliveira J. (1999) Semantic constraints for membership function optimization. *IEEE Transactions on Systems, Man, and Cybernetics. Part A: Systems and Humans* 29(1): 128–138.
- [Val99b] Valente de Oliveira J. (1999) Towards neuro-linguistic modeling: constraints for optimization of membership functions. *Fuzzy Sets and Systems* 106(3): 357–380.
- [Vel98] Velasco J. R. (1998) Genetic-based on-line learning for fuzzy process control. *International Journal of Intelligent Systems* 13(10-11): 891–903.
- [Ven92] Venturini G. (1992) SIA: a supervised inductive algorithm with genetic search for learning attribute based concepts. En *Proc. European Conference on Machine Learning*, páginas 280–296. Viena, Austria.

- [VR91] Valenzuela-Rendón M. (1991) The fuzzy classifier system: Motivations and first results. En Schwefel H. P. y Männer R. (Eds.) *Proc. First International Conference on Parallel Problem Solving from Nature - PPSN I*, páginas 330–334. Springer-Verlag, Berlin.
- [Wan94] Wang L. X. (1994) *Adaptive fuzzy systems and control*. Prentice-Hall.
- [WHT98] Wang C., Hong T. y Tseng S. (1998) Integrating fuzzy knowledge by genetic algorithms. *IEEE Transactions on Evolutionary Computation* 2(4): 138–149.
- [WK88] Whitley D. y Kauth J. (1988) Genitor: A different genetic algorithm. En *Proceedings of the Rocky Mountain Conference on Artificial Intelligence*, páginas 118–130. Denver, EE.UU.
- [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.
- [XL00] Xiong N. y Litz L. (2000) Fuzzy modeling based on premise optimization. En *Proceedings of the 9th IEEE International Conference on Fuzzy Systems*, páginas 859–864. San Antonio, TX, EE. UU.
- [YBY99] Yam Y., Baranyi P. y Yang C. (1999) Reduction of fuzzy rule base via singular value decomposition. *IEEE Transactions on Fuzzy Systems* 7: 120–132.
- [YW99] Yen J. y Wang L. (1999) Simplifying fuzzy rule-based models using orthogonal transformation methods. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics* 29: 13–24.
- [Zad65] Zadeh L. A. (1965) Fuzzy sets. *Information and Control* 8: 338–353.
- [Zad71] Zadeh L. A. (1971) Toward a theory of fuzzy systems. En *Aspects of network and system theory*, páginas 469–490. Rinehart and Winston, Nueva York, NY, EE. UU.
- [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.

- [Zad74] Zadeh L. A. (1974) On the analysis of large scale systems. En *Systems approaches and environment problems*, páginas 23–37. Vandenhoeck and Ruprecht, Gottingen, Alemania.
- [Zad75] Zadeh L. A. (1975) The concept of a linguistic variable and its applications to approximate reasoning. Parts I, II and III. *Information Science* 8, 8 and 9: 199–249, 301–357 and 43–80.
- [Zad76] Zadeh L. A. (1976) A fuzzy-algorithmic approach to the definition of complex or imprecise concepts. *International Journal on Man-Machine Studies* 8(3): 249–291.
- [Zad96] Zadeh L. A. (1996) Fuzzy logic and the calculi of fuzzy rules and fuzzy graphs: a precis. *International Journal of Multiple-Valued Logic* 1: 1–38.
- [Zhe92] Zheng L. (1992) A practical guide to tune proportional and integral (pi) like fuzzy controllers. En *Proc. of the First IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'92)*, páginas 633–640. San Diego.