

UNIVERSIDAD DE GRANADA

E.T.S. DE INGENIERÍA INFORMÁTICA



Departamento de
Ciencias de la Computación e Inteligencia Artificial

Aprendizaje Cooperativo para Modelado
Lingüístico Flexible Basado en Reglas Difusas:
Interpretabilidad y Precisión

Tesis Doctoral

Jorge Casillas Barranquero

Granada, mayo de 2001



Aprendizaje Cooperativo para Modelado Lingüístico Flexible Basado en Reglas Difusas: Interpretabilidad y Precisión

MEMORIA QUE PRESENTA

Jorge Casillas Barranquero

PARA OPTAR AL GRADO DE DOCTOR EN INFORMÁTICA

MAYO DE 2001

DIRECTORES

Óscar Cerdón García

Francisco Herrera Triguero

DEPARTAMENTO DE
CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIAL

E.T.S. DE INGENIERÍA INFORMÁTICA

UNIVERSIDAD DE GRANADA

La memoria titulada **Aprendizaje Cooperativo para Modelado Lingüístico Flexible Basado en Reglas Difusas: Interpretabilidad y Precisión**, que presenta D. Jorge Casillas Barranquero para optar al grado de doctor, ha sido realizada en el Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada bajo la dirección de los doctores D. Óscar Cordon García y D. Francisco Herrera Triguero.

Granada, mayo de 2001

El doctorando:

Jorge Casillas

Los directores:

Óscar Cordon

Francisco Herrera

Agradecimientos

Existe un tipo de relación básica en la vida humana. Empieza con la relación entre los padres y los hijos. Sigue con los profesores y alumnos. Y luego, si tenemos suerte en la vida laboral, quizás haya alguien que reconozca que poseemos alguna aptitud que ni siquiera sabíamos, y que nos motiva a hacer cosas, que tal vez ignorábamos que fuéramos capaces de hacer. He tenido la gran fortuna de experimentar esa relación mayéutica y por ello agradezco a mis mentores Óscar Cordón y Francisco Herrera su apoyo y confianza incondicional.

También desearía dar las gracias a todas aquellas personas que tanto me han ayudado en numerosos aspectos de la preparación de este manuscrito, anímica o intelectualmente.

A mis padres por la comprensión y dedicación que me ofrecen, por el interés que muestran por mi trabajo y por hacer mis problemas suyos. Agradezco a mi hermana Anabel la compañía que me aporta y la contagiosa alegría que transmite. A Velén, mi musa particular, que me dio toda su fuerza durante el desarrollo de esta memoria y que estuvo a mi lado en cada infortunio que sufrí.

Mi agradecimiento a Rafa por su cercanía y el tiempo que hemos compartido desde que nos embarcamos en esta aventura. Una mención muy especial al resto de compañeros de Mecenaz: Nico, los Fernandos y José María; mi reconocimiento por su compañía y amistad.

Tampoco puedo olvidar a María José del Jesus, por su ayuda; a Igor Zwir, por sus lecciones bonarenses; y a José Manuel Benítez, por su amable asistencia con \LaTeX .

Quiero expresar mi profunda gratitud a José Piñar y Ángeles Agrela por su genialidad en el diseño de la portada, por las cenas de pasta y su generosa disponibilidad.

Y, finalmente, mi agradecimiento y disculpas a todos aquellos amigos y amigas que descuidé mientras me ahogaba en esta memoria, pero a los que prometo recuperar el tiempo perdido, porque ¡vuelvo a ser persona!

Índice General

Introducción	1
A. Planteamiento	1
B. Objetivos	4
C. Resumen	6
1. Modelado de Sistemas Mediante Sistemas Basados en Reglas Difusas	9
1.1. Identificación de Sistemas	10
1.2. Modelado de Sistemas con Sistemas Basados en Reglas Difusas	14
1.2.1. Tipos de Modelado con Sistemas Basados en Reglas Difusas	15
1.2.2. Modelado Lingüístico o Difuso: Interpretabilidad o Precisión	17
1.3. Modelado Difuso Restringido	19
1.4. Modelado Lingüístico Extendido	21
1.4.1. Mejoras al Proceso de Derivación del Modelo Lingüístico	22
1.4.2. Extensiones de la Estructura del Modelo Lingüístico	24
1.4.2.1. Uso de Modificadores Lingüísticos	24
1.4.2.2. Uso de más de un Consecuente para cada Regla	26
1.4.2.3. Uso de Reglas Ponderadas	26
1.4.3. Interpretabilidad del Modelado Lingüístico Extendido	27
1.5. Estudio Experimental del ML Extendido frente al MD	30
1.5.1. Métodos de Derivación de Reglas Difusas Analizados	32
1.5.1.1. Métodos de Derivación de Modelos Lingüísticos	32
1.5.1.2. Métodos de Derivación de Modelos Lingüísticos Extendidos	32
1.5.1.3. Métodos de Derivación de Modelos Difusos Aproximativos	37
1.5.2. Experimentos	39

1.5.3.	Análisis de Resultados	39
1.5.3.1.	Modelado Lingüístico frente a Modelado Difuso	42
1.5.3.2.	Modelado Lingüístico frente a Modelado Lingüístico Extendido	43
1.5.3.3.	Modelado Lingüístico Extendido frente a Modelado Difuso	53
1.6.	Sumario	54
1.7.	Nuevas Líneas de Trabajo Propuestas en esta Memoria	57
2.	Metodología COR: Cooperación entre Reglas	59
2.1.	Métodos Específicos de Envoltura de Ejemplos: Características y Taxonomía	61
2.1.1.	Características de los Métodos Específicos de Envoltura de Ejemplos	61
2.1.2.	Taxonomía de los Métodos Específicos de Envoltura de Ejemplos	63
2.1.2.1.	Métodos Específicos de Envoltura de Ejemplos Guiados por Ejemplos	63
2.1.2.2.	Métodos Específicos de Envoltura de Ejemplos Guiados por Rejilla Difusa	64
2.1.2.3.	Relación entre los Métodos Guiados por Ejemplos y los Guiados por Rejilla Difusa	67
2.1.2.4.	Esquema Genérico de los Métodos Específicos de Envoltura de Ejemplos	68
2.2.	Una Nueva Metodología para Mejorar la Precisión con Reglas Más Cooperativas	70
2.2.1.	La Metodología de Cooperación entre Reglas	71
2.2.2.	Análisis de la Metodología de Cooperación entre Reglas	72
2.3.	Uso de Diferentes Técnicas de Búsqueda en la Metodología COR	75
2.3.1.	Aplicación del ES a la Metodología COR	75
2.3.2.	Aplicación de la Búsqueda Tabú a la Metodología COR	77
2.3.3.	Aplicación de los AGs a la Metodología COR	79
2.3.4.	Aplicación de los Algoritmos de Optimización Basados en Colonias de Hormigas a la Metodología COR	80
2.4.	Estudio Experimental de la Metodología COR	82

2.4.1.	Tablas de Resultados y Valores de Parámetros	83
2.4.2.	Análisis de los Resultados Obtenidos	87
2.4.2.1.	Métodos Guiados por Ejemplos frente a Métodos Guiados por Rejilla Difusa	88
2.4.2.2.	Métodos Basados en COR frente al Resto	89
2.4.2.3.	Enfoques de Generación de Reglas y Técnicas de Búsqueda en los Métodos Basados en COR	93
2.5.	Extensiones a la Metodología COR	97
2.5.1.	Uso del Sistema de Colonias de Hormigas	99
2.5.2.	Uso de una Búsqueda Local dentro del Sistema de Colonias de Hormigas	99
2.5.3.	Extensión de la Metodología COR para Eliminar Reglas Lin- güísticas	100
2.6.	Estudio Experimental de las Extensiones a la Metodología COR	103
2.7.	Sensibilidad de los Parámetros en los Métodos de OCH Basados en COR y CORR	106
2.8.	Sumario	109
3. Ajuste de las Estructuras Profundas de las Reglas Lingüísticas		113
3.1.	Estructuras Profundas de las Reglas Lingüísticas Difusas	114
3.2.	Ajuste de Estructuras Profundas en SBRDs	115
3.2.1.	Ajuste de la Estructura Superficial usando Modificadores Lin- güísticos	116
3.2.2.	Ajuste de la Estructura Profunda Con Variaciones Lineales y no Lineales	117
3.3.	Un Método para el Ajuste Completo de las Estructuras Profundas	120
3.4.	Estudio Experimental	126
3.4.1.	Tablas de Resultados y Valores de Parámetros	127
3.4.2.	Análisis de los Resultados Obtenidos	130
3.5.	Sumario	133
4. Hibridación de Diferentes Mejoras al Modelado Lingüístico		139
4.1.	Derivación Secuencial o Simultánea	140
4.2.	Algoritmos Coevolutivos Cooperativos	141
4.3.	Hibridaciones Propuestas	143
4.3.1.	Método CPAL	145

4.3.2.	Métodos Coevolutivos: PA-L, C-PAL y CL-PA	146
4.3.2.1.	Esquema de Interacción Cooperativa Considerado	146
4.3.2.2.	Operadores Evolutivos	148
4.4.	Estudio Experimental	150
4.4.1.	Tabla de Resultados y Valores de Parámetros	150
4.4.2.	Análisis de los Resultados Obtenidos	153
4.5.	Sumario	157
Comentarios Finales		163
A.	Resultados Obtenidos y Conclusiones	163
	Modelado Lingüístico frente a Modelado Difuso	163
	Aprendizaje Rápido de Modelos Lingüísticos con Buena Precisión y Máxima Interpretabilidad	164
	Ajuste de Modelos Lingüísticos	165
	Hibridación de las Mejoras al Modelado Lingüístico	165
B.	Líneas de Investigación Futuras	166
	Mejoras en el Método de Ajuste	166
	Mejoras en la Coevolución Cooperativa	167
	Proceso de Derivación más Completo	167
	Mecanismos para Evitar el Sobreaprendizaje	168
	Interpretación Lingüística <i>a Posteriori</i>	168
A. Sistemas Basados en Reglas Difusas		171
A.1.	Grafos Difusos para Modelado de Sistemas	171
A.2.	Sistemas Basados en Reglas Difusas Lingüísticos	174
A.3.	Sistemas Basados en Reglas Difusas Aproximativos	176
A.4.	Análisis de los Componentes de los SBRDs Lingüísticos	177
A.4.1.	La Base de Conocimiento	178
A.4.2.	La Interfaz de Fuzzificación	179
A.4.3.	El Sistema de Inferencia	180
A.4.4.	La Interfaz de Defuzzificación	181
B. Técnicas de Búsqueda		185
B.1.	Enfriamiento Simulado	185
B.1.1.	Estructura del Enfriamiento Simulado	187
B.1.2.	Ejemplo de Mecanismo de Enfriamiento	188

B.2. Búsqueda Tabú	189
B.3. Algoritmos Genéticos	189
B.3.1. Representación de las Soluciones	192
B.3.2. El Mecanismo de Selección	193
B.3.3. El Operador de Cruce	194
B.3.4. El Operador de Mutación	195
B.4. Optimización Basada en Colonias de Hormigas	197
B.4.1. Introducción	197
B.4.2. El Sistema de Hormigas	199
B.4.3. El Sistema de Colonias de Hormigas	200
C. Consideraciones en la Experimentación Realizada	201
C.1. Aspectos Generales de la Experimentación	201
C.2. Descripción de las Aplicaciones Consideradas	202
C.2.1. Modelado de la Función Tridimensional F_1	202
C.2.2. Modelado de la Función Tridimensional F_2	204
C.2.3. Modelado de la Función Tridimensional F_3	204
C.2.4. Estimación de la Calidad del Arroz	205
C.2.5. Problemas de Distribución de Energía Eléctrica	206
C.2.5.1. Estimación de la Longitud de las Líneas de Baja Tensión	206
C.2.5.2. Estimación del Costo de Mantenimiento de la Lí- nea de Media Tensión	207
Bibliografía	209
Índice de Autores	219

Índice de Figuras

0.1. Interpretación de la edad como un valor lingüístico	2
0.2. Equilibrio entre flexibilidad y simplicidad en modelado con SBRDs	4
1.1. Modelado de sistemas con SBRDs: interpretabilidad y flexibilidad dependiendo del tipo de SBRD considerado	17
1.2. Equilibrio entre flexibilidad y simplicidad: ML extendido y MD res- tringido	18
1.3. Aprendizaje con o sin restricciones en el MD	20
1.4. Variaciones lineales y no lineales de las funciones de pertenencia . .	25
1.5. Efectos de los modificadores lingüísticos ‘muy’ y ‘más-o-menos’ . .	25
1.6. Pérdida de interpretabilidad dependiendo de la extensión realizada en el ML	27
1.7. Se pierde cierta interpretabilidad al hacer una modificación brusca de las funciones de pertenencia	29
1.8. Obtención de reglas lingüísticas a partir de reglas tipo TSK simplifi- cado en el método NIT	36
1.9. Modelos lingüísticos obtenidos por los métodos T y WM-RP para el problema F ₂	44
1.10. Modelo lingüístico obtenido por el método NIT para el problema F ₂	45
1.11. Modelo difuso generado por el método P-FCS1 en el problema F ₂ .	46
1.12. Modelo difuso generado por el método LM para el problema F ₃ . .	47
1.13. Modelo difuso generado por el método WCA para el problema F ₃ .	48
1.14. Algunos modelos generados para el problema del arroz	49
1.15. Modelos generados por T y T+M-L-Ajus en el problema de las líneas de baja tensión	50
1.16. Algunos modelos generados en el problema de las líneas eléctricas de baja tensión	51

2.1. Representación gráfica de una partición difusa uniforme	62
2.2. Proceso de generación de reglas seguido en los métodos guiados por ejemplos	63
2.3. Método WM	64
2.4. Proceso de generación de reglas seguido en los métodos guiados por rejilla difusa	65
2.5. Método CH	66
2.6. Métodos guiados por ejemplos frente a métodos guiados por rejilla difusa	68
2.7. Esquema de los métodos específicos de envoltura de ejemplos . . .	69
2.8. Interpretación de los métodos WM y CH según el esquema genérico	70
2.9. Ilustración gráfica del proceso de aprendizaje basado en COR en un problema de ejemplo sencillo	73
2.10. Proceso de generación de reglas seguido por los enfoques CORWM y CORCH	74
2.11. Comparación entre el espacio de búsqueda manejado por el método Thrift y la metodología COR en el problema de la línea de media tensión con cinco términos lingüísticos en cada variable	76
2.12. Proceso de construcción del grafo para el algoritmo COR basado en OCH	81
2.13. Grafo del algoritmo de OCH correspondiente al ejemplo de la figura 2.9	81
2.14. Método WM frente al método CORWM-ES modelando F_1	90
2.15. Método WM frente al método CORWM-ES en el problema del arroz	91
2.16. BRs generadas por los métodos NIT y CORCH-SH en el problema de la línea de baja tensión	92
2.17. Espacio de búsqueda manejado por los dos enfoques de la metodología COR y el método T	93
2.18. BRs generadas por el método T y la metodología COR en F_2	94
2.19. Gráfica de evolución de los métodos CORWM-ES y CORCH-ES en el problema F_1	95
2.20. Gráfica de evolución de los cuatro métodos que siguen el enfoque CORCH en una partición del problema del arroz	97
2.21. Proceso de búsqueda local usado en el algoritmo SCH	100

2.22. Extensión permitiendo que la metodología COR elimine reglas lingüísticas	102
2.23. Espacio de búsqueda manejado por CORCH, CORRCH y T	103
2.24. BRs generadas por los métodos CORCH-SH y CORRCH-SCHBL en el problema de la línea de baja tensión	107
3.1. Estructuras superficiales y profundas en una regla lingüística difusa	114
3.2. Tipos de modificadores lingüísticos	118
3.3. Tipos de ajuste considerados en el método propuesto	121
3.4. Esquema de codificación para el ajuste de SBRDs	122
3.5. Intervalos de variación considerados en el ajuste	122
3.6. Ejemplo de proceso de ajuste genético	125
3.7. Proceso de derivación secuencial: aprendizaje más ajuste	126
3.8. Estructuras profundas generadas por el método WM+PA-Ajus en F_1	131
3.9. Estructuras profundas generadas por el método T+PL-Ajus en F_2 . .	132
3.10. BD generada por el método T+AL-Ajus en el problema del arroz . .	134
3.11. BD generada por el método WM+PAL-Ajus para el problema de la línea de media tensión	136
4.1. Coevolución cooperativa	143
4.2. Hibridaciones analizadas	145
4.3. Esquema de interacción considerado en el método de aprendizaje cooperativo coevolutivo	147
4.4. Modelo lingüístico generado por el método CL-PA en F_1	155
4.5. Modelo lingüístico generado por el método C+PAL en F_2	156
4.6. BD generada por el método C-PAL para el problema del arroz . . .	158
4.7. BD generada por el método CPAL para el problema de la línea de media tensión	160
A.1. Representación aproximada de relaciones con grafos difusos restringidos y sin restricciones	173
A.2. Paralelismo entre SBRDs lingüísticos y SBRDs aproximativos . . .	174
A.3. Estructura básica de un SBRD lingüístico	175
A.4. Comparación gráfica entre BC lingüísticas y BRD aproximativas . .	177
B.1. Algoritmo básico del enfriamiento simulado	188

B.2. Algoritmo básico de la búsqueda tabú	190
B.3. Estructura básica de un Algoritmo Genético	192
B.4. Ejemplo de aplicación del mecanismo de selección	193
B.5. Ejemplo de aplicación del operador de cruce simple en un punto	195
B.6. Ejemplo de aplicación del operador de mutación	196
C.1. Representación gráfica, expresión matemática, y universos de discurso de las variables de la función tridimensional F_1	203
C.2. Representación gráfica, expresión matemática y universos de discurso de las variables de la función tridimensional F_2	204
C.3. Representación gráfica, expresión matemática y universos de discurso de las variables de la función tridimensional F_3	205

Índice de Tablas

1.1. Resumen de los métodos de derivación considerados y sus principales características	31
1.2. Resultados obtenidos por los métodos de ML, ML extendido y MD analizados (I)	40
1.3. Resultados obtenidos por los métodos de ML, ML extendido y MD analizados (II)	41
2.1. Métodos considerados en el estudio experimental de la metodología COR	84
2.2. Resultados obtenidos por los métodos basados en la metodología COR y el resto de métodos de comparación (I)	85
2.3. Resultados obtenidos por los métodos basados en la metodología COR y el resto de métodos de comparación (II)	86
2.4. Valores de los parámetros considerados en cada método para cada problema	87
2.5. Resultados obtenidos por el método CORCH-SH con y sin información heurística	98
2.6. Resultados obtenidos por las extensiones sobre la metodología COR y el resto de métodos de comparación	104
2.7. Valores de los parámetros considerados en cada método para cada problema en la segunda experimentación	105
2.8. Sensibilidad a los parámetros y algunos valores recomendados para los diferentes métodos de OCH basados en COR y CORR	108
2.9. Notación empleada en el capítulo 2	111
3.1. Procesos de ajuste considerados en el estudio experimental	127
3.2. Resultados obtenidos por los métodos de ajuste propuestos (I)	128

3.3. Resultados obtenidos por los métodos de ajuste propuestos (II) . . .	129
3.4. BR generada por el método T+AL-Ajus en el problema del arroz . .	135
3.5. BR generada por el método WM+PAL-Ajus para el problema de la línea de media tensión	137
4.1. Resultados obtenidos por las diferentes hibridaciones propuestas . .	151
4.2. Mejores resultados obtenidos por los métodos de ML y MD analiza- dos en el capítulo 1 y por el método PAL-Ajus del capítulo 3	153
4.3. BR generada por el método C-PAL para el problema del arroz . . .	159
4.4. BR generada por el método CPAL para el problema de la línea de media tensión	161
C.1. Principales características de las aplicaciones consideradas en esta memoria	203
C.2. Notación y universos de discurso en el problema de la estimación de la longitud de las líneas de baja tensión	207
C.3. Notación y universos de discurso en el problema de mantenimiento de las líneas de media tensión	208

Introducción

A. Planteamiento

¿Qué se entiende por *modelado*? Si acudimos al Diccionario de la Lengua Española¹, encontramos que modelado es la acción y efecto de ajustarse a un modelo, esto es, a un «*esquema teórico, generalmente en forma matemática, de un sistema o de una realidad compleja [...] que se elabora para facilitar su comprensión y el estudio de su comportamiento*».

El desarrollo de modelos matemáticos de sistemas reales es un aspecto vital en numerosas disciplinas de ingeniería y ciencia. Los modelos se pueden usar, por ejemplo, para simulaciones, para analizar el comportamiento de un sistema, para comprender mejor los mecanismos subyacentes en el mismo, para diseñar nuevos procesos o para controlar automáticamente sistemas. Resulta crucial diseñar modelos que representen de forma adecuada la realidad. Si el modelo no es lo suficientemente preciso, es decir, lo suficientemente fiel al sistema que se está modelando, los posteriores pasos de análisis, predicción, síntesis, etc., no podrán realizarse con éxito. Sin embargo, existe una relación evidente entre la precisión necesaria en un modelo y su complejidad. Si el modelo es demasiado simple, no podrá representar adecuadamente las características relevantes del sistema y no servirá para nuestros propósitos. Por otro lado, un modelo demasiado complejo no satisface su principal función, la de facilitar la comprensión del comportamiento del sistema modelado, resultando inútil en la práctica.

Los sistemas se pueden representar con modelos matemáticos con estructuras muy diferentes, tales como ecuaciones algebraicas, ecuaciones diferenciales, máquinas de estado finitas, etc. Una de las formas más interesantes de realizar esta representación consiste en emplear reglas SI-ENTONCES del siguiente modo:

¹ <http://www.rae.es/>

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

Estas reglas establecen relaciones lógicas entre las variables del sistema con valores cualitativos. Dicha representación tiene la gran virtud de ser fácilmente comprensible por el ser humano, por lo que el comportamiento del sistema se refleja eficazmente. Sin embargo, aunque al razonamiento humano no le supone ningún esfuerzo manejar términos del tipo '*alta*' o '*subirá rápidamente*', en un proceso automático este tratamiento resulta más complejo.

Para trabajar adecuadamente con valoraciones cualitativas de este tipo, se emplean variables lingüísticas basadas en la Teoría de Conjuntos Difusos y la Lógica Difusa, de modo que la regla anterior se conoce como *regla difusa*. La lógica difusa proporciona mayor generalidad, poder expresivo, habilidad para modelar problemas reales y, no menos importante, una metodología para explotar la tolerancia frente a la imprecisión. Por ejemplo, podemos considerar la variable lingüística '*edad*', cuyos valores lingüísticos podrían ser '*adolescente*', '*joven*', '*adulto*' y '*viejo*'. La figura 0.1 ilustra gráficamente cómo se interpreta el término '*adulto*' empleando variables lingüísticas.

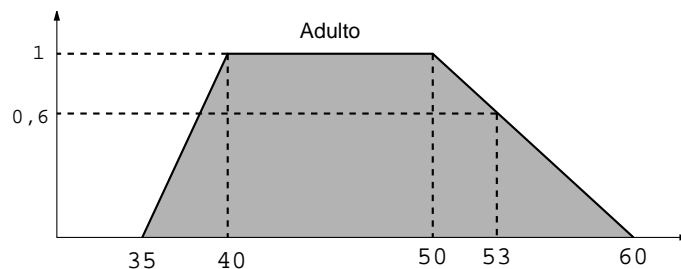


Figura 0.1.: Interpretación de la edad como un valor lingüístico. Podríamos decir que una edad de 53 años corresponde en un 60% con el concepto lingüístico de '*adulto*'

En el proceso de modelado, estas reglas se emplean dentro de lo que se conoce como un *Sistema Basado en Reglas Difusas* (SBRD). Dependiendo del tipo de SBRD que consideremos, podemos realizar distintas clases de modelado atendiendo a diferentes requisitos. Principalmente, podemos distinguir dos tipos de modelado con características enfrentadas: *modelado lingüístico* (ML) y *modelado difuso* (MD). Mientras que el primero atiende fundamentalmente al poder descriptivo de los

SBRDs, buscando modelos con una gran *comprensibilidad* (legibilidad o interpretabilidad), el segundo busca fundamentalmente alcanzar un alto grado de *precisión*.

En general, la alta interpretabilidad se obtiene imponiendo ciertas restricciones mediante el uso de variables lingüísticas, lo cual conlleva que los SBRDs sean más rígidos. Por contra, la precisión se persigue relajando dichas restricciones para hacer a los SBRDs más flexibles, resignándose a la consecuente pérdida de legibilidad.

Esta *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 sobreajustarse 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. En efecto,

¿de qué sirve un modelo con gran flexibilidad si el proceso de diseño tiene serias dificultades para aprovecharla?

Así, mayor flexibilidad no significa mayor precisión y un equilibrio entre la libertad de la estructura del modelo (*flexibilidad del modelo*) y un buen rendimiento del proceso de diseño del mismo (*simplicidad del modelado*) podría proporcionar mejor comportamiento, esto es, una mayor precisión (véase la figura 0.2).

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

Para alcanzar este compromiso de flexibilidad y simplicidad, o bien se pueden realizar mejoras al ML para hacer más flexible el diseño y/o la estructura del modelo, o bien se pueden imponer restricciones al diseño y/o a la estructura del modelo difuso.

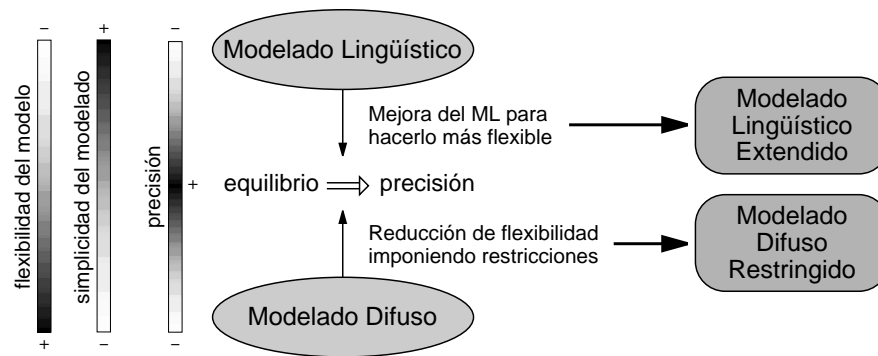


Figura 0.2.: Un compromiso entre flexibilidad del modelo y simplicidad del modelado podría proporcionar mejor precisión. Esto se puede conseguir extendiendo el ML o restringiendo el MD

Si dicho equilibrio se alcanza partiendo del ML, los modelos obtenidos serán más interpretables que partiendo del MD. Además, podríamos aducir el principio de la navaja de Occam: «*La pluralidad no se debe postular sin necesidad*»². O, en nuestro caso, *no debemos complicar el modelo más allá de lo necesario*. Por tanto, parece muy interesante afrontar el modelado de sistemas intentado mejorar, tanto como sea posible, la precisión del ML sin perder su descripción en exceso. Este será el objetivo principal de la presente memoria.

B. Objetivos

El propósito general de esta memoria se centrará en el desarrollo de nuevos métodos de aprendizaje que permitan obtener modelos lingüísticos más precisos con un alto grado de interpretabilidad atendiendo principalmente al aspecto cooperativo entre los diferentes componentes del modelo. Esta intención global se desglosa en los siguientes objetivos:

1. *Análisis de la literatura actual en el campo del Modelado de Sistemas con SBRDs* — Realizaremos un análisis profundo de las propuestas existentes actualmente para realizar el Modelado de Sistemas con SBRDs, bien sea me-

² «*Pluralitas non est ponenda sine neccesitate*», William de Ockham (c. 1285–c. 1349), filósofo medieval inglés y monje franciscano

dianete ML o mediante MD. Asimismo, escogeremos varios métodos representativos para realizar un estudio experimental que nos permitirá corroborar las hipótesis expuestas y observar el comportamiento de las diferentes estrategias de modelado. Este análisis nos conducirá a plantear los diferentes métodos de modelado presentados en esta memoria.

2. *Mejora en la generación de reglas lingüísticas* — Propondremos una nueva metodología de generación de reglas con buena cooperación entre sí que permita obtener modelos con buena precisión. Su principal virtud será la rapidez del proceso, que hará que la metodología sea adecuada para aquellas situaciones en las que se requiere un diseño rápido o para encadenar este aprendizaje previo con otro proceso posterior de refinamiento de las soluciones obtenidas. Naturalmente, dado que la estructura de los modelos lingüísticos permanece inalterada, los modelos obtenidos gozarán de la máxima interpretabilidad.

Por tanto, los principales objetivos en este método son:

- *Eficiencia alta*: generar rápidamente modelos lingüísticos.
- *Precisión media*: hacer que los modelos obtenidos tengan un grado de precisión aceptable.
- *Interpretabilidad alta*: obtener modelos con el máximo grado de interpretabilidad posible.

3. *Extensiones ajustando funciones de pertenencia y usando una estructura más flexible con modificadores lingüísticos* — Propondremos un método que ajustará modelos previamente diseñados efectuando modificaciones en dos niveles de descripción diferentes (con efectos macroscópicos y microscópicos). En este proceso se perseguirá mantener un alto grado de interpretabilidad.

En este caso, los principales objetivos son:

- *Precisión alta*: hacer que los modelos lingüísticos obtenidos tengan una gran precisión.
- *Interpretabilidad media*: hacer que los modelos lingüísticos obtenidos mantengan un grado de interpretabilidad adecuado.

4. *Diferentes formas de combinar las mejoras anteriores* — Por último, propondremos varios métodos de derivación del modelo global que agrupan las

propuestas anteriores de diferente forma. Distinguiremos dos mecanismos de hibridación: derivación secuencial o simultánea y empleo de Algoritmos Genéticos o Coevolución Cooperativa.

Estos métodos tienen como principales objetivos los siguientes:

- *Precisión muy alta*: generar modelos lingüísticos tan precisos como sea posible.
- *Interpretabilidad media*: hacer que los modelos lingüísticos obtenidos mantengan un grado de interpretabilidad adecuado.

C. Resumen

La memoria está organizada en cuatro capítulos, una sección de comentarios finales y tres apéndices. A continuación introducimos brevemente los contenidos de cada uno de ellos.

En el capítulo 1 plantaremos tres paradigmas diferentes en cuanto a la tarea de modelado de datos, es decir, de la comprensión de la naturaleza y del comportamiento de un sistema y la consecuente tarea de creación de un modelo que pueda ser posteriormente utilizado para emularlo. Centrándonos en el modelado por medio de SBRDs, 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 varios métodos de diseño de SBRDs, realizaremos un profundo estudio experimental que nos permitirá analizar el comportamiento de los diferentes enfoques mencionados y arrojar algunas conclusiones que servirán de base para los métodos propuestos en la memoria.

En el capítulo 2 propondremos una metodología de aprendizaje de reglas difusas que permitirá obtener una precisión adecuada favoreciendo la cooperación entre ellas. Para tal fin, comenzaremos el capítulo presentando una taxonomía de un determinado tipo de métodos de aprendizaje caracterizado por su rapidez de proceso. Basándonos en este análisis, introduciremos la metodología propuesta, así como varios métodos de aprendizaje basados en ella que emplean distintas técnicas de búsqueda. Realizaremos posteriormente un estudio experimental para comprobar la eficacia de la metodología propuesta. Este estudio nos conducirá a proponer algunas mejoras sobre dicha metodología. Completaremos el capítulo con un nuevo estudio experimental de las extensiones mencionadas.

En el capítulo 3 presentaremos un método de diseño consistente en ajustar modelos lingüísticos previamente obtenidos. Veremos cómo efectuar dicho refinamiento alterando el modelo en dos niveles de especificidad diferentes, lo que redundará en una mayor precisión. Después de este estudio, presentaremos un método de ajuste basado en un algoritmo genético y realizaremos la correspondiente experimentación para demostrar su buen comportamiento.

En el capítulo 4 propondremos varios métodos de aprendizaje del modelo lingüístico que aglutina las propuestas de diseño anteriores de diferentes formas. Para ello, distinguiremos entre derivación secuencial o simultánea y consideraremos un novedoso paradigma de optimización que permite manejar eficazmente estas tareas de hibridación, la Coevolución Cooperativa. Analizaremos el comportamiento de las diferentes hibridaciones consideradas con un estudio experimental.

Incluiremos una sección de comentarios finales que sintetizará los resultados obtenidos en esta memoria y servirá como compendio y conclusión de los aportes realizados en la misma. Introduciremos igualmente una serie de líneas futuras de investigación que proponemos como continuación del trabajo realizado.

Por último, incluiremos tres apéndices. El primero estará destinado a introducir los SBRDs, estudiando los tipos existentes, su composición y funcionamiento. El siguiente apéndice contiene una introducción a las diferentes técnicas de búsqueda empleadas en la memoria: el enfriamiento simulado, la búsqueda tabú, los algoritmos genéticos y la optimización basada en colonias de hormigas. El tercer apéndice incluye algunas consideraciones sobre las experimentaciones realizadas y la descripción de los diferentes problemas utilizados en la memoria para mostrar el comportamiento de los métodos analizados.

Finalizaremos la memoria con una recopilación bibliográfica que recoge aquellas referencias que hemos considerado relevantes.

1. Modelado de Sistemas Mediante Sistemas Basados en Reglas Difusas

Es señal de una mente instruida reposar satisfecha con aquel grado de precisión que admite la naturaleza de un tema, y no buscar exactitud allá donde sólo es posible una aproximación de la verdad.

ARISTÓTELES (384–322 A.C.), FILÓSOFO Y CIENTÍFICO GRIEGO

Cada vez que resolvemos un problema correspondiente al mundo real debemos comprender que en realidad solamente estamos encontrando la solución de un modelo del problema [MF00]. Todos los modelos son simplificaciones del mundo real, de otro modo serían tan complejos e intrincados como lo es el escenario natural. El problema que existe con los modelos es que cada uno de ellos debe hacer una serie de asunciones respecto del problema real, en otras palabras, dejar algo de este a un lado. En consecuencia, el error más frecuente que existe cuando tratamos con modelos es olvidar dichas asunciones. En este sentido, un problema puede tener varias soluciones correctas posibles, tan solo dependientes de cómo fue construido el modelo del mismo.

El proceso de resolución de un problema consiste básicamente en dos pasos generales que se dan por separado:

- Creación de un modelo del problema.
- Utilización de ese modelo para la generación de una solución:

Problema \Rightarrow Modelo \Rightarrow Solución.

Tal como hemos dicho, la solución de un problema está dada solamente en términos del modelo y depende de su grado de fidelidad. Esto es, confiamos en que el mejor modelo produzca la mejor solución. Sin embargo, algunas veces los modelos más perfectos resultan inútiles para que un método de solución preciso decida qué hacer, es decir, permita derivar una solución basada en dicho modelo. En las secciones siguientes trataremos con modelos ($Modelo_{\alpha}$) y soluciones ($Solución_{\alpha}$) aproximadas, con el propósito de representar y resolver de la mejor manera posible cierta clase de problemas para los cuales, debido a su nivel de complejidad e imprecisión, resulta difícil obtener una solución adecuada.

En este sentido, la construcción de modelos basados en la lógica difusa ha adquirido un gran interés gracias a que proporciona mayor generalidad, poder expresivo y tolerancia frente a la imprecisión. Por tanto, este capítulo se centrará en el Modelado de Sistemas por medio de Sistemas Basados en Reglas Difusas (SBRDs). Distinguiremos entre dos enfoques de modelado distintos dependiendo de que nuestro principal requisito sea la facilidad de comprensión o la capacidad de aproximación del modelo, respectivamente. Realizaremos un análisis comparativo de ambos enfoques atendiendo al grado de flexibilidad que aporta cada uno de ellos y mostraremos diversas formas de mejorar la precisión.

En este capítulo no incluiremos una descripción detallada de los SBRDs, en el apéndice A puede consultarse un estudio más profundo de los tipos existentes, su composición y funcionamiento.

1.1. Identificación de Sistemas

El desarrollo de modelos matemáticos de sistemas reales es un tópico central en diferentes disciplinas como la ingeniería y las ciencias. Como hemos mencionado, dichos modelos sirven para resolver problemas reales a través de simulaciones, análisis del comportamiento de un sistema, diseño de nuevos procesos para controlar sistemas, predicciones, etc. El desarrollo inadecuado de dichos modelos puede conducir a la obtención de resultados no exitosos y erróneos de los sistemas en cuestión.

Gran parte de estos sistemas actuales 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, posesión de un comportamiento fuertemente no lineal, con alto grado de incertidumbre y características con variaciones en 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.

Tipos de Modelado de Sistemas

A continuación plantearemos al menos tres paradigmas diferentes en cuanto a la tarea de modelado, es decir, de la comprensión de la naturaleza y el comportamiento de un sistema, y la consecuente tarea de creación de un modelo que pueda ser posteriormente utilizado [Bab98]:

- *Modelado de caja blanca* — Tradicionalmente, el modelado de datos se ha visto como una conjunción de una actividad que permite la comprensión de la naturaleza del sistema y su comportamiento, y de un tratamiento matemático adecuado que conduce hacia la realización de un modelo utilizable. Este enfoque, denominado *modelado de «caja blanca»*, está limitado en la práctica cuando se emplean sistemas complejos y poco comprensibles. Como consecuencia directa de este planteamiento, los sistemas no lineales se simplifican, pasando a ser tratados como sistemas lineales, debido a la imposibilidad de comprender su mecanismo de funcionamiento. En vista del diagrama previamente empleado, observamos esta situación de modelado como:

$$\text{Problema} \Rightarrow \text{Modelo}_\alpha \Rightarrow \text{Solución}_p ,$$

lo que corresponde a solucionar un modelo simplificado (α) de un problema por medio de un método (p) que produce soluciones óptimas y precisas.

- *Modelado de caja negra* — Este es un planteamiento diferente que consiste en estimar los parámetros del modelo en base a los datos de entrada disponibles. A pesar de que estos modelos generalmente pueden ser desarrollados en forma bastante fácil, la estructura identificada no posee un significado físico claro. Estos modelos resultan difícilmente escalables y el comportamiento real del sistema puede ser solamente analizado a través de la simulación numérica. Un inconveniente común de la mayoría de estas técnicas de modelado es que no permiten el uso efectivo de información adicional, tal como conocimiento y experiencia de expertos, y operaciones previas, las cuales son imprecisas y cualitativas por naturaleza. Nuevamente, esto puede ser visto como:

$$\text{Problema} \Rightarrow \text{Modelo}_p \Rightarrow \text{Solución}_\alpha .$$

Es decir, un modelo muy preciso solucionado por un método cercano al óptimo, es decir, una solución aproximada.

- *Modelado de caja gris* — Los hechos descritos anteriormente motivan el desarrollo de técnicas de *modelado híbrido o de «caja gris»*, que combinan las ventajas de los enfoques de tipo caja blanca, utilizando conocimiento físico para modelar las partes conocidas del sistema, y de los de caja negra, para aproximar las partes más inciertas. Para realizar esto, se han introducido algunas metodologías inteligentes, las cuales emplean técnicas motivadas por sistemas biológicos e inteligencia humana (lenguaje natural, reglas, redes semánticas o modelos cualitativos) para desarrollar modelos y controladores para sistemas dinámicos.

En consecuencia, obtenemos una solución aproximada de un problema en base a un modelo aproximado, con el propósito de ser interpretable y tan preciso como se pueda. Posteriormente, volveremos a este punto para distinguir varias clases de modelos aproximados, esto es, con distinto grado de precisión (β):

$$\text{Problema} \Rightarrow \text{Modelo}_\beta \Rightarrow \text{Solución}_\alpha .$$

Debemos señalar que un modelo aproximado de un problema no necesariamente es una simplificación del mismo, sino que por el contrario puede ser una descripción aún más precisa que la dada por un modelo de caja blanca. Esto se debe a que estos modelos son capaces de captar la imprecisión e incertidumbre de los problemas del mundo real, que muchas veces son asunciones o excepciones no tenidas en cuenta en el otro caso.

El Modelado de Sistemas con Sistemas Basados en Reglas Difusas

Considerando esta última clase de modelado, 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 forma tal que dicha representación aproximada (β) se encuentre más ajustada a la realidad que otras formas que no puedan

capturar estas cuestiones. A este tipo de modelado nos referiremos mayormente a lo largo de esta memoria.

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

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

Estas reglas permiten establecer una relación lógica entre variables cualitativas de un sistema. Dichos valores cualitativos poseen una interpretación lingüística clara, que se denomina usualmente *etiqueta o término lingüístico*. 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, esto es, la inferencia puede considerarse como un proceso de interpolación entre los resultados de las reglas individuales.

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

- El *modelado mediante SBRDs* 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 con SBRDs desarrollados a partir de datos numéricos, debido a que estas reglas pueden revelar una descripción adecuada del

sistema que generan los datos. Este tipo de descripción puede ser confrontada y posiblemente combinada con conocimiento experto. Todo esto colabora en la comprensión del sistema y en la validación del modelo al mismo tiempo.

- El uso de *términos difusos* 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 considerado (granularidad).

1.2. Modelado de Sistemas con 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 constituye una extensión de los Sistemas Basados en Reglas de la lógica clásica puesto que emplean reglas del tipo comentado 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].

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 a lo largo del capítulo, 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, tal como indicaba Zadeh en su principio de incompatibilidad [Zad73]:

«Al aumentar la complejidad de un sistema, disminuye nuestra capacidad de hacer declaraciones precisas y además significativas acerca de su comportamiento, hasta que se alcanza un umbral más allá del que la precisión y la significancia o relevancia se convierten en rasgos mutuamente excluyentes ».

La decisión de cuán interpretable y de cuán preciso debe ser el modelo a obtener, suele depender de las necesidades que surjan a partir de un problema concreto.

1.2.1. Tipos de Modelado con Sistemas Basados en Reglas Difusas

Podemos distinguir entre dos clases de modelado según el tipo de SBRDs que se emplee:

- *Modelado Lingüístico (ML)* — Este tipo de modelado se realiza generalmente por medio de SBRD lingüísticos (también conocidos como de tipo Mamdani [Mam74, MA75]). En este caso, el principal requisito es la interpretabilidad (*Modelo* _{β_1} en términos de los diagramas de la sección 1.1) 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 ... 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 conjuntos difusos 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 Base de Conocimiento (BC), que consta de una Base de Reglas (BR) —que contiene las reglas en la estructura simbólica mencionada— y una Base de Datos (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.

- *Modelado Difuso (MD)* — 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. 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 ($Modelo_{\beta_2}$ tal que β_2 es más preciso que β_1 , esto es, $\beta_2 > \beta_1$).

La estructura de regla considerada 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.

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 detallan estos sistemas en profundidad.

Existen otros tipos de SBRDs que pueden emplearse también en el modelado de sistemas. Principalmente, podemos destacar los SBRDs de tipo Takagi-Sugeno-Kang [SK88, TS85]. 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, permitimos mayor grado de aproximación. Por tanto, estos sistemas son adecuados para realizar el MD.

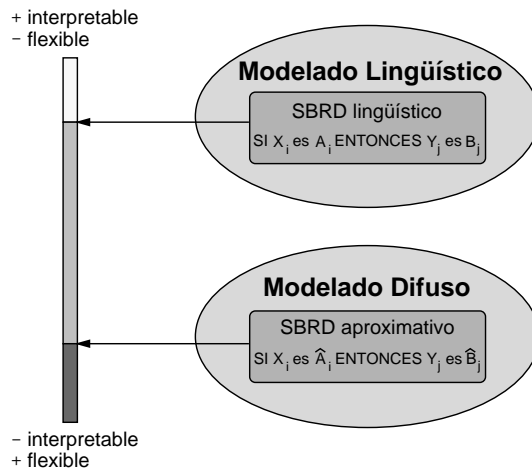


Figura 1.1.: Modelado de sistemas con SBRDs: interpretabilidad y flexibilidad dependiendo del tipo de SBRD considerado

1.2.2. Modelado Lingüístico o Difuso: Interpretabilidad o Precisión

En la figura 1.1 se ilustran los distintos SBRDs comentados en el campo del modelado de sistemas. Como podemos observar, dichos SBRDs tienen 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 la regla, perdiendo así flexibilidad [Bas94]. Por el contrario, relajar tales restricciones, como hacen los SBRDs aproximativos, permite obtener modelos más flexibles pero 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 sobreajustarse 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. En efecto,

¿de qué sirve un modelo con gran flexibilidad si el proceso de diseño tiene serias dificultades para aprovecharla?

Así, mayor flexibilidad no significa mayor precisión y 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).

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

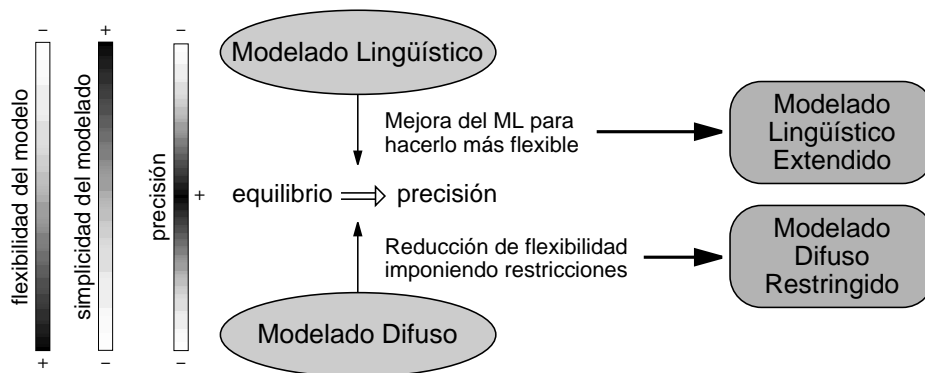


Figura 1.2.: Un compromiso entre flexibilidad del modelo y simplicidad del modelado podría proporcionar mejor precisión. Esto se puede conseguir extendiendo el ML o restringiendo el MD

Para alcanzar este compromiso de flexibilidad y simplicidad, o bien se pueden realizar mejoras al ML para hacer más flexible el aprendizaje y/o la estructura del modelo (*ML extendido*), o bien se pueden imponer restricciones al aprendizaje y/o a la estructura del modelo difuso (*MD restringido*).

Si dicho equilibrio se alcanza partiendo del ML, los modelos obtenidos serán más interpretables que partiendo del MD. Además, podríamos aducir el principio de la navaja de Occam: «*La pluralidad no se debe postular sin necesidad*»¹ [Tho15, Tho18]. O, en nuestro caso, *no complicar el modelo más allá de lo necesario*. Por tanto, parece muy interesante afrontar el modelado de sistemas intentado mejorar, tanto como sea posible, la precisión del ML sin perder su descripción en exceso. Este será el objetivo principal de la presente memoria.

No obstante, antes de analizar en profundidad distintos mecanismos para extender el ML, veremos algunas posibilidades existentes en la literatura para imponer ciertas restricciones al MD. Con este análisis, podremos comprender mejor las distintas perspectivas existentes dentro del modelado de sistemas usando SBRDs para alcanzar la relación precisión-interpretabilidad deseada.

1.3. Modelado Difuso Restringido

El proceso de MD puede restringirse en la fase de derivación de las reglas difusas de diferentes formas. Una de las más empleadas consiste en imponer ciertas restricciones a los conjuntos difusos que se van generando.

Según este enfoque, podemos distinguir entre dos tipos de métodos de aprendizaje (en la figura 1.3 se muestra gráficamente la clasificación mencionada) [ACCH99]:

- *Aprendizaje no restringido* (ANR) — Un proceso de aprendizaje de SBRDs aproximativo es *no restringido* cuando no se imponen restricciones a los conjuntos difusos empleados en las diferentes reglas y se les permite situarse en cualquier región del dominio de las variables correspondientes. Ejemplos de este tipo de métodos son los propuestos en [CFM96, CV94, DGM97, HLV98a].
- *Aprendizaje restringido* (AR) — Por el contrario, un proceso de aprendizaje es *restringido* cuando se imponen limitaciones que obligan a los conjuntos difusos a situarse en algunos intervalos concretos durante la fase de aprendizaje. Dentro de este tipo, podemos distinguir dos posibilidades:

- *Aprendizaje restringido estricto* (ARE) — Existen unos intervalos de variación que determinan la región en la que cada parámetro que define

¹ «*Pluralitas non est ponenda sine neccesitate*», William de Ockham (c. 1285–c. 1349), filósofo medieval inglés y monje franciscano

las funciones de pertenencia puede tomar valores durante el proceso de aprendizaje. Por ejemplo, en [BD95, CH01] se proponen métodos que siguen este enfoque.

- *Aprendizaje restringido relajado (ARR)* — En este segundo caso, la única restricción impuesta a las localizaciones y formas de las funciones de pertenencia es encontrarse en un intervalo concreto y tener una forma coherente. Por ejemplo, el método propuesto en [CH97] realiza este tipo de aprendizaje.

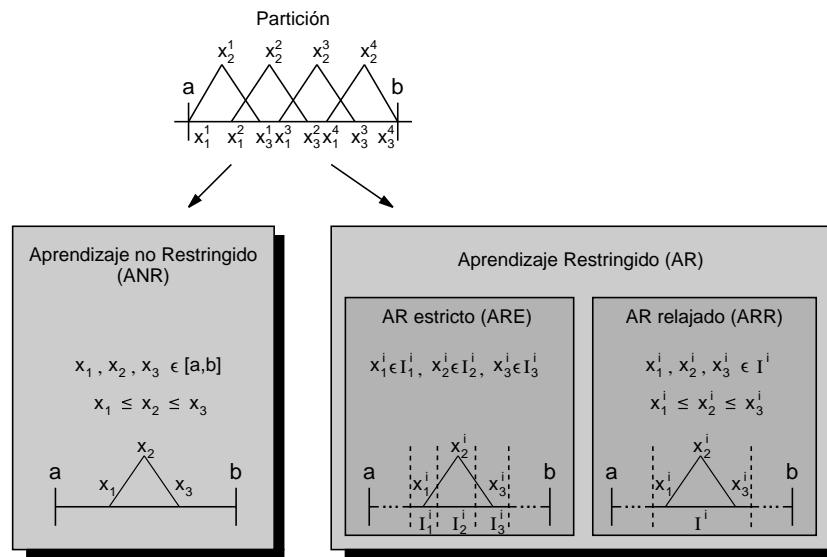


Figura 1.3.: Aprendizaje con o sin restricciones en el MD

Intuitivamente, podemos observar que los tipos de aprendizaje mencionados están relacionados y podemos considerar al ANR como el caso más extremo de AR donde el intervalo de variación asociado a cada conjunto difuso corresponde al dominio completo de la variable del sistema. De la misma forma, el ARR se puede considerar un caso particular del ARE donde se emplea el mismo intervalo para todos los intervalos de variación de los parámetros de la función de pertenencia. No obstante, para evitar confusiones, clasificaremos cada proceso de aprendizaje atendiendo a la definición más específica, es decir, la más restrictiva.

Generalmente, la información para determinar las restricciones se obtiene a partir

de particiones difusas preliminares de las variables del problema. De esta forma, los intervalos de variación (para el AR) se pueden calcular a partir de los parámetros de las funciones de pertenencia iniciales.

El grado de libertad es la principal propiedad que caracteriza a los diferentes tipos de aprendizaje introducidos. Mientras que el ANR es totalmente libre para determinar los conjuntos difusos a lo largo del universo de discurso completo, el ARE los genera en espacios muy confinados. Claramente, existe otro aspecto relacionado con esta propiedad, la dimensión del espacio de búsqueda. Cuanto mayor es el grado de libertad, mayor será también el espacio de búsqueda. Una mayor libertad permite al modelo difuso obtenido sacar partido de la alta capacidad de aproximación de los SBRDs aproximativos pero sin embargo, la complejidad para llevar a cabo el proceso de aprendizaje crece debido al enorme espacio de búsqueda que se maneja. El enfoque de ARR realiza el mejor equilibrio entre ambos aspectos: generación de conjuntos difusos con un alto grado de libertad y reducción del espacio de búsqueda. Por las razones de flexibilidad argumentadas en la sección anterior, este balance podría proporcionar mejor precisión [ACCH99].

1.4. Modelado Lingüístico Extendido

Como hemos comentado con anterioridad, el ML desarrollado por los SBRDs lingüísticos presenta cierta inflexibilidad debido 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 el 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 altamente dimensionales.
- 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 tan rígido efectuado sobre el espacio de entrada.

Sin embargo, se pueden realizar algunas mejoras para afrontar estas limitaciones. Básicamente, distinguiremos dos tipos distintos de mejoras según se realicen en

- el *proceso de derivación del modelo*, extendiendo su diseño a otros componentes como, por ejemplo, la BD [BKC96, CdJH98, CH97, GP99, Kar91, LM94, SKG93]), o en
- la *estructura del modelo*, cambiándola ligeramente para hacerla más flexible y permitir así incrementar la precisión [CH97, CH00, NIT97, PP99].

Generalmente, todas estas mejoras tienen el objetivo final de favorecer el *razonamiento interpolativo* que desarrolla el SBRD. 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.

1.4.1. Mejoras al Proceso de Derivación del Modelo Lingüístico

Los métodos de ML básicos 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 con SBRDs. El procedimiento generalmente consiste en, o bien determinar la forma más adecuada de 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.

Según cuándo y cómo se realice el diseño de la BD, podemos distinguir entre cuatro tipos distintos:

- *Diseño preliminar* — Consiste en extraer la BD por inducción a partir del conjunto de ejemplos disponible. Tradicionalmente, este proceso se ha realizado con técnicas de agrupamiento no supervisado [LMV99].
- *Diseño incrustado* — Recientemente, ha surgido un nuevo enfoque que realiza el procedimiento con un método de aprendizaje básico incrustado [CHV01, Glo96, IM96]. Esta técnica consiste en disponer de un método de aprendizaje sencillo que diseña, a partir de una BD determinada, otros componentes distintos de esta (v.gr., la BR). Mediante un proceso de meta-aprendizaje, el método genera distintas BDs y prueba su eficacia ejecutando el método de aprendizaje básico.
- *Diseño simultáneo* — El proceso de diseño de la BD se realiza conjuntamente con otros componentes tales como la BR en un procedimiento simultáneo [KS95, LM94, MMH97, WHT98].
- *Diseño posterior* — 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. Generalmente, el ajuste consiste en variar la forma de las funciones de pertenencia [BKC96, CdJH98, CH97, GP99, Kar91, LM94, 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].

Por otro lado, podemos distinguir entre dos enfoques distintos según las operaciones efectuadas sobre las funciones de pertenencia en el proceso de definición de la BD:

- *Variaciones lineales* — Consiste en alterar los valores de los distintos parámetros que definen la función de pertenencia para realizar desplazamientos y/o ensanchamientos de los conjuntos difusos. Esto se puede conseguir ajustando directamente cada uno de los parámetros [BKC96, CH97, Kar91, KS95, LM94, MMH97, SKG93, WHT98] o usando diferentes factores de escala lineales [CHV01, MMH97]. Por ejemplo, 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 figura 1.4(a)), afectando así al comportamiento del SBRD.

- *Variaciones no lineales* — Otra posibilidad consiste en extender la definición de las funciones de pertenencia empleando expresiones más flexibles que permitan variar el grado de compatibilidad de los conjuntos difusos de forma no lineal. Por ejemplo, podemos elevar el grado de pertenencia a la potencia de un parámetro real positivo α [CdJH98, GP99], es decir,

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

En este caso, el proceso de ajuste supondría obtener el valor más adecuado para este parámetro adicional α para mejorar el rendimiento del SBRD. La figura 1.4(b) muestra el efecto obtenido considerando distintos valores de α . Se pueden obtener efectos similares aplicando factores de escala no lineales a la partición difusa completa [PGG97]. En algunas ocasiones, incluso se emplea un único factor de escala no lineal para toda la BD [NIT97].

1.4.2. Extensiones de la Estructura del Modelo Lingüístico

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

1.4.2.1. 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

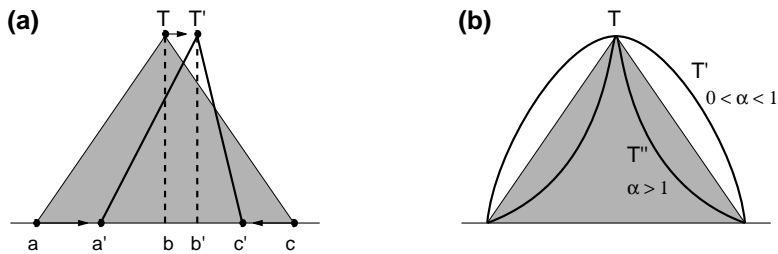


Figura 1.4.: Dos enfoques para variar la forma de una función de pertenencia: (a) alterar los parámetros que la definen, (b) ajustar una función de escala no lineal

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 sobre una función de pertenencia triangular se muestran en la figura 1.5.

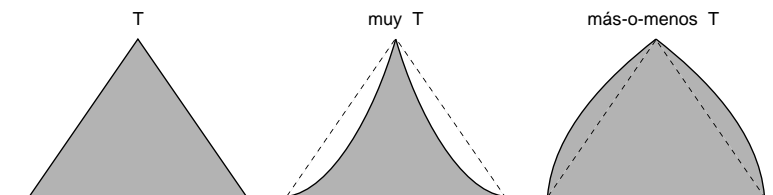


Figura 1.5.: Efectos de los modificadores lingüísticos ‘muy’ y ‘más-o-menos’

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

$$\text{SI } X_1 \text{ es } ml_{X_1} A_1 \text{ y } \dots \text{ y } X_n \text{ es } ml_{X_n} A_n \text{ ENTONCES } Y \text{ es } ml_Y B,$$

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*.

En la sección 3.2.1 del capítulo 3 se detalla el efecto de este tipo de estructura y se introducen más posibilidades de modificadores lingüísticos.

1.4.2.2. 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 esta estructura son las reglas de doble consecuente (es decir, $c = 2$), propuestas recientemente en [CH00, NIT97].

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

1.4.2.3. Uso de Reglas Ponderadas

El uso de reglas ponderadas consiste en emplear un parámetro adicional para cada regla que indica su grado de importancia en el proceso de inferencia [CP00, NIT97, PP99, YB94], en lugar de considerar todas las reglas con la misma importancia como en el caso usual. De este modo se consigue más flexibilidad para mejorar el razonamiento interpolativo y, por ende, el rendimiento del modelo. La estructura de regla es la siguiente:

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

siendo w un valor real que indica el peso de la regla. En este enfoque es necesario algunos cambios en el sistema de inferencia clásico para considerar la acción ponderada de cada regla.

El operador *con*, que liga un peso a una regla, se puede definir de diferentes formas. Una de las opciones más usuales es multiplicar el grado de emparejamiento del antecedente por el correspondiente peso antes de aplicar el operador de implicación, que relaciona el antecedente con el consecuente. Otra posibilidad sería cambiar la conclusión derivada del operador de implicación atendiendo al correspondiente peso (por ejemplo, cambiar el soporte del conjunto difuso obtenido).

1.4.3. Interpretabilidad del Modelado 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 lingüísticos obtenidos. La figura 1.6 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 teóricamente más interpretable. Naturalmente, este orden es ambiguo y el único propósito de la figura 1.6 es formar al lector una idea de la consecuencia en la pérdida de interpretabilidad sufrida al aplicar las diferentes extensiones al ML.

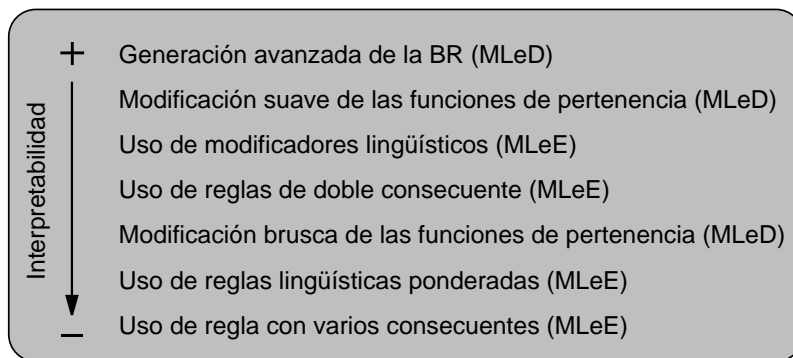


Figura 1.6.: Pérdida de interpretabilidad dependiendo de la extensión realizada en el ML —MLeD indica ML extendido en el proceso de derivación y MLeE indica ML extendido en la estructura del modelo—

- *Generación avanzada de la BR* — Obviamente, esta extensión no supone ninguna pérdida de interpretabilidad sobre el ML clásico, ya que se centra en realizar un proceso de derivación sofisticado sin alterar ni la BD ni la estructura del modelo.
- *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 algo grado de interpretabilidad se pueden imponer ciertas restricciones [Val99] 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 [Kar91, LM94]. Estas restricciones hacen menos flexible el proceso de derivación pero, además de facilitar la legibilidad, reducen el riesgo de sobreajustar el problema.

- *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 ML 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 perturbar la interpretación del modelo, perdiendo así legibilidad. La figura 1.7 muestra un ejemplo donde una alteración excesiva de los parámetros de las funciones de pertenencia puede implicar perder interpretabilidad.

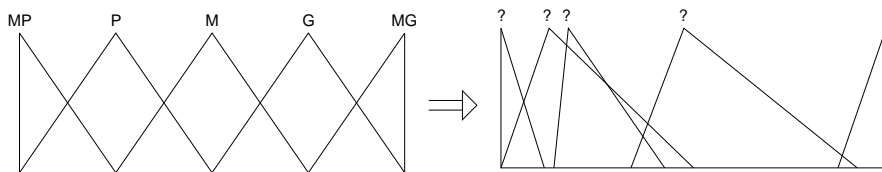


Figura 1.7.: Se pierde cierta interpretabilidad al hacer una modificación brusca de las funciones de pertenencia

- *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].
- *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.

1.5. Estudio Experimental del Modelado Lingüístico Extendido frente al Modelado Difuso

En esta sección vamos a realizar un estudio experimental para analizar el comportamiento de las diferentes propuestas de ML extendido mencionadas con anterioridad frente al MD. Consideraremos varios métodos de derivación representativos que desarrollan el ML, el ML extendido, el MD restringido y el MD con SBRDs lingüísticos y aproximativos. La tabla 1.1 muestra las características principales de los métodos considerados, que serán descritos brevemente en el siguiente subapartado.

Para analizar el comportamiento de estos métodos de derivación vamos a considerar cuatro aplicaciones de distinta complejidad (dos problemas de «laboratorio» y dos problemas reales): modelado de la función tridimensional F_2 , modelado de la función tridimensional F_3 , estimación de la calidad del arroz cocido y estimación de la longitud de línea eléctrica de baja tensión en zonas rurales. En el apéndice C se puede encontrar una descripción detallada de cada uno de estos problemas.

En principio, la aplicación de los SBRDs aproximativos a problemas sencillos como F_2 o de los SBRDs lingüísticos a problemas complejos como F_3 podría parecer incoherente, ya que no están diseñados respectivamente para tales casos [ACCH99]. Sin embargo, hemos considerado apropiado emplear estas aplicaciones con la intención de analizar el comportamiento comparativo de tales sistemas resolviendo problemas de diferente naturaleza.

En los subapartados siguientes mostramos una breve descripción de los métodos considerados en este estudio experimental, los resultados obtenidos y el análisis correspondiente.

Tabla 1.1.: Resumen de los métodos de derivación considerados y sus principales características

Ref.	Método	Tipo de Modelado	Tipo de Algoritmo
[WM92] [Thr91]	WM T	ML ML	MEEE AG
[CH97] [LM94] [CH97]	M-L-Ajus LM M-L	MLe MLe MLe	AG AG MEEE+AG
— [CH00] [NIT97] —	WM-LH ALM NIT WM-RP	MLe MLe MLe MLe	AG MEEE+AG MEEE AG
[BD95] [CH01] [CH97] [CFM96]	WCA M-AARE M-AARR P-FCSI	MDR MDR MDR MD	MEEE AG AG AG

ED	EE	DS	DE	FPL	FPN	LH	VC	DC	RP
✓	✓		✓	✓					
✓	✓	✓	✓	✓			✓		
✓	✓	✓	✓	✓					
✓	✓		✓		✓			✓ _a	✓
✓	✓				✓			✓ _t	✓
✓	✓								✓

Restricciones en el Aprendizaje	
[BD95]	WCA
[CH01]	M-AARE
[CH97]	M-AARR
[CFM96]	P-FCSI

- **MLe** = ML extendido.
- **MDR** = MD restringido.
- **MEEE** = método específico de envoltura de ejemplos.
- **AG** = método basado en algoritmos genéticos.
- **ED** = ML extendido en la derivación.
- **EE** = ML extendido en la estructura.
- **DS** = derivación simultánea.
- **DE** = derivación secuencial por etapas.
- **FPL** = funciones de pertenencia lineales.
- **FPN** = funciones de pertenencia no lineales.
- **LH** = modificadores lingüísticos (del inglés *linguistic hedges*).
- **VC** = varios consecuentes por regla.
- **DC** = reglas de doble consecuente.
- **RP** = reglas ponderadas.
- **✓_a** = algunas reglas de la BR tienen dos consecuentes.
- **✓_t** = todas las reglas de la BR tienen dos consecuentes.

1.5.1. Métodos de Derivación de Reglas Difusas Analizados

En esta sección vamos a introducir una breve descripción de los diferentes métodos de aprendizaje que se han considerado en el análisis experimental para realizar el ML, ML extendido y MD. En [ACCH99, ACC⁺00] puede consultarse una descripción más detallada de muchos de estos métodos.

1.5.1.1. Métodos de Derivación de Modelos Lingüísticos

Consideraremos dos métodos específicos para derivar modelos lingüísticos:

- En [WM92], Wang y Mendel proponen un método específico de envoltura de ejemplos para generar reglas lingüísticas (*método WM*). El algoritmo consiste en dividir los espacios de entrada y salida en regiones difusas, generar la regla lingüística que mejor envuelve a cada ejemplo y, finalmente, para cada región de entrada difusa, seleccionar entre aquellas reglas con antecedente común (reglas conflictivas) la que mayor grado de importancia tenga (calculado según el grado de envoltura). En el capítulo 2 se describe el funcionamiento de este algoritmo con mayor detalle.
- El método propuesto por Thrift (*método T*) [Thr91] se basa en un algoritmo genético (AG) que codifica en el cromosoma todas las celdas de una tabla de decisión completa. De esta forma, el método establece una correspondencia entre el conjunto de términos lingüísticos asociados a la variable de salida y un conjunto de enteros ordenados (conteniendo un elemento que codifica el valor nulo) que representa el conjunto de alelos. Cada uno de los cromosomas se construye uniendo la codificación parcial asociada a cada una de las etiquetas lingüísticas contenidas en las celdas de la tabla de decisión. Un gen conteniendo el alelo nulo representará la ausencia de la regla lingüística localizada en la correspondiente celda de la BR. Se proponen operadores de reproducción adecuados para tratar esta representación entera.

1.5.1.2. Métodos de Derivación de Modelos Lingüísticos Extendidos

Consideraremos siete métodos específicos para extender el ML de diferentes formas. Estos métodos cubren los distintos enfoques presentados en la sección 1.4 como se muestra en la tabla 1.1: un método para realizar sólo un ajuste *a posteriori*; un método que aprende la BD y BR al mismo tiempo; un método completo con procesos

de aprendizaje de BR y ajuste *a posteriori* donde se permiten varios consecuentes por regla; un ajuste *a posteriori* de la BR empleando modificadores lingüísticos; un método que considera la estructura de reglas con doble consecuente; un método con reglas de doble consecuente ponderadas y una factor de escala no lineal de la BD completa; y un método para aprender reglas ponderadas.

- El proceso de ajuste *a posteriori* introducido en [CH97] por Cordón y Herrera (*método M-L-Ajus*) se basa en un AG donde cada cromosoma que forma la población genética codifica una definición completa de la BD que será combinada con la BR existente para evaluar la adaptación del individuo. Puesto que cada función de pertenencia triangular tiene una representación paramétrica asociada basada en una 3-tupla de valores reales, la partición difusa primaria se puede representar mediante un vector compuesto por $3 \cdot N_l$ valores reales, siendo N_l el número de términos de la variable lingüística. La BD se codifica en un cromosoma de longitud fija con codificación real construido por la unión de las representaciones parciales de cada una de las particiones de las variables difusas. Se asocia un intervalo de variación a cada gen (parámetro de una función de pertenencia) en el cromosoma. La función de adecuación penaliza la falta de completitud y se emplean operadores genéticos apropiados.
- Liska y Melsheimer presentan en [LM94] un método genético de aprendizaje (*método LM*) que realiza un proceso simultáneo de diseño de la BR y la BD. La BC se representa como un cromosoma compuesto por tres subcadenas: una de codificación real con las funciones de pertenencia asociadas a todas las variables del sistema, representándose cada una de ellas con dos parámetros (centro y ancho); otra subcadena con codificación entera que contiene la estructura de cada regla, es decir, los términos usados en cada una de las variables, con la posibilidad de eliminar alguna variable si se aparece el término nulo; y una tercera subcadena, también con codificación entera, que contiene el consecuente asociado a cada regla, también considerando la posibilidad de eliminar una regla si el correspondiente consecuente es el valor nulo. Los operadores genéticos (cruce y mutación) actuarán de diferente forma dependiendo de la subcadena del cromosoma donde se aplique. La adecuación de cada cromosoma en la población se evalúa con una técnica de clasificación exponencial basada en la función de error. Una vez que el proceso de aprendizaje genético ha finalizado, el método aplica una segunda fase de ajuste de las funciones de pertenencia para asegurar el ajuste óptimo.

- El método para generar SBRDs lingüísticos propuesto por Cordon y Herrera (*método M-L*) [CH97] sigue el paradigma de derivación MOGUL presentado en [CdJHL99]. El método se compone de tres etapas: un *proceso de generación de la BR iterativo*, un *proceso de simplificación genético* y un *proceso de ajuste genético* (esta tercera etapa es el método M-L-Ajus mencionado anteriormente). En la primera etapa, la BR se deriva regla por regla, seleccionando la más precisa en cada paso del algoritmo. Una vez que esta regla se ha obtenido, se calcula su envoltura sobre el conjunto de ejemplos de entrenamiento. Aquellos ejemplos cubiertos a un grado mayor que un valor específico se eliminan del conjunto de entrenamiento. Cada vez que la mejor regla debe seleccionarse, la precisión de las reglas candidatas se calcula empleando una función de adecuación multicriterio que permite al método asegurar un grado de completitud y consistencia en la BC finalmente generada. Ya que la misma combinación de antecedentes se puede generar para varias reglas, el método hace implícitamente más flexible la estructura del modelo permitiendo la existencia de reglas con varios consecuentes.

La segunda etapa, la simplificación (propuesta en [HLV98a]), combina reglas y elimina reglas redundantes, seleccionando el conjunto más cooperativo. Se basa en un AG con codificación binaria (un bit para cada regla perteneciente a la BR previa) donde cada gen indica la consideración o no de la correspondiente regla para formar parte de la BR final. Se emplean operadores de selección y recombinación apropiados. El método usa la misma función de adecuación considerada en el método M-L-Ajus.

- Consideraremos un método que añade modificadores lingüísticos a la BR previamente diseñada por WM (*método WM-LH*²). De este modo, mediante un AG simple se añadirán los operadores ‘muy’ y ‘más-o-menos’ para modificar los términos lingüísticos implicados en cada regla difusa. Para ello, el esquema de codificación genera cromosomas con representación entera de longitud $N_r \cdot (n + 1)$ (siendo N_r el número de reglas aprendidas por WM y n el número de variables de entrada). Cada gen toma tres alelos posibles —0, 1 ó 2— que indican respectivamente las tres posibilidades para cada función de pertenencia —usar el modificador lingüístico ‘más-o-menos’, no usar modificador o usar

² Hemos empleado las siglas LH, del inglés *linguistic hedges*, para distinguirlas de las consideradas al designar el Modelado Lingüístico (ML)

el modificador lingüístico ‘muy’, respectivamente.

La población inicial se genera introduciendo un cromosoma que represente la BR previamente obtenida, es decir, todos los genes con el alelo 1. El resto de cromosomas se generan aleatoriamente. Se emplea el operador de cruce multipunto en dos puntos. El operador de mutación cambia el gen al alelo 1 cuando se debe mutar un gen con alelos 0 ó 2, y aleatoriamente a 0 ó 2 cuando se debe mutar un gen con alelo 1. Se emplea el muestro universal estocástico de Baker con un esquema de selección elitista y la función de adecuación del error cuadrático medio (véase el apéndice C).

- El método propuesto por Cordón y Herrera según la metodología de modelado lingüístico preciso (*método ALM*) [CH00] se basa en los siguientes dos aspectos. En primer lugar, la estructura del modelo se extiende para permitir que una combinación de antecedentes tenga dos consecuentes asociados cuando sea necesario. Por tanto, se obtienen dos reglas, la primaria y secundaria en importancia, en cada combinación considerando un proceso de generación específico. En esta memoria, emplearemos el proceso de generación basado en el método WM. Entonces, después de descomponer cada regla con doble consecuente en dos reglas simples independientes, se emplea el proceso de selección propuesto en [HLV98a] y previamente descrito en el método M-L para seleccionar el subconjunto de reglas con mejor cooperación. Gracias a este proceso sólo se emplearán reglas con doble consecuente cuando sea necesario.
- La propuesta de Nozaki, Ishibuchi y Tanaka (*método NIT*) en [NIT97] también emplea dos consecuentes en cada combinación de antecedentes para mejorar el rendimiento. En este caso, además de usar siempre reglas con doble consecuente, estas reglas se ponderan para considerar diferentes grados de importancia durante el proceso de inferencia. Nos extenderemos algo más en la descripción de este método ya que lo consideraremos como comparación en los estudios experimentales del capítulo 2. El método se compone de dos fases:
 1. *Primera fase, generación de reglas TSK simplificado* — En un primer paso, se aprende una BC cuyas reglas contienen números reales en lugar de términos lingüísticos en la variable de salida (es decir, del tipo TSK simplificado). Para determinar el consecuente real de cada regla, se define un peso para cada vector de datos de entrenamiento como el producto de

los grados de pertenencia de la entrada, elevados cada uno de ellos a la potencia de α (un parámetro que define una función de escalado no lineal).

El consecuente real se obtendrá como la media ponderada del valor de salida conocido asociado a cada vector de ejemplos. Si el peso (grado de emparejamiento) es cero, la regla no se considerará.

2. *Segunda fase, conversión a reglas lingüísticas* — En este paso, se traduce la representación con consecuente real a reglas lingüísticas manteniendo el mismo grado de precisión. Esto se consigue gracias al uso de dos BRs lingüísticas: la primaria y la secundaria. En primer lugar, es necesario realizar una partición difusa del universo de discurso del consecuente en varios términos lingüísticos. Los autores consideran particiones distribuidas uniformemente con funciones de pertenencia triangulares.

A continuación, se calcula el grado de pertenencia del consecuente real (p_0) a cada conjunto difuso de la partición difusa de la variable de salida. La etiqueta lingüística asociada al conjunto difuso con el grado de pertenencia mayor (T_3) se considera para componer la regla lingüística de la BR primaria, mientras que la etiqueta lingüística con el siguiente valor de pertenencia más alto (T_2) se considerará para la BR secundaria. La figura 1.8 clarifica gráficamente el funcionamiento de esta fase.

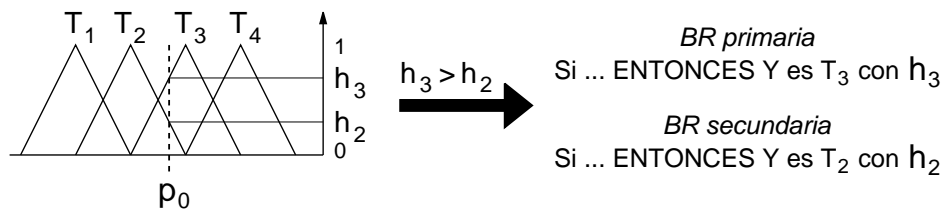


Figura 1.8.: Obtención de reglas lingüísticas a partir de reglas tipo TSK simplificado en el método NIT

Además, se consideran los grados de pertenencia como pesos de la correspondiente regla (h_3 y h_2 en la figura). El sistema de inferencia considerado hace uso de estos pesos para sacar provecho de las dos BRs.

- Por último, consideraremos también un método de aprendizaje de reglas ponderadas que añade pesos a la BR diseñada por el método de WM (*método*

WM-RP). La búsqueda de los pesos se realiza con un AG. El operador *con*, que asocia el peso a la correspondiente regla, se aplicará multiplicando el peso por el grado de emparejamiento del antecedente.

Cada cromosoma es una cadena de codificación real con una longitud igual al número de reglas. Cada gen toma un valor en el rango $[0, 1]$ para representar el peso considerado en la correspondiente regla. La población inicial se genera introduciendo un cromosoma en el que todos los genes tienen el valor 1 (BR original) y con el resto de cromosomas generados aleatoriamente. Se usa el operador de cruce multipunto en dos puntos. La mutación simplemente consiste en asignar al gen seleccionado un valor aleatorio generado en el rango $[0, 1]$. Se emplea el muestro universal estocástico de Baker con un esquema de selección elitista y la función de adecuación del ECM (véase el apéndice C).

1.5.1.3. Métodos de Derivación de Modelos Difusos Aproximativos

Consideraremos cuatro métodos de aprendizaje para generar SBRDs aproximativos, tres de los cuales realizan el MD imponiendo diferentes grados de restricciones a las funciones de pertenencia.

- El *algoritmo de conteo ponderado (método WCA)*, propuesto por Bárdossy y Duckstein en [BD95], se basa en el principio de generar en primer lugar los antecedentes de las reglas y obtener entonces los consecuentes tomando como base los ejemplos de entrenamiento que tengan un grado de emparejamiento alto con el antecedente de la regla. Se requiere una definición previa de los conjuntos soporte del antecedente difuso así como el número de reglas. El algoritmo genera el vértice de las funciones triangulares determinando así las formas de los conjuntos difusos usados por los antecedentes, y posteriormente identifica los correspondientes consecuentes. Debido a esta forma de proceder, el método establece una restricción estricta en el antecedente mientras que el consecuente se obtiene sin restricciones.
- Consideraremos dos métodos diferentes propuestos por Cordón y Herrera, los métodos de aprendizaje aproximativo restringido relajado (método M-AARR) [CH97] y restringido estricto (método M-AARE) [CH01]. Estos métodos siguen el paradigma MOGUL [CdJHL99] mencionado en el método M-L (sección 1.5.1.2), siendo la primera etapa un *proceso de generación evolutiva* y

manteniendo el mismo proceso de simplificación. La fase de ajuste, originalmente propuesta en [HLV95], se realiza de diferente forma al método M-L-Ajus, ya que no se consideran semánticas globales en los SBRDs aproximativos y, por tanto, cada conjunto difuso implicado en cada regla difusa se ajusta independientemente.

En el *método M-AARR*, el proceso de generación se inspira en el método M-L comentado con anterioridad. A diferencia de este, una vez que se obtiene una regla lingüística, una estrategia evolutiva (1+1) la ajusta localmente haciendo que tome una naturaleza aproximativa. Se consideran intervalos de variación para cada conjunto difuso y, por tanto, se imponen restricciones relajadas al proceso de aprendizaje.

- Por el contrario, el *método M-AARE* realiza un proceso de generación diferente basado en un AG con restricciones estrictas donde se considera un intervalo de variación para cada parámetro que define las funciones de pertenencia. Esto se realiza gracias al uso de operadores genéticos especiales que mantienen las restricciones.
- El *sistema clasificador difuso estilo Pittsburgh #1 (método P-FCSI)*, propuesto por Carse, Fogarty y Munro [CFM96], se basa en un AG caracterizado por un esquema de codificación y un operador de cruce muy interesantes. La representación de regla empleada en este método es un conjunto de términos que codifican los centros y las anchuras de las funciones de pertenencia de los conjuntos difusos sobre el rango de las variables de entrada y salida siguiendo un esquema de codificación real. Un cromosoma será una cadena concatenada de longitud variable de tales reglas difusas. El operador de cruce se basa en el clásico cruce en dos puntos pero con una consideración n -dimensional (siendo n el número de variables de entrada). Se emplean igualmente un operador de mutación así como operadores para crear y eliminar reglas. El método no considera ninguna restricción sobre los parámetros de los conjuntos difusos ya que la generación de la población inicial y los operadores genéticos generan libremente estos valores, realizando por tanto un proceso de aprendizaje no restringido.

1.5.2. Experimentos

Como hemos comentado, el método M-L-Ajus será considerado como un proceso de ajuste *a posteriori* independiente del método de aprendizaje empleado para obtener la BC previa. Por tanto, en los experimentos desarrollados, este método de ajuste será aplicado a las BCs previamente generadas por los dos métodos de ML (el método WM y el método T).

Las tablas 1.2 y 1.3 recogen los resultados obtenidos con los métodos y aplicaciones considerados en este estudio. En dicha tabla, #R indica el número de reglas y ECM_{entr} y ECM_{prue} los valores del error cuadrático medio (véase el apéndice C) sobre los conjuntos de entrenamiento y prueba. En el problema del arroz, los valores mostrados en las columnas se han calculado como la media de los diez modelos generados para cada método (véase la sección C.2.4 del apéndice C). En el caso de los métodos con varios consecuentes por regla (los métodos M-L, ALM y NIT), el número de reglas mostrado es el correspondiente tras descomponerlas en reglas simples.

Se ha incluido también una columna indicando el tipo de modelado realizado por cada método analizado. Mostraremos en **negrita** los mejores resultados en cada grupo (ML, ML extendido y MD) para cada problema y en **negrita-cursiva** si dicho valor es, además, el mejor valor obtenido de entre todos los métodos analizados.

1.5.3. Análisis de Resultados

A continuación se realizará un análisis de los resultados obtenidos considerando el comportamiento de los diferentes métodos al resolver los distintos tipos de problemas. Atenderemos a tres puntos de vista distintos: ML frente a MD, ML frente a ML extendido y ML extendido frente a MD. En cada caso tendremos en cuenta aspectos de precisión e interpretabilidad.

También consideraremos un criterio global que combine ambos aspectos y presentaremos diferentes recomendaciones sobre la mejor elección dependiendo de la dificultad del problema (según la tabla C.1 del apéndice C, página 203). Por supuesto, no podemos generalizar las conclusiones obtenidas en una aplicación específica para todos los problemas con complejidad similar, de modo que estas recomendaciones deberían considerarse como simples directivas derivadas de la experimentación realizada.

Tabla 1.2.: Resultados obtenidos por los métodos de ML, ML extendido y MD analizados en las dos funciones tridimensionales

Modelado	Método	F_2			F_3		
		#R	ECM_{ent}	ECM_{pntc}	#R	ECM_{ent}	ECM_{pntc}
ML	WM	49	0,194386	0,044466	49	1,82425	2,099754
ML	T	47	0,067518	0,032442	48	0,503970	0,547687
MLeD	WM + M-L-Ajus	49	0,060296	0,028621	49	0,882026	1,024698
MLeD	T + M-L-Ajus	49	0,030928	0,021698	49	0,403650	0,432996
MLeD	LM	43	0,105432	0,174930	49	0,330614	0,370192
MLeDE	M-L	96	0,011865	0,014518	253	0,370366	0,375440
MLeE	WM-LH	49	0,071854	0,026771	49	1,563712	1,799811
MLeE	ALM	55	0,019083	0,026261	32	0,548304	0,632155
MLeE	NIT	98	0,074073	0,015948	98	0,481266	0,519737
MLeE	WM-RP	49	0,017589	0,022608	49	1,352281	1,581560
MDR	WCA	49	0,288179	0,138663	49	0,494001	0,536226
MDR	M-AARE	55	0,092998	0,066488	236	0,177442	0,183578
MDR	M-AARR	86	0,056008	0,055712	239	0,173270	0,168424
MD	P-FCSI	42	0,044838	0,038075	74	0,220314	0,222865

Tabla 1.3.: Resultados obtenidos por los métodos de ML, ML extendido y MD analizados en los dos problemas reales

Modelado	Método	Arroz.			Línea de baja tensión		
		#R	ECM _{entr}	ECM _{prue}	#R	ECM _{entr}	ECM _{prue}
ML	WM	15	0,013284	0,013119	24	222.654	239.962
ML	T	16	0,004949	0,005991	49	169.077	175.739
MLeD	WM + M-L-Ajus	15	0,001106	0,002138	24	145.273	171.998
MLeD	T + M-L-Ajus	16	0,001153	0,002927	49	147.844	164.993
MLeD	LM	31	0,000810	0,002342	49	167.014	167.383
MLeDE	M-L	6	0,001075	0,002331	35	137.905	173.096
MLeE	WM-LH	15	0,001758	0,001731	24	181.609	209.756
MLeE	ALM	5	0,003416	0,003984	20	155.866	178.601
MLeE	NIT	64	0,002996	0,003520	64	173.230	190.808
MLeE	WM-RP	15	0,003224	0,003109	24	191.420	233.592
MDR	WCA	32	0,017729	0,020026	34	223.985	213.906
MDR	M-AARE	5	0,001432	0,006094	16	149.129	168.210
MDR	M-AARR	4	0,003468	0,005290	20	142.109	166.579
MD	P-FCSI	51	0,000596	0,009055	71	141.023	197.941

1.5.3.1. Modelado Lingüístico frente a Modelado Difuso

- *Precisión:* Mientras que el ML obtiene mejores resultados de precisión en F_2 , el MD obtiene modelos más precisos al resolver un problema complejo como F_3 gracias al uso de semánticas locales que introducen grados adicionales de libertad en los modelos difusos. Por otro lado, ML y MD obtienen resultados de precisión similares en el problema del arroz mientras que el MD obtiene modelos algo más precisos en el problema eléctrico. Sin embargo, los SBRDs aproximativos generados en el MD muestran, en los dos problemas de aplicación real, una tendencia clara al sobreajuste (grados de aproximación notablemente superiores que los de generalización) debido a la alta capacidad de aproximación que caracteriza a estos sistemas.

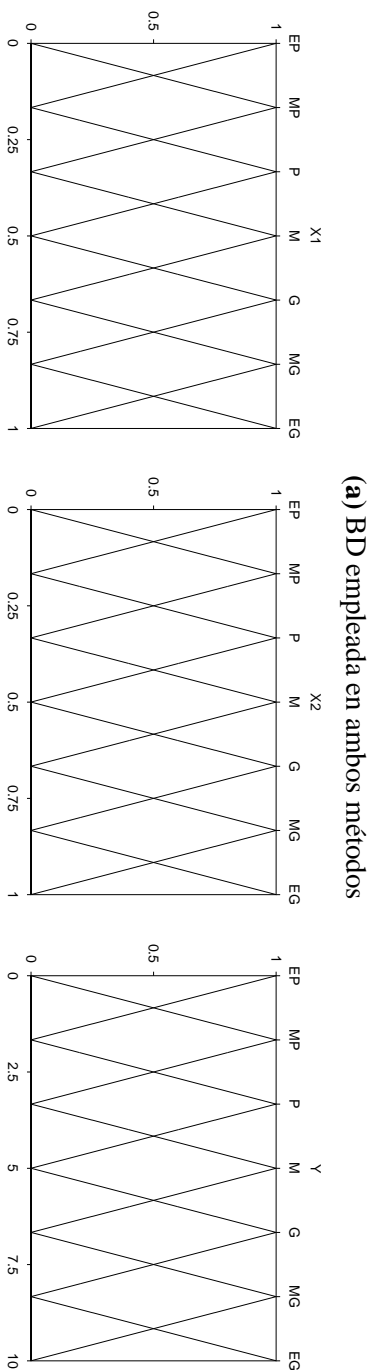
Hay dos métodos, el WM en ML y el WCA en MD, que no generan modelos apropiados en términos de precisión. En el primer caso, estos resultados quizás estén relacionados con el hecho de que las reglas lingüísticas generadas hayan sido definidas considerando criterios locales, sin tener en cuenta el comportamiento global del sistema. Profundizaremos en este aspecto en el capítulo próximo. En el segundo caso, el mal comportamiento desarrollado por el método muestra la dificultad lógica de diseñar SBRDs apropiados con algoritmos específicos de envoltura de ejemplos.

- *Interpretabilidad:* En cuanto a la legibilidad de los modelos generados, los obtenidos por los métodos de ML son significativamente más interpretables que los generados en el MD, gracias al uso de una semántica global. No obstante, merecen una mención especial los modelos difusos con sólo cuatro y cinco reglas (en media para los diez modelos) obtenidos por los métodos M-AARR y M-AARE en el problema del arroz —la figura 1.14(b) muestra una BRD con sólo dos reglas generada por el método M-AARR en este problema—.
- *Recomendaciones según el problema:*
 - *Problemas con una complejidad baja:* En problemas sencillos como F_2 , el ML es claramente mejor que el MD ya que se obtienen modelos más precisos y, al mismo tiempo, más interpretables. Podemos comprobar cómo el modelo obtenido por el método P-FCS1 (figura 1.11), además de ser menos preciso, no puede interpretarse en absoluto. Por contra,

- el método T genera un modelo (figuras 1.9(a)(b)) con buena precisión y fácilmente comprensible.
- *Problemas con una complejidad media:* A igualdad de resultados de precisión entre el ML y el MD en el problema del arroz, el ML desarrollado con los SBRDs lingüísticos debería ser preferible al MD con SBRDs aproximativos, ya que el primero posee, además, una alta interpretabilidad. Por tanto, los modelos obtenidos por el método de aprendizaje T son recomendables en este problema frente a modelos como los generados por los métodos M-AARR y M-AARE en los que, a pesar de sus buenos resultados, resulta muy difícil entender las acciones de las reglas difusas (en la figura 1.14(b) se muestra un modelo generado por el método M-AARR).
 - *Problemas con una complejidad media-alta:* En el problema de distribución eléctrica, la preferencia entre uno u otro enfoque dependerá del principal requisito que se desee satisfacer: interpretabilidad o precisión. Se obtienen modelos más precisos con los métodos de MD a expensas de perder la capacidad de interpretar su comportamiento en cierta medida. En las figuras 1.15(a) y 1.16(b) podemos ver dos modelos obtenidos por sendos enfoques. En efecto, aunque el modelo generado por el método M-AARR (figura 1.16(b)) es más preciso que el obtenido por el método T (figura 1.15(a)), la dificultad para entender el primero es significativamente superior.
 - *Problemas con una complejidad alta:* El MD es claramente recomendable en un problema complejo como F_3 , ya que obtiene modelos significativamente más precisos.

1.5.3.2. Modelado Lingüístico frente a Modelado Lingüístico Extendido

- *Precisión:* Podemos observar que los métodos de ML extendido obtienen, en general, modelos lingüísticos más precisos. No obstante, algunos modelos lingüísticos extendidos presentan cierto sobreaprendizaje del problema perdiendo generalización, como el caso de los modelos generados por los métodos LM y T+M-L-Ajus en el problema del arroz, o los métodos ALM, NIT y WM-RP en el problema eléctrico. Este empeoramiento de la predicción es uno de los



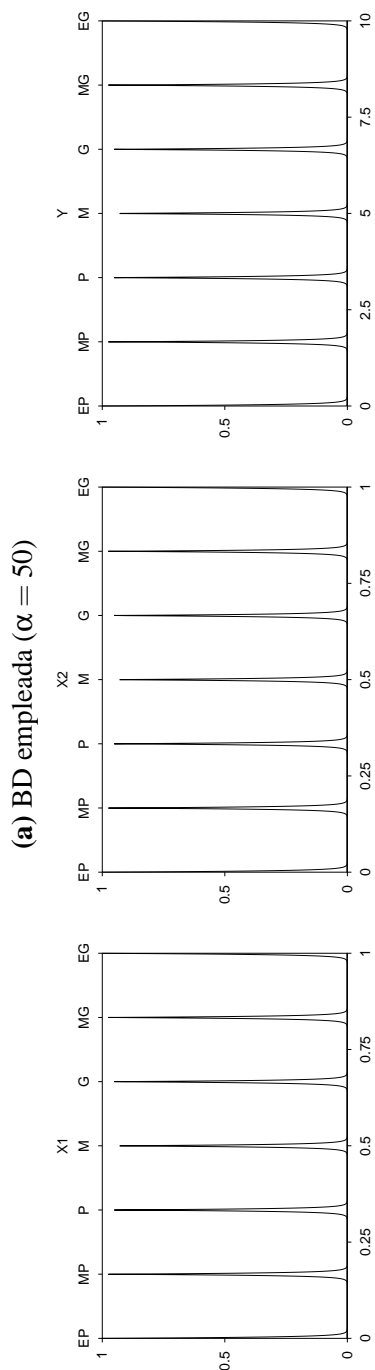
(b) BR generada por el método T

	X1						X2				
EP	M	EP	M	EP	M	EP	M	EP	M	EP	M
MP	EP	M	MP	M	EP	M	MP	M	EP	M	MP
P	EP	P	P	M	G	EP	P	P	M	G	EP
M	EP	MP	MP	P	M	G	EP	MP	MP	P	M
G	EP	MP	MP	P	M	G	EP	MP	MP	P	M
MG	EP	EP	EP	EP	EP	EP	EP	EP	EP	EP	EP
EG	EP	EP	EP	EP	EP	EP	EP	EP	EP	EP	EP

(c) BR generada por el método WM-RP

	X1						X2				
EP	EP	EP	EP	EP	EP	EP	EP	EP	EP	EP	EP
MP	EP	EP	EP	EP	EP	EP	EP	EP	EP	EP	EP
P	EP	EP	EP	EP	EP	EP	EP	EP	EP	EP	EP
M	EP	EP	EP	EP	EP	EP	EP	EP	EP	EP	EP
G	EP	EP	EP	EP	EP	EP	EP	EP	EP	EP	EP
MG	EP	EP	EP	EP	EP	EP	EP	EP	EP	EP	EP
EG	EP	EP	EP	EP	EP	EP	EP	EP	EP	EP	EP

Figura 1.9.: Modelos lingüísticos obtenidos por los métodos T y WM-RP para el problema F₂



(b) BRs primaria (izquierda) y secundaria (derecha)

		X1							X2							X1							X2						
		EP	MP	P	M	G	MG	EG	EP	MP	P	M	G	MG	EG	EP	MP	P	M	G	MG	EG	EP	MP	P	M	G	MG	EG
EP	M	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
MP	EP	1.0000	1.0000	0.7266	0.9618	0.5055	0.7899	1.0000	1.0000	1.0000	0.7266	0.9618	0.5055	0.7899	1.0000	1.0000	1.0000	0.7266	0.9618	0.5055	0.7899	1.0000	1.0000	1.0000	0.7266	0.9618	0.5055	0.7899	1.0000
P	EP	1.0000	0.7266	1.0000	0.9240	0.9087	0.5055	1.0000	1.0000	0.7266	1.0000	0.9240	0.9087	0.5055	1.0000	1.0000	0.7266	1.0000	0.9240	0.9087	0.5055	1.0000	1.0000	0.7266	1.0000	0.9240	0.9087	0.5055	1.0000
M	EP	1.0000	0.9618	0.9240	1.0000	0.9240	0.9618	1.0000	1.0000	0.9618	0.9240	1.0000	0.9240	0.9618	1.0000	1.0000	0.9618	0.9240	1.0000	0.9240	0.9618	1.0000	1.0000	0.9618	0.9240	1.0000	0.9240	0.9618	1.0000
G	EP	1.0000	0.5055	0.9087	0.9240	1.0000	0.7266	1.0000	1.0000	0.5055	0.9087	0.9240	1.0000	0.7266	1.0000	1.0000	0.5055	0.9087	0.9240	1.0000	0.7266	1.0000	1.0000	0.5055	0.9087	0.9240	1.0000	0.7266	1.0000
MG	EP	1.0000	0.7899	0.5055	0.9618	0.7266	1.0000	1.0000	1.0000	0.7899	0.5055	0.9618	0.7266	1.0000	1.0000	1.0000	0.7899	0.5055	0.9618	0.7266	1.0000	1.0000	1.0000	0.7899	0.5055	0.9618	0.7266	1.0000	1.0000
EG	EP	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

Figura 1.10.: Modelo lingüístico obtenido por el método NIT para el problema F₂

Regla	X ₁	X ₂	Y	Regla	X ₁	X ₂	Y	Regla	X ₁	X ₂	Y	Regla	X ₁	X ₂	Y
R ₁				R ₁₂				R ₂₃				R ₃₄			
R ₂				R ₁₃				R ₂₄				R ₃₅			
R ₃				R ₁₄				R ₂₅				R ₃₆			
R ₄				R ₁₅				R ₂₆				R ₃₇			
R ₅				R ₁₆				R ₂₇				R ₃₈			
R ₆				R ₁₇				R ₂₈				R ₃₉			
R ₇				R ₁₈				R ₂₉				R ₄₀			
R ₈				R ₁₉				R ₃₀				R ₄₁			
R ₉				R ₂₀				R ₃₁				R ₄₂			
R ₁₀				R ₂₁				R ₃₂							
R ₁₁				R ₂₂				R ₃₃							

Figura 1.11.: Modelo difuso generado por el método P-FCSI en el problema F₂

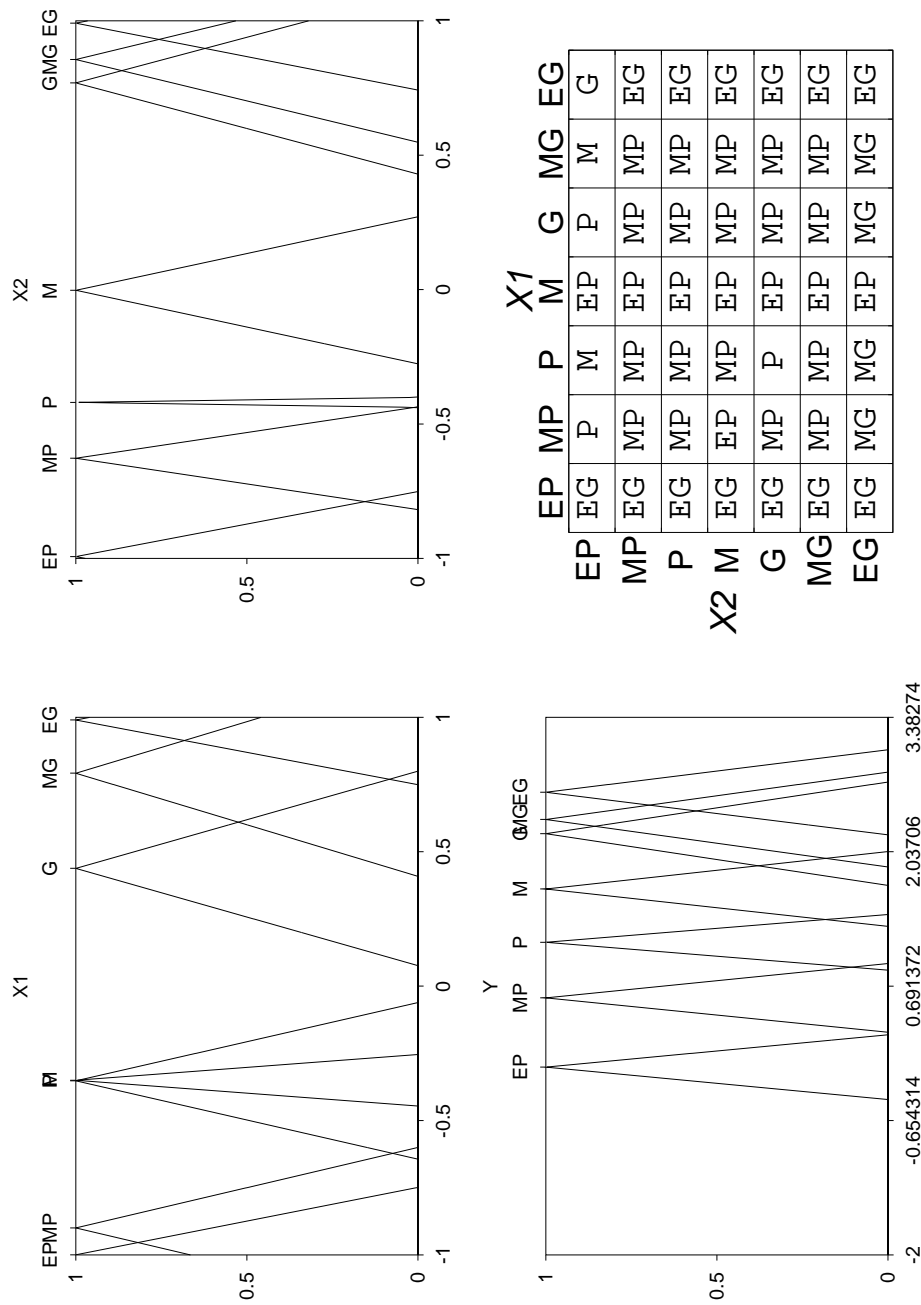


Figura 1.12.: Modelo difuso generado por el método LM para el problema F₃

Regla	X ₁	X ₂	Y	Regla	X ₁	X ₂	Y	Regla	X ₁	X ₂	Y	Regla	X ₁	X ₂	Y
R ₁	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₁₄	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₂₇	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₄₀	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
R ₂	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₁₅	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₂₈	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₄₁	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
R ₃	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₁₆	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₂₉	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₄₂	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
R ₄	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₁₇	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₃₀	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₄₃	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
R ₅	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₁₈	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₃₁	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₄₄	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
R ₆	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₁₉	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₃₂	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₄₅	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
R ₇	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₂₀	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₃₃	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₄₆	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
R ₈	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₂₁	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₃₄	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₄₇	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
R ₉	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₂₂	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₃₅	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₄₈	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
R ₁₀	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₂₃	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₃₆	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₄₉	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
R ₁₁	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₂₄	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₃₇	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>				
R ₁₂	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₂₅	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₃₈	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>				
R ₁₃	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₂₆	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	R ₃₉	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>				

Figura 1.13.: Modelo difuso generado por el método WCA para el problema F₃

1.5. Estudio Experimental del ML Extendido frente al MD

(a) BR del método ALM ($ECM_{entr/prue} = 0,003838 / 0,002852$)

Regla	Sabor	Aspecto	Gusto	Pegajos.	Dureza	Eval.
R_1	Malo	Malo	Malo	Suelto	Duro	Baja
R_2	Malo	Bueno	Malo	Suelto	Blando	Baja
R_3	Bueno	Malo	Malo	Suelto	Blando	Baja
R_4	Malo	Bueno	Bueno	Suelto	Blando	Alta
R_5	Bueno	Bueno	Bueno	Pegajoso	Blando	Alta
R_6	Bueno	Bueno	Bueno	Pegajoso	Duro	Alta

(b) BRD de M-AARR ($ECM_{entr/prue} = 0,0025218 / 0,00329$)

Regla	S	A	G	P	D	E
R_1						
R_2						

(c) BR generada por el método WM —sin modificadores lingüísticos— ($ECM_{entr/prue} = 0,014704 / 0,016700$) y WM-LH —con modificadores lingüísticos— ($ECM_{entr/prue} = 0,002230 / 0,000901$)

Regla	Sabor	Aspecto	Gusto	Pegajosidad	Dureza	Evaluación
R_1	<i>mom</i> Malo	<i>mom</i> Malo	<i>muy</i> Malo	Suelto	<i>mom</i> Blando	Baja
R_2	<i>mom</i> Malo	<i>muy</i> Malo	<i>muy</i> Malo	<i>muy</i> Suelto	<i>muy</i> Duro	Baja
R_3	<i>mom</i> Malo	<i>mom</i> Bueno	<i>muy</i> Malo	Suelto	<i>muy</i> Blando	<i>muy</i> Baja
R_4	<i>mom</i> Malo	Bueno	<i>muy</i> Bueno	<i>muy</i> Suelto	<i>mom</i> Blando	<i>muy</i> Baja
R_5	<i>muy</i> Malo	Bueno	Bueno	<i>muy</i> Pegajoso	Blando	Baja
R_6	<i>mom</i> Bueno	<i>muy</i> Malo	<i>muy</i> Malo	<i>muy</i> Suelto	<i>mom</i> Blando	<i>mom</i> Baja
R_7	<i>mom</i> Bueno	Malo	<i>muy</i> Malo	<i>mom</i> Suelto	<i>mom</i> Duro	Baja
R_8	<i>mom</i> Bueno	Malo	<i>muy</i> Bueno	<i>muy</i> Suelto	<i>mom</i> Blando	<i>mom</i> Baja
R_9	<i>muy</i> Bueno	Bueno	<i>muy</i> Malo	<i>muy</i> Suelto	<i>muy</i> Duro	Baja
R_{10}	<i>muy</i> Bueno	<i>muy</i> Bueno	<i>muy</i> Malo	<i>muy</i> Suelto	<i>mom</i> Blando	<i>mom</i> Alta
R_{11}	<i>muy</i> Bueno	Bueno	<i>muy</i> Malo	<i>muy</i> Pegajoso	Blando	Alta
R_{12}	Bueno	Bueno	<i>muy</i> Bueno	<i>mom</i> Suelto	<i>muy</i> Blando	Alta
R_{13}	<i>mom</i> Bueno	<i>mom</i> Bueno	<i>mom</i> Bueno	<i>mom</i> Suelto	<i>mom</i> Duro	<i>muy</i> Alta
R_{14}	<i>muy</i> Bueno	<i>muy</i> Bueno	Bueno	Pegajoso	<i>mom</i> Blando	<i>mom</i> Alta
R_{15}	<i>muy</i> Bueno	Bueno	<i>muy</i> Bueno	Pegajoso	Duro	<i>mom</i> Alta

Figura 1.14.: Algunos modelos generados para una partición concreta del conjunto de datos en el problema de la evaluación de la calidad del arroz

(a) BR de los métodos T ($ECM_{entr/prue} = 169.077 / 175.739$) y T+M-L-Ajus

		X_1						
		EP	MP	P	M	G	MG	EG
X_2	EP	EP	MP	EP	EG	EP	EG	P
	MP	EP	MP	MP	MP	EG	MG	EP
	P	MP	P	P	M	P	MG	EG
	M	MP	M	M	EG	G	MP	P
	G	M	P	G	EG	G	EP	EG
	MG	MP	G	M	EP	P	M	MG
	EG	P	MG	EP	G	MG	MP	MP

(b) BD del método T+M-L-Ajus ($ECM_{entr/prue} = 147.844 / 164.993$)

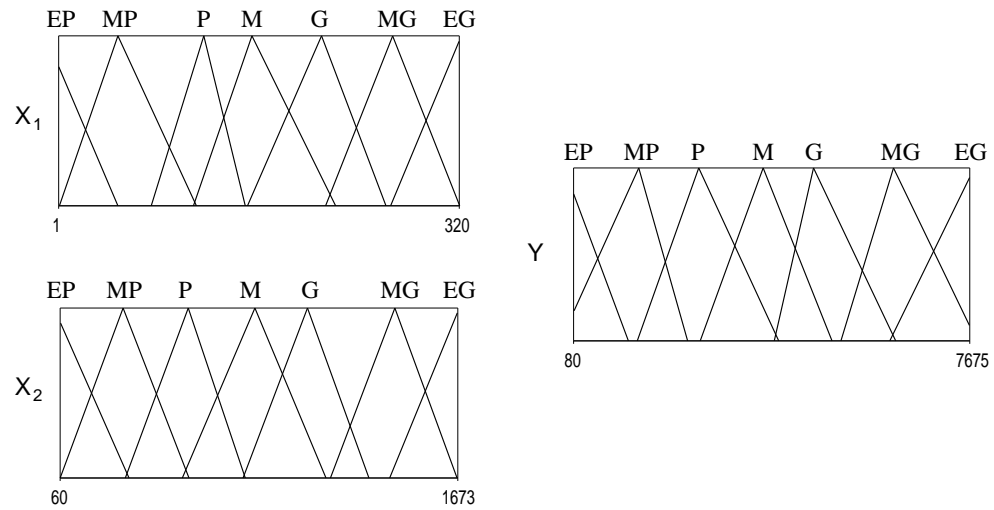


Figura 1.15.: Modelos generados por los métodos T y T+M-L-Ajus en el problema de las líneas eléctricas de baja tensión

(a) BR del método ALM ($ECM_{entr/prue} = 155.866 / 178.601$)

		X_1						
		ES	VS	S	M	L	VL	EL
X_2	ES	ES	ES					
	VS	ES VS	VS	VS				
	S	VS	VS M		S	M		
	M	VS	S L		VL			S
	L							
	VL	VS	L		M			
	EL	M						

(b) BRD del método M-AARR ($ECM_{entr/prue} = 142.109 / 166.579$)

<i>Regla</i>	X_1	X_2	Y	<i>Regla</i>	X_1	X_2	Y
R_1				R_{11}			
R_2				R_{12}			
R_3				R_{13}			
R_4				R_{14}			
R_5				R_{15}			
R_6				R_{16}			
R_7				R_{17}			
R_8				R_{18}			
R_9				R_{19}			
R_{10}				R_{20}			

Figura 1.16.: Modelos generados por los métodos ALM y M-AARR en el problema de las líneas eléctricas de baja tensión

riesgos que supone emplear una excesiva flexibilidad del modelo y demuestra la dificultad de superar al ML en algunos casos.

- *Interpretabilidad:* Las extensiones al ML realizadas por algunos métodos, a pesar de obtener buenos grados de precisión, conllevan el uso de un alto número de reglas en algunos casos (por ejemplo, el método LM en el problema del arroz, el método M-L en F_2 y F_3 , o el método NIT en los cuatro problemas). Además de la correspondiente pérdida de interpretabilidad derivada de tales extensiones (ajuste de las funciones de pertenencia, empleo de varios consecuentes por regla, reglas ponderadas, etc.), la existencia de muchas reglas hace que el modelo obtenido sea aún más difícil de entender. Frente a estos casos, algunos métodos logran mejorar significativamente el rendimiento manteniendo una buena interpretabilidad, como mencionamos en la sección 1.4.3.

Por ejemplo, las figuras 1.15(a) y 1.15(b) muestran respectivamente la BR generada por el método T y la correspondiente BD alterada por el método de ajuste M-L-Ajus en el problema eléctrico. Como podemos observar en este segundo método, usando la misma BR y manteniendo prácticamente la BD original (generada uniformemente) se puede obtener un modelo notablemente más preciso con una interpretabilidad similar.

- *Recomendaciones según el problema:* A continuación indicaremos qué métodos obtienen los mejores resultados en cada problema atendiendo a la precisión (con buen equilibrio entre aproximación y generalización) e interpretabilidad de los modelos generados.
 - *Problemas con una complejidad baja:* El método T+M-L-Ajus obtiene un modelo muy bueno en F_2 . Aunque los métodos M-L, NIT y WM-RP también obtienen buenos resultados de precisión, sus modelos son difíciles de interpretar debido al excesivo número de reglas (algunas de ellas con varios consecuentes) en los dos primeros casos y al uso de reglas con pesos en los dos últimos casos (figuras 1.9(c) y 1.10(b)).
 - *Problemas con una complejidad media:* En el problema del arroz, el método WM-LH (con buena precisión) y el método ALM (aunque su modelo no es tan preciso, tiene una excelente interpretabilidad ya que mantiene la semántica original y emplea sólo seis reglas) pueden considerarse como

los que mejores soluciones obtienen. Los modelos generados por estos métodos se representan en las figuras 1.14(a) y 1.14(c), respectivamente.

- *Problemas con una complejidad media-alta*: El método T+M-L-Ajus genera un modelo con un buen grado de aproximación y predicción en el problema eléctrico, teniendo además una buena interpretabilidad.
- *Problemas con una complejidad alta*: El método LM es claramente la mejor opción en el problema F₃. No obstante, en la figura 1.12 queda patente la pérdida de interpretabilidad que supone alcanzar ese grado de precisión. Aunque el método M-L también obtiene un modelo preciso, su interpretabilidad es muy mala debido al gran número de reglas empleado.

En general, debemos destacar que se obtienen modelos muy buenos combinando métodos simples con un proceso de refinamiento posterior, bien ajustando las funciones de pertenencia (WM+M-L-Ajus) o bien añadiendo modificadores lingüísticos (WM-LH). Esta combinación presenta resultados similares a los de mecanismos complejos de generación de reglas como los métodos LM y M-L, que también realizan un proceso de ajuste *a posteriori*, siendo además significativamente más simples y rápidos.

1.5.3.3. Modelado Lingüístico Extendido frente a Modelado Difuso

- *Precisión*: Los modelos obtenidos en el ML extendido superan significativamente al MD en los problemas F₂ y arroz. En el problema eléctrico, se obtienen resultados de precisión similares o ligeramente mejores con los métodos que realizan un ML extendido. Sólo en un problema tan complejo como F₃, los métodos de MD generan modelos claramente más precisos.
- *Interpretabilidad*: En cuanto a la facilidad para entender el comportamiento del modelo obtenido, a pesar de la ligera pérdida de legibilidad experimentada por el ML extendido, estos modelos son con diferencia más interpretables que los modelos aproximativos gracias a la consideración de una semántica global. Por ejemplo, podemos comprobar cómo el modelo obtenido en el problema eléctrico por el método T+M-L-Ajus (cuyas BR y BD se muestran en la figura 1.15) es claramente más interpretable que el modelo difuso obtenido por el método M-AARR (figura 1.16(b)).

- *Recomendaciones según el problema:*
 - *Problemas con una complejidad baja:* Los métodos de aprendizaje del ML extendido son claramente mejores tanto en precisión como en interpretabilidad en el problema F₂.
 - *Problemas con una complejidad media:* En el problema de la valoración de la calidad del arroz, el ML extendido demuestra ser mejor que el MD ya que el primero obtiene los modelos con los mejores grados de predicción, siendo además más fácilmente interpretables. No obstante, el modelo obtenido por el método de MD M-AARR con sólo cuatro reglas (en media para las diez particiones consideradas en este problema) es un resultado interesante, aunque su precisión es peor que, por ejemplo, la del modelo obtenido por el método de ML extendido ALM que sólo presenta cinco reglas. En las figuras 1.14(a) y 1.14(b) se muestran dos modelos obtenidos por estos métodos para una partición de datos concreta. Como podemos observar, el método de ML extendido obtiene una solución ligeramente más precisa pero notablemente más interpretable.
 - *Problemas con una complejidad media-alta:* Teniendo en cuenta conjuntamente la precisión y la interpretabilidad de los modelos generados en el problema eléctrico, los métodos de ML extendido son preferibles a los del MD. En la figura 1.15(b) se puede observar que el ajuste realizado a la semántica global sobre la BC generada por el método T (figura 1.15(a)) hace que el modelo obtenido sea más preciso e interpretable que el mejor modelo difuso analizado (obtenido por el método M-AARR, figura 1.16(b)).
 - *Problemas con una complejidad alta:* En el problema F₃, decidirse por una u otra opción dependerá del principal requisito que estemos exigiendo: mientras que en el ML extendido se obtiene una precisión satisfactoria con buena interpretabilidad, la excelente precisión del MD se consigue sacrificando en gran medida la interpretabilidad.

1.6. Sumario

El objetivo de este capítulo ha sido introducir el Modelado de Sistemas con SBRDs y presentar las dos vertientes principales existentes en esta área: el ML y

el MD. Se ha realizado también un análisis comparativo entre ambos enfoques más allá del clásico punto de vista resignado a considerar simplemente al ML como una herramienta con buena interpretabilidad pero mala precisión.

Así, se han introducido algunas posibilidades para mejorar la precisión de los modelos lingüísticos manteniendo una buena legibilidad bajo la suposición de que se podría obtener mejor precisión buscando un buen balance entre flexibilidad del modelo y simplicidad del modelado. Se ha demostrado el buen comportamiento de tales extensiones frente al MD mediante un estudio experimental donde se han aplicado trece métodos de derivación (cuidadosamente seleccionados para abarcar el espectro de posibilidades analizado) a cuatro problemas de modelado de diferente naturaleza.

Las conclusiones vertidas a lo largo de este capítulo se pueden resumir contestando a las siguientes preguntas:

¿Qué posibilidades existen en el Modelado de Sistemas con SBRDs?

En el Modelado de Sistemas con SBRDs, normalmente encontramos dos requisitos contradictorios, la interpretabilidad y la precisión del modelo obtenido. Como es sabido, el ML (cuyo principal requisito es la interpretabilidad) se desarrolla con los SBRDs lingüísticos, mientras que el MD (cuyo principal requisito es la precisión) se desarrolla, entre otros, con los SBRDs aproximativos. Mientras que hacer a los SBRDs lingüísticos altamente interpretables implica establecer duras restricciones a la estructura de la regla (debido al uso de una semántica global), perdiendo así flexibilidad, relajar tales restricciones, como hacen los SBRDs aproximativos (usando una semántica local), puede dar lugar a que se obtengan modelos más flexibles pero a costa de que se pierda su interpretabilidad.

¿Es equivalente flexibilidad a precisión?

Esta pregunta se puede contestar formulando otra: ¿De qué sirve un modelo con gran flexibilidad si el proceso de derivación tiene serias dificultades para aprovecharla? En efecto, mayor flexibilidad no implica mayor precisión ya que la complejidad para realizar el proceso de modelado crece. Se podría obtener un mejor comportamiento manteniendo un buen equilibrio entre la libertad del modelo (flexibilidad del modelo) y la dificultad que supone en el proceso de derivación (simplicidad del modelado).

¿Cómo podemos obtener modelos más precisos con buena interpretabilidad?

Para alcanzar este equilibrio entre flexibilidad del modelo y simplicidad del modelado, o bien se extiende el ML para hacer más flexible la derivación y/o la estructura, o bien se restringe el proceso de derivación y/o la estructura del MD.

Si dicho equilibrio se alcanza partiendo del ML, los modelos obtenidos serán más interpretables que partiendo del MD. Por tanto, parece muy interesante afrontar el modelado de sistemas intentado mejorar, tanto como sea posible, la precisión del ML sin perder su descripción en exceso. Para hacer esto, hemos introducido dos tipos diferentes de extensiones:

- Mejoras en el *proceso de derivación*, definiendo las funciones de pertenencia de diferentes formas.
- Extensión en la *estructura del modelo*, empleando reglas con modificadores lingüísticos, reglas con varios consecuentes o reglas ponderadas.

Se han obtenido resultados muy interesantes en la experimentación de este capítulo combinando métodos de aprendizaje lingüísticos sencillos, que no emplean mecanismos de generación de reglas complejos, con un proceso de refinamiento *a posteriori* (o bien ajustando las funciones de pertenencia o bien añadiendo modificadores lingüísticos). Por tanto, parece una buena idea profundizar en estos dos enfoques para realizar procesos de aprendizaje rápidos y aceptables por un lado, y métodos de ajuste de los modelos obtenidos por otro.

¿Se puede realizar un ML tan preciso como el MD sin perder excesiva descripción?

En este capítulo hemos visto que, a pesar de que el ML es generalmente menos preciso que el MD, se puede extender adecuadamente para hacerlo más flexible. Con el ML extendido, el campo de actuación del ML se extiende cubriendo problemas con una complejidad media o media-alta, donde se alcanzan cotas similares o mejores de precisión que el MD pero con modelos más interpretables. Por tanto, esto nos conduce a pensar que, si no siempre, en la mayoría de los casos el ML extendido puede ser tan preciso como el MD sin perder su legibilidad en exceso.

1.7. Nuevas Líneas de Trabajo Propuestas en esta Memoria

Después de haber analizado varias de las extensiones al ML existentes y su comportamiento en problemas de distinta naturaleza, vamos a finalizar este capítulo proponiendo varias líneas de trabajo que pretenden agrupar muchas de las extensiones analizadas y que serán desarrolladas con profundidad en el resto de capítulos que componen la presente memoria:

- *Mejora en la generación de reglas lingüísticas* — Propondremos una nueva metodología de generación de reglas con buena cooperación que permita obtener rápidamente modelos con buena precisión. Naturalmente, los modelos obtenidos gozarán de la máxima interpretabilidad.
- *Extensiones ajustando funciones de pertenencia y usando una estructura más flexible con modificadores lingüísticos* — Propondremos un método que ajustará BCs previamente diseñadas efectuando modificaciones en dos niveles de descripción diferentes (con efectos macroscópicos y microscópicos). En este proceso se perseguirá mantener un buen grado de interpretabilidad.
- *Diferentes formas de combinar las mejoras anteriores* — Por último, propondremos varios métodos de derivación del modelo global que agrupan las propuestas anteriores de diferente forma. Distinguiremos dos mecanismos de hibridación: derivación secuencial o simultánea y empleo de Algoritmos Genéticos o Coevolución Cooperativa.

2. Metodología COR: Cooperación entre Reglas

Se necesitan dos para inventar cualquier cosa. Uno realiza las combinaciones; el otro elige, reconoce lo que desea y lo que es importante para él en la masa de cosas que el primero le ha ofrecido.

PAUL VALÉRY (1871–1945), POETA Y ENSAYISTA FRANCÉS

Como hemos visto en el capítulo anterior, los modelos lingüísticos se obtienen generalmente mediante un proceso de aprendizaje automático a partir de un conjunto de pares de datos entrada-salida 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, tales como las redes neuronales [Ful00, NKK97] o los AGs [CHHM01, HV96, Ped97].

Sin embargo, los complejos mecanismos de generación de reglas empleados por estos métodos pueden provocar que sean difíciles de diseñar y que el proceso de aprendizaje resulte tremendamente lento. Frente a estos, se ha propuesto una familia de métodos de aprendizaje eficientes y simples que no siguen ninguna técnica concreta y que se guían por criterios de envoltura¹ de los datos del conjunto de ejemplos para realizar la extracción de reglas, a los que llamaremos «*métodos específicos de envoltura de ejemplos*» [CH00, INT⁺94, NIT97, WM92].

Estos métodos disfrutan de algunas ventajas interesantes que los hacen muy adecuados para el aprendizaje:

¹ Traducido del término inglés *covering*. En este contexto se refiere a la acción de abarcar los distintos vectores de datos con reglas lingüísticas difusas

- Los algoritmos son fáciles de entender e implementar.
- Llevan a cabo un proceso de aprendizaje muy rápido.
- Además, gracias a las dos ventajas anteriores, este tipo de métodos son ideales para integrarse en procedimientos de modelado más complejos, lo cual ha favorecido que un aprendizaje rápido y con resultados de precisión aceptables haya cobrado un renovado interés dentro del Modelado de Sistemas mediante SBRDs.

En efecto, podemos encontrar en la literatura reciente interesantes aportaciones de estos métodos ya sea para servir como una primera aproximación para entender las características del problema que se desea resolver [CDM99], para generar soluciones iniciales que son posteriormente refinadas por otros procesos [CH97], o para integrarse en procesos de meta-aprendizaje como criterio de bondad [CHV01], entre otros.

No obstante, a pesar de estas ventajas, los modelos obtenidos no son, en algunos casos, tan precisos como se desearía. Esto se debe a que el objetivo que generalmente persiguen estos métodos es el de obtener reglas lingüísticas con el mejor comportamiento individual. Esto puede provocar que, en algunas ocasiones, el SBRD contenga reglas que no cooperan adecuadamente cuando actúan de forma colectiva y que, debido al razonamiento interpolado que experimentan estos sistemas, los resultados no sean lo suficientemente precisos.

En este capítulo abordaremos dicho problema proponiendo una nueva metodología de aprendizaje que mejorará significativamente la cooperación de las reglas lingüísticas, incrementando así la eficacia de los modelos obtenidos: la *Metodología de Cooperación entre Reglas* (COR). Su funcionamiento consistirá en generar un conjunto de reglas candidatas para representar cada subespacio de entrada difuso y realizar a continuación una búsqueda combinatoria rápida en estos conjuntos para encontrar la combinación de reglas que obtiene el mejor comportamiento globalmente, aunque las reglas seleccionadas no sean las mejores individualmente en sus correspondientes subespacios.

Para introducir esta metodología, comenzaremos el capítulo con una introducción a los métodos específicos de envoltura de ejemplos presentando y analizando una taxonomía según la información usada para guiar el proceso de generación de reglas. A continuación, presentaremos la metodología propuesta y propondremos varios mé-

todos concretos basados en la misma que emplean distintas técnicas de búsqueda. Posteriormente, presentaremos un análisis comparativo aplicando los métodos propuestos a diferentes problemas. Como fruto de este análisis, estudiaremos algunas extensiones sobre la metodología y finalizaremos el capítulo con un nuevo análisis experimental de las extensiones consideradas.

La notación empleada a lo largo de este capítulo se irá introduciendo según vaya siendo necesario. Al final del mismo (tabla 2.9, página 111) puede consultarse un resumen de toda la notación considerada.

2.1. Métodos Específicos de Envoltura de Ejemplos: Características y Taxonomía

Se ha propuesto una gran cantidad de métodos para generar automáticamente reglas difusas a partir de datos numéricos. Frente a los mecanismos de generación de reglas complejos, ha aparecido en la literatura una familia de métodos eficientes y simples, guiados por criterios de envoltura de los datos del conjunto de ejemplos, específicamente diseñados para el aprendizaje de reglas lingüísticas [CH00, INT⁺94, NIT97, WM92].

En los siguientes subapartados introduciremos las principales características que definen a estos métodos y realizaremos un análisis profundo de su comportamiento clasificándolos según la información que utilizan para guiar el proceso de aprendizaje.

2.1.1. Características de los Métodos Específicos de Envoltura de Ejemplos

Dentro de los métodos de aprendizaje de reglas difusas, podemos clasificar bajo el nombre de *métodos específicos de envoltura de ejemplos* a aquellos que comparten los cuatro aspectos siguientes:

1. El proceso de aprendizaje se basa en un conjunto de datos E que representa el comportamiento del problema que se pretende resolver, siendo $E = \{e_1, \dots, e_l, \dots, e_N\}$ —donde $e_l = (x_1^l, \dots, x_n^l, y_1^l, \dots, y_m^l)$ es el l -ésimo ejemplo, N es el tamaño del conjunto de datos, n es el número de variables de entrada y m es el número de variables de salida—. En esta memoria trabajaremos con sistemas

MISO (del inglés *Multiple-Input Single-Output*) en los que se consideran varias variables de entrada pero sólo una de salida, es decir, $m = 1$, $e_l = (x_1^l, \dots, x_n^l, y^l)$.

2. Se hace uso de una definición previa de la BD compuesta por las particiones difusas de las variables de entrada y salida. Como comentamos en el capítulo anterior, dichas particiones pueden obtenerse directamente a partir de la información proporcionada por expertos (si está disponible) o mediante un proceso de normalización. En el segundo caso, lo más común es dividir cada universo de discurso asociado a cada variable en un determinado número de subespacios, elegir un tipo de función de pertenencia y asignar un conjunto difuso a cada uno de ellos. En nuestro caso vamos a considerar particiones difusas simétricas con funciones de pertenencia triangulares que se cruzan a una altura de 0,5 (en la figura 2.1 se muestra un ejemplo de este tipo de particiones con cinco etiquetas).

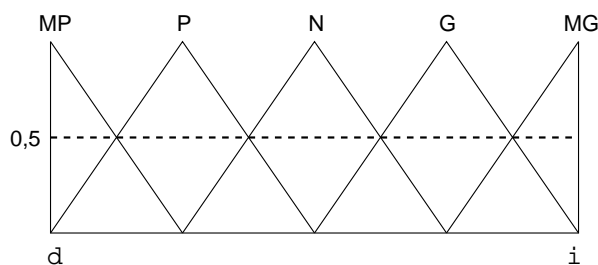


Figura 2.1.: Representación gráfica de una partición difusa uniforme, donde P significa ‘pequeño’, N ‘normal’, G ‘grande’, M ‘muy’ y $[d, i]$ es el dominio de la variable

El conjunto de términos lingüísticos de la i -ésima variable de entrada se notará por \mathcal{A}_i —donde $i \in \{1, \dots, n\}$ — y el conjunto de términos lingüísticos de la variable de salida por \mathcal{B} , siendo $|\mathcal{A}_i|$ el número de términos lingüísticos de la i -ésima variable de entrada y $|\mathcal{B}|$ el número de términos lingüísticos de la variable de salida.

3. El aprendizaje de las reglas lingüísticas hace uso de criterios de envoltura de los datos del conjunto de ejemplos.
4. La forma de realizar el proceso de aprendizaje no considera ninguna de las técnicas de búsqueda u optimización conocidas (tales como los AGs o las redes neuronales), sino que está específicamente desarrollada para este propósito.

2.1.2. Taxonomía de los Métodos Específicos de Envoltura de Ejemplos

Dentro de los métodos específicos de envoltura de ejemplos podemos distinguir dos enfoques distintos para obtener las reglas lingüísticas: los métodos guiados por *ejemplos* y los guiados por *rejilla difusa*. Mientras que en el primer caso cada regla lingüística candidata se obtiene a partir de un ejemplo concreto, en el segundo se agrupan los ejemplos en subespacios (definidos por la rejilla difusa) y se obtiene una regla para cada uno de ellos. En los siguientes subapartados, analizaremos ambos enfoques y obtendremos un esquema de funcionamiento genérico sirviéndonos de dos métodos concretos.

2.1.2.1. Métodos Específicos de Envoltura de Ejemplos Guiados por Ejemplos

En este enfoque se obtiene una regla lingüística para cada ejemplo del conjunto E , es decir, la regla RC^l se obtiene a partir del ejemplo e_l . Después, estas reglas, que pertenecen a un conjunto de reglas candidatas, sufren un proceso de selección con el que se obtiene la BR final. En la figura 2.2 se representa gráficamente este proceso de generación de reglas.

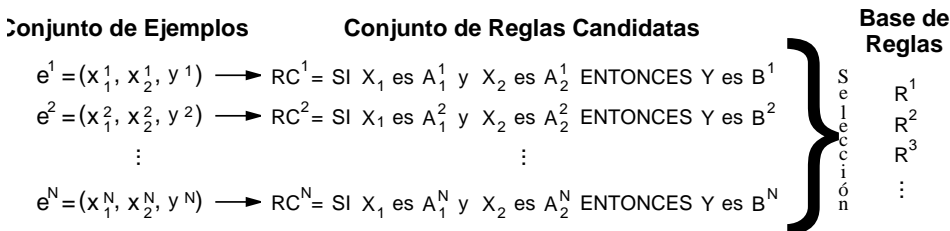


Figura 2.2.: Proceso de generación de reglas seguido en los métodos guiados por ejemplos

Uno de los métodos guiados por ejemplos más conocidos y usados es el propuesto por Wang y Mendel [WM92] (*método WM*). Dicho método lleva a cabo el aprendizaje por medio de los pasos indicados en la figura 2.3.

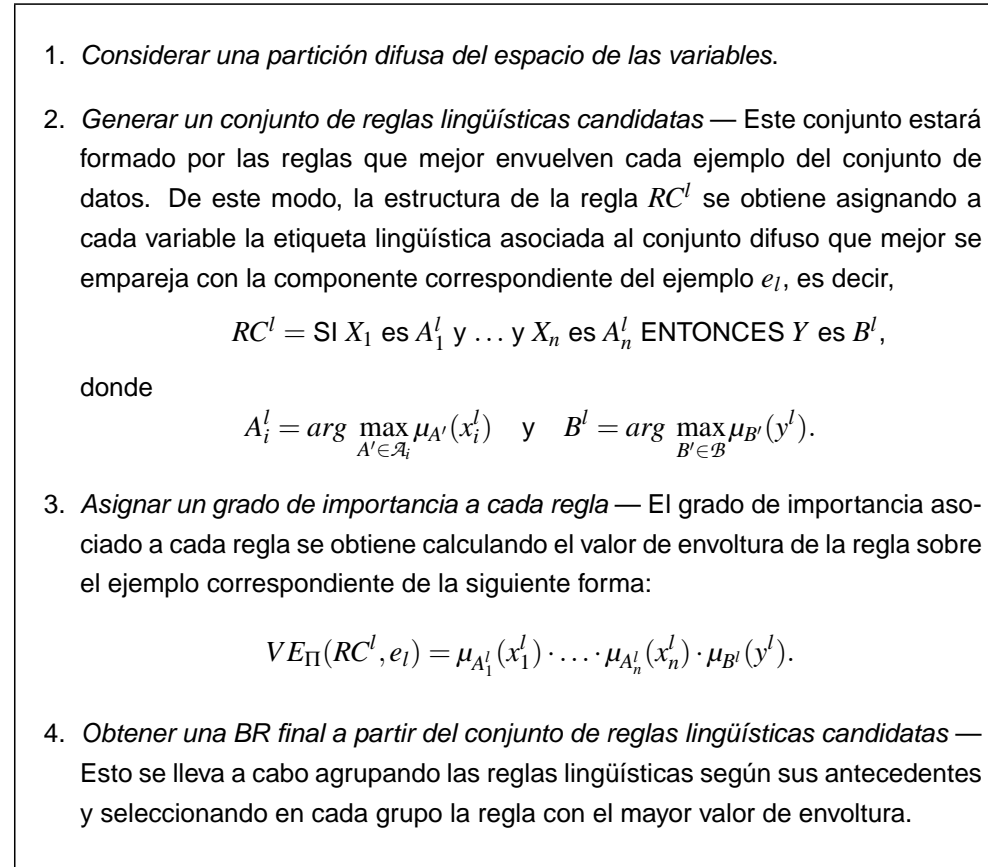


Figura 2.3.: Método WM

2.1.2.2. Métodos Específicos de Envoltura de Ejemplos Guiados por Rejilla Difusa

Otra posibilidad para generar las reglas lingüísticas es agrupar los ejemplos de acuerdo a una rejilla difusa y obtener después una regla para cada grupo (subespacio) considerando todos los ejemplos contenidos en él. La rejilla difusa se obtiene combinando las particiones difusas de las variables de entrada. De esta forma, el subespacio difuso multidimensional de entrada $S_s = (A_1^s, \dots, A_i^s, \dots, A_n^s)$ —donde $s \in \{1, \dots, N_S\}$, $N_S = \prod_{i=1}^n |\mathcal{A}_i|$ es el número de subespacios de entrada difusos multidimensionales y $A_i^s \in \mathcal{A}_i$ — vendrá definido por el producto cartesiano de los términos lingüísticos de las variables de entrada, es decir, $S_s \in \mathcal{A}_1 \times \dots \times \mathcal{A}_n$. La figura 2.4 ilustra gráficamente

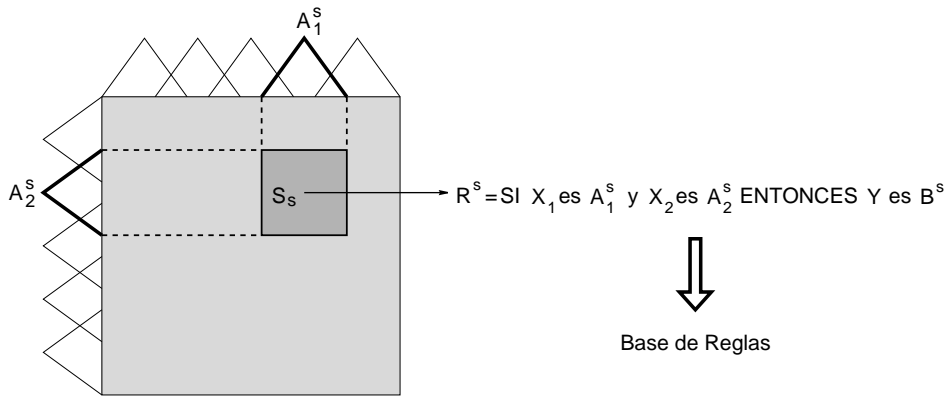


Figura 2.4.: Proceso de generación de reglas seguido en los métodos guiados por rejilla difusa

este proceso de generación de reglas según esta segunda filosofía.

En [CH00] se propone un método basado en este segundo enfoque. En él, los autores adaptan el método de aprendizaje de reglas del tipo TSK simplificado (es decir, reglas con un valor real en el consecuyente) propuesto por Ishibuchi *et al.* en [INT⁺94] para generar reglas lingüísticas de tipo Mamdani (*método CH*). El método se compone de los pasos descritos en la figura 2.5.

Existen muchas opciones basadas en criterios de envoltura para formular la función de valoración de la regla FVR . En esta memoria trabajaremos con las tres siguientes:

- *Grado de envoltura del ejemplo mejor envuelto* — El grado de envoltura de la regla $R_{k^s}^s$ sobre el ejemplo mejor envuelto se calcula como:

$$FVR_1(R_{k^s}^s) = \max_{e_{l^s} \in E_s^l} VE_T(R_{k^s}^s, e_{l^s}).$$

- *Grado medio de envoltura sobre el conjunto de ejemplos* — El grado medio de envoltura de la regla $R_{k^s}^s$ sobre todos los ejemplos contenidos en E_s^l se calcula como:

$$FVR_2(R_{k^s}^s) = \frac{\sum_{e_{l^s} \in E_s^l} VE_T(R_{k^s}^s, e_{l^s})}{|E_s^l|}.$$

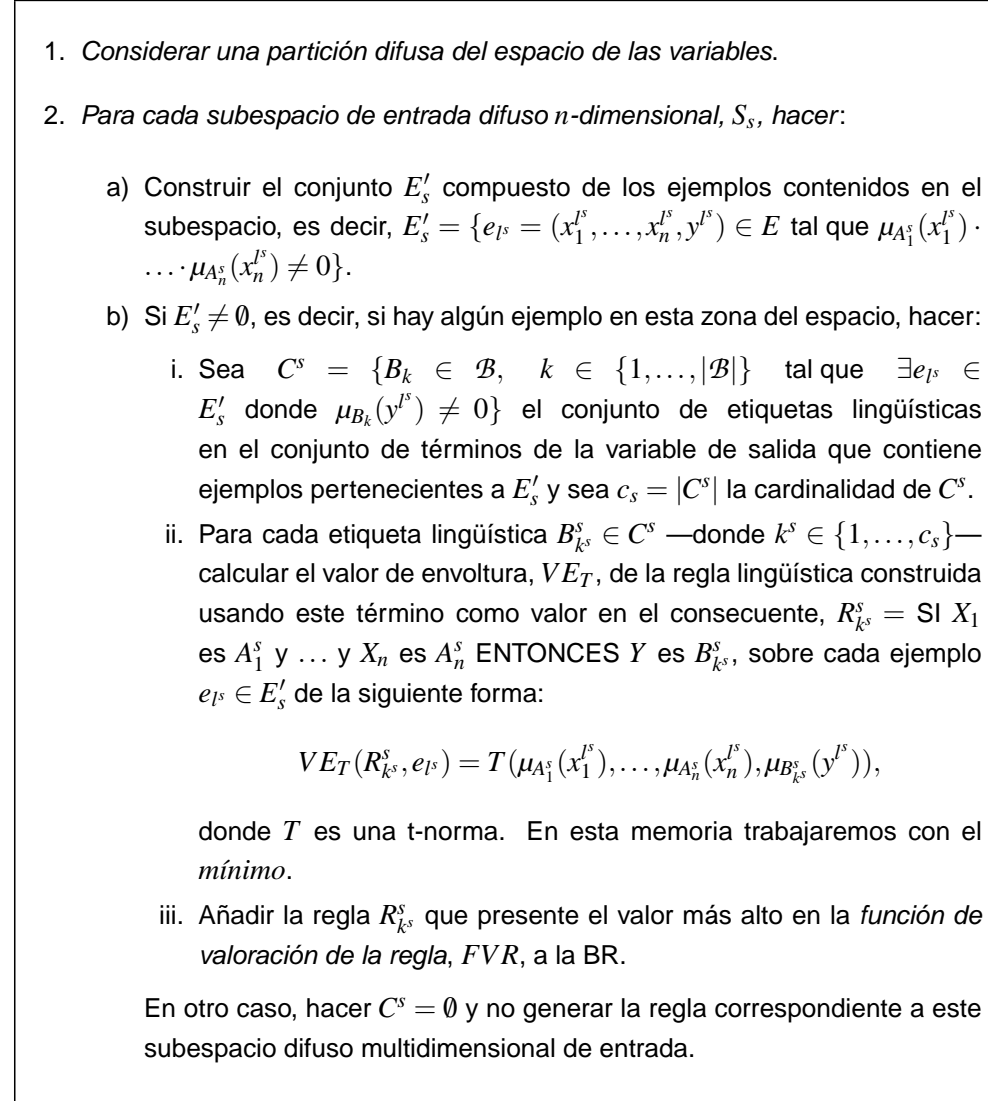


Figura 2.5.: Método CH

- *Promedio de los dos grados de envoltura anteriores* — Es la combinación de los grados de envoltura mejor y medio. Se calcula como:

$$FVR_3(R_{k^s}^s) = FVR_1(R_{k^s}^s) \cdot FVR_2(R_{k^s}^s) =$$

$$\max_{e_{l^s} \in E'_s} VE_T(R_{k^s}^s, e_{l^s}) \cdot \frac{\sum_{e_{l^s} \in E'_s} VE_T(R_{k^s}^s, e_{l^s})}{|E'_s|}.$$

2.1.2.3. Relación entre los Métodos Guiados por Ejemplos y los Guiados por Rejilla Difusa

Mientras que en un método guiado por ejemplos un ejemplo sólo participa en la generación de una regla lingüística, el proceso de obtención de reglas seguido por un método guiado por rejilla difusa implica que un ejemplo pueda contribuir a la generación de varias reglas. En efecto, podemos trazar un paralelismo entre ambos enfoques considerando el uso de diferentes tipos de rejilla (la figura 2.6 muestra gráficamente este hecho):

- *Métodos guiados por ejemplos* — Puesto que el antecedente de cada regla en un método guiado por ejemplos se obtiene con respecto al mejor emparejamiento obtenido para el ejemplo concreto (las etiquetas con el mayor grado de pertenencia), se puede considerar que un método guiado por ejemplos separa el conjunto de ejemplos atendiendo a una rejilla precisa (es decir, no difusa) delimitada por los puntos de cruce entre las etiquetas. Así, un ejemplo sólo puede pertenecer a un subespacio.
- *Métodos guiados por rejilla difusa* — Sin embargo, con la rejilla difusa usada en este segundo enfoque, un ejemplo puede pertenecer a más de un subespacio. En la figura 2.6(b), los ejemplos influyen en la generación de una, dos o cuatro reglas según caigan respectivamente en una zona blanca, gris claro o gris oscuro.

En resumen, hay varios subespacios que contienen ejemplos y un conjunto de reglas en cada uno de ellos. Dependiendo del enfoque seguido, un ejemplo sólo participará en un subespacio (generación a partir de ejemplos) o podrá participar en varios subespacios (generación a partir de rejilla difusa). Resulta difícil determinar qué enfoque es más apropiado. El enfoque guiado por ejemplos siempre obtiene

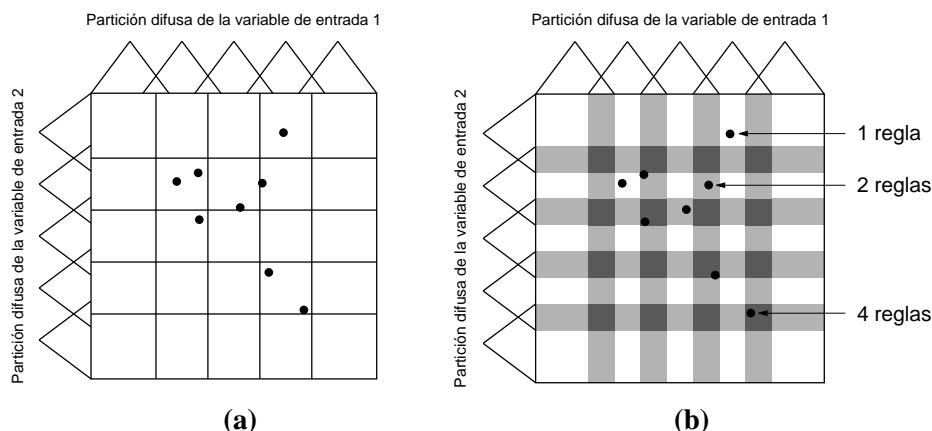


Figura 2.6.: (a) Métodos guiados por ejemplos (rejilla precisa) —un ejemplo sólo contribuye a generar una regla—, (b) Métodos guiados por rejilla difusa —un ejemplo puede contribuir a generar varias reglas—

igual o menor número de reglas que el guiado por rejilla difusa —efectivamente, al trabajar con regiones más reducidas, hay más probabilidad de que un subespacio no contenga ejemplos y, por consiguiente, no genere ninguna regla—, lo cual es muy importante en ML y puede ser determinante en ciertos casos con fuertes exigencias de interpretabilidad. Sin embargo, este hecho puede hacer que en otros casos el modelo obtenido por un método guiado por ejemplos no sea tan preciso como se desearía.

2.1.2.4. Esquema Genérico de los Métodos Específicos de Envoltura de Ejemplos

De acuerdo a las relaciones existentes entre los métodos guiados por ejemplos y los guiados por rejilla difusa señaladas en la sección anterior, podemos establecer un esquema genérico de funcionamiento de los métodos de aprendizaje de reglas lingüísticas específicos de envoltura de ejemplos. La figura 2.7 recoge los tres pasos básicos del proceso de aprendizaje seguido por estos métodos. Según este esquema, los métodos WM y CH se pueden interpretar como se muestra en la figura 2.8 (los detalles de cada proceso pueden encontrarse en las secciones anteriores).

Como queda reflejado en esta última figura, los métodos difieren en dos aspectos concretos, marcados como **1?** y **2?**, encuadrados respectivamente dentro del proceso de generación de reglas candidatas (paso 1) y de la fase de selección de reglas

Entradas:

- Considerar un conjunto de datos numéricos de entrada-salida que representen el comportamiento del problema que se desea resolver.
- Considerar una partición difusa de los espacios de las variables.

Algoritmo:

1. Generar reglas candidatas en cada subespacio de entrada difuso que presente ejemplos positivos (S_{s+}) siguiendo un enfoque de generación concreto (guiado por ejemplos o por rejilla difusa).

Después de este paso se dispondrá de un conjunto de consecuentes candidatos (etiquetas lingüísticas) en cada subespacio S_{s+} ,

$$C^{s+} = \{B_1^{s+}, \dots, B_{k^{s+}}^{s+}, \dots, B_{c_{s+}}^{s+}\},$$

así como una regla para cada uno de ellos,

$$R_{k^{s+}}^{s+} = \text{SI } X_1 \text{ es } A_1^{s+} \text{ y } \dots \text{ y } X_n \text{ es } A_n^{s+} \text{ ENTONCES } Y \text{ es } B_{k^{s+}}^{s+}.$$

2. Elegir la regla con el mejor consecuente en cada subespacio de entrada difuso atendiendo a algún criterio de envoltura.

Figura 2.7.: Esquema genérico de los métodos específicos de envoltura de ejemplos

(paso 2). El primero corresponde al enfoque de generación seguido por el método, guiado por ejemplos en el caso de (1a) y guiado por rejilla difusa en (1b). El segundo aspecto determina el criterio para elegir una regla concreta en un subespacio. Podemos comprobar que el criterio seguido por el método WM y el criterio FVR_1 del método CH son idénticos, ya que ambos atienden a la envoltura del ejemplo mejor envuelto.

Desde este nuevo punto de vista, resulta intuitivo pensar en posibles combinaciones entre estos aspectos para obtener nuevos métodos. De las seis combinaciones posibles, cuatro de ellas corresponden a los cuatro métodos hasta ahora presentados: el método WM, (1a2a); y las tres variantes del método CH, (1b2a), (1b2b) y (1b2c).

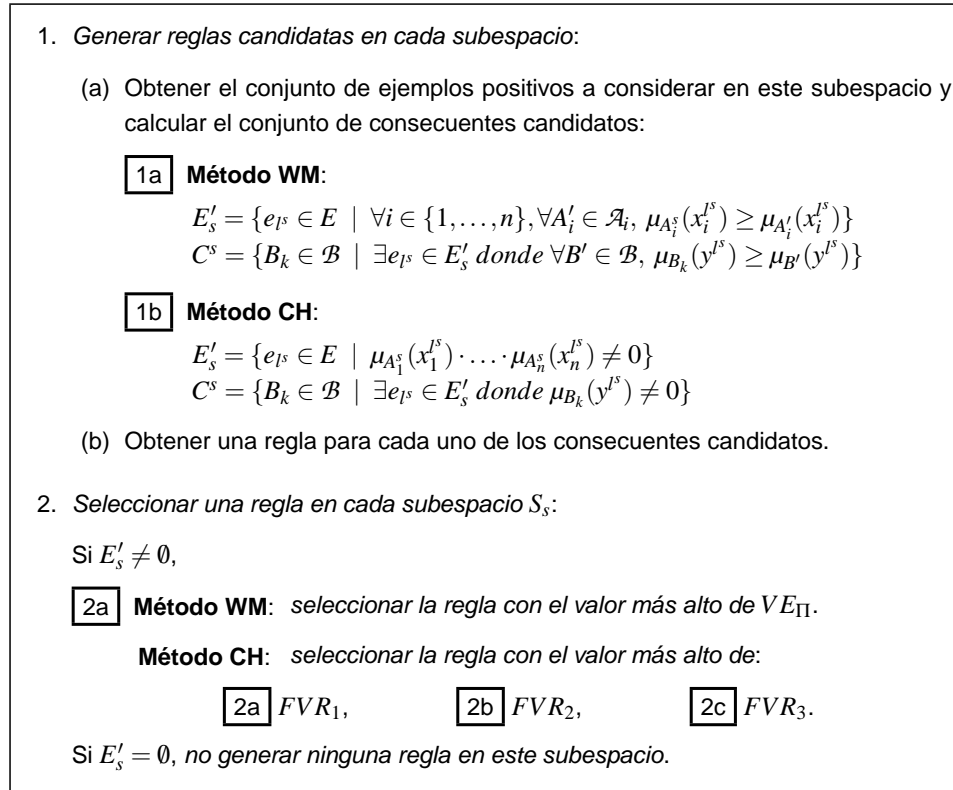


Figura 2.8.: Interpretación de los métodos WM y CH según el esquema genérico

Sin embargo, sería posible usar las funciones de valoración de reglas FVR_2 y FVR_3 , consideradas por el método CH, para dar un grado de importancia a cada regla en cada uno de los grupos obtenidos por el método WM, y seleccionar la regla con el mejor valor en cada grupo. De esta forma, surgen dos nuevos algoritmos, fruto de las combinaciones (1a2b) y (1a2c), a los que llamaremos en adelante *método WM₂* y *método WM₃*, respectivamente.

2.2. Una Nueva Metodología para Mejorar la Precisión Obteniendo Reglas con Mejor Cooperación

Una vez analizado el funcionamiento de los métodos específicos de envoltura de ejemplos, en esta sección nos proponemos presentar una nueva metodología para

generar modelos lingüísticos eficaces y simples considerando la cooperación global que desempeñan las reglas lingüísticas de la BC. Siguiendo esta metodología, introduciremos también dos nuevos métodos guiados respectivamente por ejemplos y por rejilla difusa.

2.2.1. La Metodología de Cooperación entre Reglas

Una de las propiedades más interesantes de un SBRD es el razonamiento interpolado que desarrolla. Esta característica desempeña un importante papel en el buen funcionamiento de los SBRDs y es una consecuencia de la cooperación entre las reglas lingüísticas que componen la BC. Como es sabido, la salida obtenida por un SBRD no se debe normalmente a una sola regla lingüística, sino a la actuación cooperativa de varias reglas que actúan al emparejarse con el sistema en un cierto grado.

Sin embargo, el procedimiento seguido por los métodos específicos de envoltura de ejemplos comentados, independientemente de estar guiados por ejemplos o por rejilla difusa, consiste en agrupar los ejemplos en subespacios difusos según un grado de envoltura y obtener después la regla con el mejor comportamiento en cada subespacio. De esta forma no se considera la interacción global entre las reglas de la BC. Esto causa que la BR finalmente obtenida, a pesar de presentar un buen comportamiento local, no presente un nivel de cooperación adecuado. Además, el hecho de procesar localmente estas reglas provoca que estos métodos sean más sensibles frente al ruido (datos erróneos) que pudiera contener el conjunto de ejemplos.

Con el propósito de afrontar estos inconvenientes, vamos a proponer una nueva metodología para mejorar la precisión de los modelos lingüísticos obteniendo mayor cooperación entre las reglas: la metodología COR. Dicha metodología realiza una *búsqueda combinatoria de reglas cooperativas* sobre el conjunto de reglas difusas lingüísticas candidatas para encontrar el conjunto de reglas final que mejor coopera. En lugar de seleccionar el mejor consecuente en cada subespacio, la metodología COR considera la posibilidad de usar otro, diferente del mejor, si con ello hace que el SBRD sea más preciso gracias a tener una BC más cooperativa. Con esta intención, COR realiza una búsqueda combinatoria entre las reglas candidatas tratando de obtener el conjunto de consecuentes con mayor precisión según un criterio de calidad global.

Para realizar esta búsqueda combinatoria se puede considerar una *enumeración explícita* o una *técnica de búsqueda aproximada*:

- Con la enumeración explícita se efectúa un recorrido completo en el conjunto de combinaciones posibles. Aunque con esta técnica nos aseguramos de que se obtiene la mejor solución posible, su ejecución puede ser extremadamente lenta, o simplemente inviable en términos de tiempo, cuando se trabaja con un gran número de combinaciones. Por tanto, esta técnica sólo es recomendable para espacios de búsqueda reducidos.
- Por otro lado, cuando resulta intratable usar la enumeración explícita, se requiere el uso de una técnica aproximada de búsqueda. Puede emplearse cualquier técnica pero, dado que una de las principales ventajas de los métodos específicos de envoltura de ejemplos es su habilidad para encontrar modelos lingüísticos adecuados rápidamente, la técnica de búsqueda debería ser a la vez eficaz y rápida. En la sección 2.3 analizaremos el empleo de algunas metaheurísticas concretas.

De este modo, la metodología COR sólo incide en la segunda fase del esquema genérico seguido por los métodos específicos de envoltura de ejemplos presentado en la sección 2.1.2.4 (figura 2.7). En la figura 2.9 se muestra un ejemplo detallado de su funcionamiento.

Independientemente de la técnica de búsqueda combinatoria empleada, podemos obtener nuevos métodos muy eficaces siguiendo la metodología COR para desarrollar el segundo paso, junto con cualquier método específico de envoltura de ejemplos que realice el primer paso del esquema genérico. Por ejemplo, de la combinación de esta metodología con los dos métodos analizados con anterioridad, métodos WM y CH, surgen dos alternativas: el *enfoque CORWM* y el *enfoque CORCH*.

2.2.2. Análisis de la Metodología de Cooperación entre Reglas

En la figura 2.10 queda reflejado gráficamente el proceso de generación de reglas basado en la metodología COR según se use el método WM o el CH para extraer los consecuentes en cada subespacio (primer paso del algoritmo). De esta forma, el enfoque CORWM considerará las regiones coloreadas en gris claro mientras que el enfoque CORCH tiene en cuenta, además, las zonas gris oscuro. Este hecho afectará al número de reglas generado (como comentamos en la sección 2.1.2.3) y al conjunto de consecuentes candidatos considerado en cada subespacio.

Puesto que en el enfoque CORCH, cada ejemplo puede participar en la obtención de varias reglas en el primer paso, normalmente se generará más diversidad de

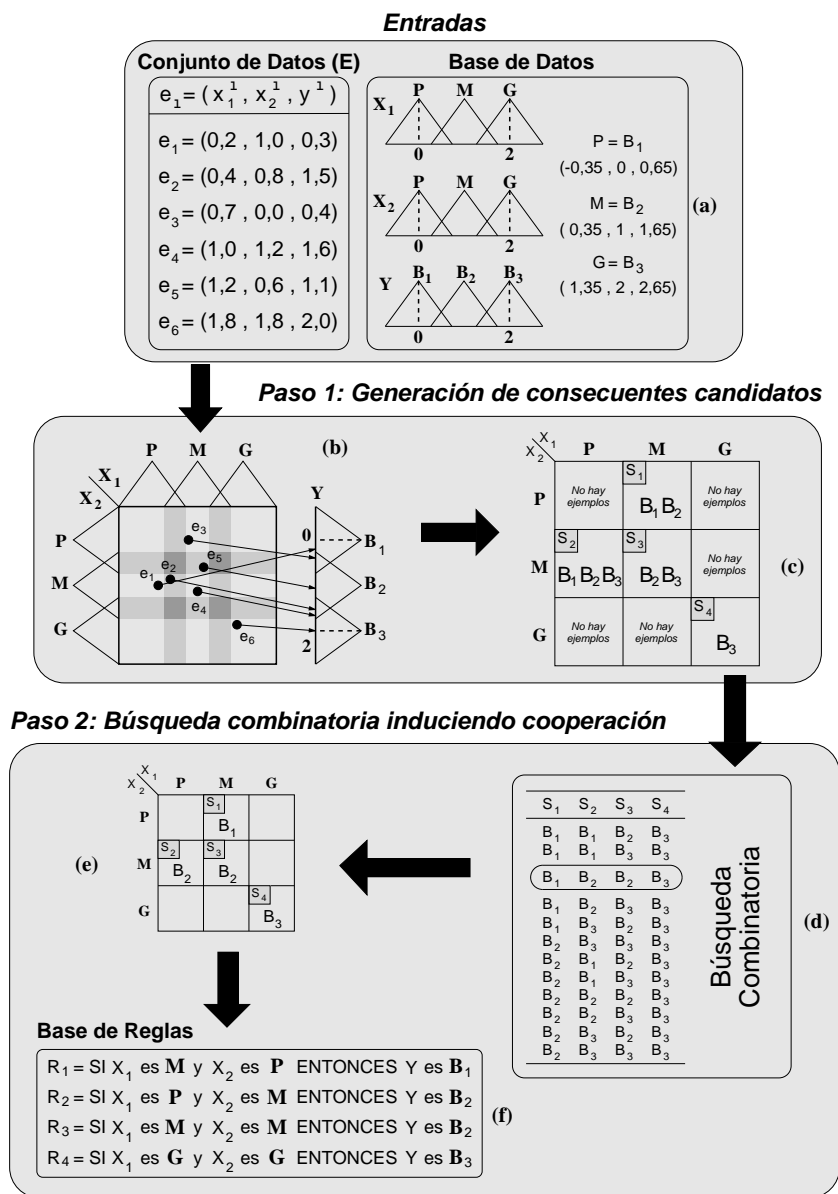


Figura 2.9.: Proceso de aprendizaje basado en COR para un problema sencillo con dos variables de entrada ($n = 2$) y tres etiquetas en la partición difusa de salida ($|\mathcal{B}| = 3$): (a) conjunto de datos (E) y BD definidos previamente; (b) los seis ejemplos ($N = 6$) se encuentran situados en cuatro ($N_S = 4$) subespacios distintos (según el enfoque guiado por rejilla difusa) que determinan las combinaciones del antecedente y consecuentes candidatos de las reglas; (c) conjunto de consecuentes candidatos para cada subespacio; (d) búsqueda combinatoria realizada dentro del espacio compuesto por las doce combinaciones posibles de consecuentes; (e) tabla de reglas de decisión para la tercera combinación; (f) BR generada a partir de la tercera combinación

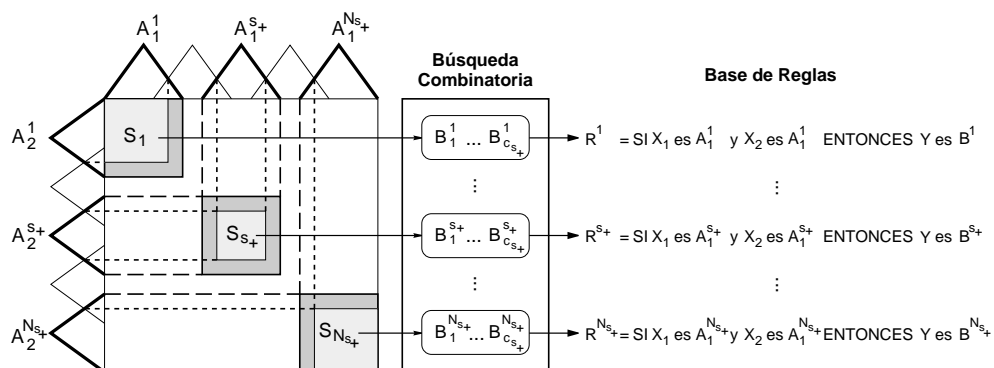


Figura 2.10.: Proceso de generación de reglas seguido por los enfoques CORWM (zonas gris claro) y CORCH (zonas gris claro y gris oscuro)

consecuentes candidatos en cada subespacio y, por tanto, el espacio de soluciones explorado por el proceso de búsqueda combinatoria será más extenso que en el enfoque CORWM (que está guiado por ejemplos). Este hecho tiene pros y contras que deben ser analizados con detenimiento antes de inclinarse por uno u otro enfoque. Por un lado, el algoritmo accede a un juego de combinaciones posibles mayor que le permite encontrar mejores soluciones. Sin embargo, esta diversidad aumenta el tamaño del espacio de búsqueda e incrementa el riesgo de que el algoritmo caiga en óptimos locales. Por el contrario, el enfoque CORWM tiende a generar menos combinaciones en el primer paso, lo que podría beneficiar el proceso de búsqueda del segundo paso. La conveniencia de seguir uno u otro enfoque dependerá de la naturaleza del problema —representada por el conjunto de ejemplos (número de ejemplos y su distribución)—, que será quien determine la diversidad del conjunto de consecuentes candidatos en cada subespacio según el enfoque escogido.

A diferencia de otros mecanismos de aprendizaje de la BR que también consideran un criterio de bondad en la que actúan todas las reglas globalmente (por ejemplo, el método propuesto por Thrift en [Thr91]), la metodología COR puede considerarse como una herramienta flexible (se pueden emplear diferentes técnicas de búsqueda) que reduce el espacio de búsqueda de forma heurística, haciendo así que el proceso de aprendizaje sea más rápido y obtenga modelos precisos.

Por ejemplo, en la figura 2.11 se muestra el comportamiento comparativo del método de Thrift y la metodología COR al modelar el problema de la línea de baja tensión (véase el apéndice C) con cinco términos lingüísticos en cada una de las

particiones difusas de las variables. Ante la distribución de ejemplos abordada —figura 2.11(a)—, el método de Thrift (en la sección 1.5.1.1 del capítulo 1 puede consultar su descripción) considera todos los subespacios de entrada difusos disponibles y todos los consecuentes posibles en cada subespacio incluyendo el valor nulo (que supone la eliminación de la correspondiente regla) —figura 2.11(c)—. Sin embargo, en la metodología COR, sólo se considera un subconjunto de subespacios y de consecuentes candidatos en cada uno de ellos —figura 2.11(d)—. Este significativo recorte reduce notablemente el tamaño del espacio de soluciones —figura 2.11(b)— y facilita el proceso de búsqueda. Dicha reducción se realiza a partir de la información heurística disponible en dos aspectos: en primer lugar, sólo se emplean aquellas reglas lingüísticas que representan subespacios con ejemplos positivos y, en segundo lugar, en cada subespacio sólo se considera un subconjunto de consecuentes posibles.

Podemos decir que los métodos basados en COR están a medio camino entre la simplicidad y celeridad de los métodos específicos de envoltura de ejemplos clásicos, y la complejidad y lentitud del resto de mecanismos de generación de reglas, encontrándose quizás más cercanos a los primeros. Su comportamiento dependerá en gran medida de la técnica de búsqueda combinatoria usada y del tamaño del espacio de soluciones que esta maneje.

2.3. Uso de Diferentes Técnicas de Búsqueda en la Metodología COR

En esta sección vamos a proponer el uso de cuatro técnicas de búsqueda combinatoria dentro de la metodología COR. En concreto, consideraremos dos metaheurísticas de búsqueda por entornos —enfriamiento simulado (ES) y búsqueda tabú (BT)— y dos de búsqueda global —AGs y algoritmos de optimización basados en colonias de hormigas (OCH)—. En el apéndice B se puede consultar una introducción general a estas técnicas.

2.3.1. Aplicación del Enfriamiento Simulado a la Metodología COR

El método de aprendizaje basado en la metodología COR que emplea el ES como técnica de búsqueda se caracteriza por los siguientes aspectos:

- *Representación* — Se empleará un vector de números enteros (c) de tamaño N_{S_+} (número de subespacios de entrada difusos con ejemplos positivos). Cada

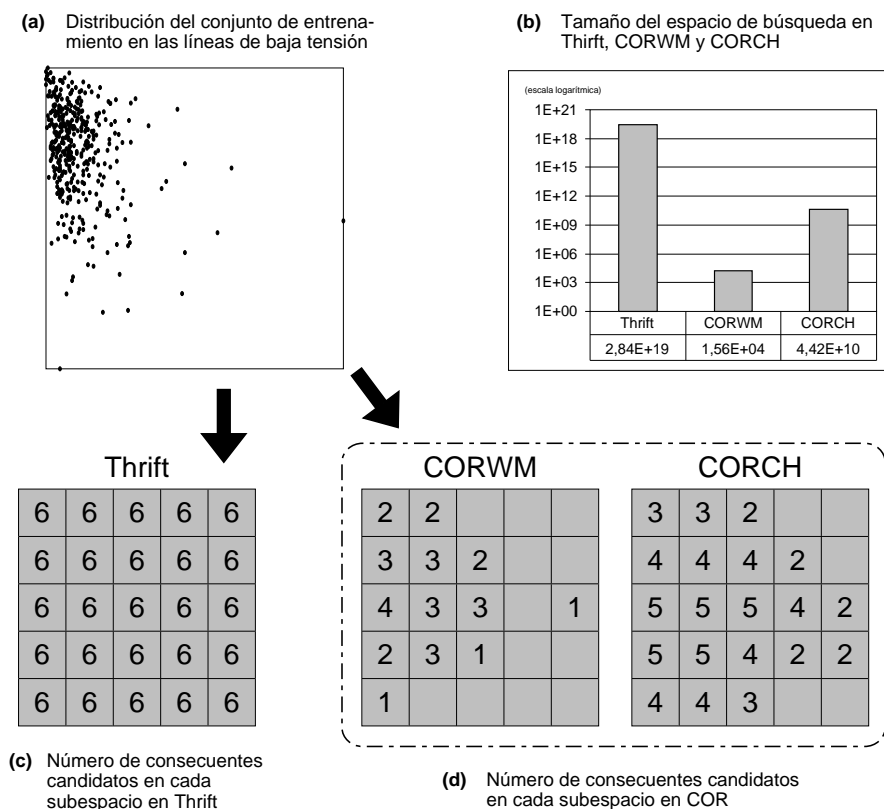


Figura 2.11.: Comparación entre el espacio de búsqueda manejado por el método Thirft y la metodología COR en el problema de la línea de media tensión con cinco términos lingüísticos en cada variable

celda del vector representa el índice del consecuente usado para construir la regla lingüística en el correspondiente subespacio:

$$\forall s_+ \in \{1, \dots, N_{S_+}\}, c[s_+] = k_{s_+} \text{ tal que } B_{k_{s_+}} \in C^{s_+}.$$

- *Función objetivo* — El objetivo será minimizar la función del *error cuadrático medio*:

$$ECM = \frac{1}{2 \cdot N} \sum_{l=1}^N (F(x_1^l, \dots, x_n^l) - y^l)^2,$$

donde $F(x_1^l, \dots, x_n^l)$ es la salida obtenida a partir del SBRD cuando se considera el ejemplo e_l como entrada e y^l es la salida conocida deseada. Cuanto más

cercano a cero sea el valor de esta medida, mayor será el rendimiento global y, por tanto, mejor será la cooperación entre reglas.

- *Esquema de enfriamiento* — Se empleará el esquema de enfriamiento exponencial propuesto por Kirkpatrick [Kir84] ($T_{t+1} = C \cdot T_t$).
- *Criterio de equilibrio* — El equilibrio en una temperatura específica se alcanzará cuando se haya generado un número máximo de vecinos o se haya conseguido un determinado número de éxitos.
- *Solución inicial* — La solución inicial se obtendrá generando aleatoriamente una combinación válida.
- *Generación de vecinos* — El operador considerado para obtener un vecino consistirá en seleccionar aleatoriamente un subespacio difuso específico ($s_+ \in \{1, \dots, N_{S_+}\}$) que contenga al menos dos consecuentes candidatos ($c^{s_+} \geq 2$) y cambiar aleatoriamente el consecuente empleado actualmente (k^{s_+}) por otro ($k^{s_+'}$) perteneciente al conjunto de consecuentes candidatos ($B_{k^{s_+'}}^{s_+} \in C^{s_+}$) distinto del actual ($k^{s_+'} \neq k^{s_+}$).
- *Criterio de parada* — El algoritmo parará cuando no se acepte ningún vecino en una temperatura específica.

De la combinación de esta técnica de búsqueda con los dos enfoques basados en COR propuestos en la sección anterior, surgen dos nuevos métodos de aprendizaje, a los que llamaremos *método CORWM-ES* y *método CORCH-ES*.

2.3.2. Aplicación de la Búsqueda Tabú a la Metodología COR

El método de aprendizaje basado en la metodología COR que emplea la BT como algoritmo de búsqueda consta de los siguientes componentes:

- *Representación* — Se empleará la propuesta para la técnica de ES (sección 2.3.1).
- *Función objetivo* — Se usará la función ECM mencionada en la sección 2.3.1 para el ES.

- *Criterio tabú* — La lista tabú (o registro histórico) contiene los movimientos aplicados para obtener las soluciones aceptadas más recientemente. El movimiento considerado es el par subespacio–consecuente, es decir (s_+, k^{s_+}) .
- *Criterio de aspiración* — Se permitirá realizar movimientos que obtengan soluciones mejores que la mejor solución global obtenida hasta el momento, incluso si dichos movimientos son tabú activos.
- *Solución inicial* — Se obtiene del mismo modo que en el ES (sección 2.3.1).
- *Generación de vecinos* — El operador será similar al empleado en el ES (sección 2.3.1).
- *Reinicialización de intensificación* — Se reemplazará la solución actual por la mejor obtenida hasta el momento cuando se satisfaga un *criterio de convergencia de intensificación* concreto. Además, se variará el tamaño de la lista tabú aleatoriamente reduciéndola o incrementándola en un 50%.

El criterio de convergencia consistirá en aplicar la reinicialización cuando la *mejor solución global* no mejore durante un número específico de iteraciones.

- *Reinicialización de diversificación* — Se empleará una *memoria a largo plazo* para llevar la cuenta de la frecuencia de uso de cada uno de los movimientos aceptados. Para hacer esto, consideraremos una matriz *LTM* inicializada a cero en la que, cada vez que se acepte un movimiento (digamos (s_+, k^{s_+})), se incrementará el correspondiente contador en uno, es decir, $LTM[s_+][k^{s_+}] \leftarrow LTM[s_+][k^{s_+}] + 1$.

Posteriormente, según un *criterio de diversificación concreto*, se generará una nueva solución seleccionando un consecuente en cada subespacio de acuerdo a una probabilidad inversamente proporcional al valor correspondiente de *LTM*:

$$c[s_+] = k^{s_+} \text{ con una probabilidad de } \frac{1}{LTM[s_+, k^{s_+}] + 1} \cdot \frac{1}{\sum_{B_{q^{s_+}}^{s_+} \in C^{s_+}} \frac{1}{LTM[s_+, q^{s_+}] + 1}}.$$

El criterio de convergencia se cumplirá cuando la *mejor solución* no se haya mejorado durante un número específico de iteraciones.

De la combinación de esta técnica de búsqueda con los dos enfoques basados en COR propuestos en la sección anterior, surgen dos nuevos métodos de aprendizaje, a los que llamaremos *método CORWM-BT* y *método CORCH-BT*.

2.3.3. Aplicación de los Algoritmos Genéticos a la Metodología COR

El método de aprendizaje basado en la metodología COR que usa un AG como técnica de búsqueda será el siguiente:

- *Esquema de codificación* — Será similar a la representación propuesta en la sección 2.3.1 para el ES.
- *Función de adaptación* — Se usará la función ECM mencionada en la sección 2.3.1 para el ES.
- *Enfoque genético* — Consideraremos un AG generacional con el procedimiento de muestreo universal estocástico de Baker [Bak87] y un criterio elitista consistente en asegurar que el mejor individuo de la población anterior permanece en la nueva.
- *Población inicial* — La población se genera con el primer individuo de la siguiente forma:

$$\forall s_+ \in \{1, \dots, N_{s_+}\},$$

$$c_1[s_+] = k^{s_+} \text{ tal que } B_{k_{s_+}} = \arg \max_{B_q \in C^{s_+}} FVR(R_q^{s_+}),$$

donde la función *FVR* puede ser cualquiera de las tres posibilidades propuestas en la sección 2.1.2.2. El resto de cromosomas se generan aleatoriamente:

$$\forall p \in \{2, \dots, \text{tam_pob}\}, \forall s_+ \in \{1, \dots, N_{s_+}\}, c_p[s_+] = k_{s_+} \text{ tal que } B_{k_{s_+}} \in C^{s_+}.$$

- *Cruce* — Se usará el operador multipunto en dos puntos.
- *Mutación* — Este operador será similar al mecanismo de generación de vecinos propuesto para el ES (sección 2.3.1).

De la combinación de esta técnica de búsqueda con los dos enfoques basados en COR propuestos en la sección anterior, surgen dos nuevos métodos de aprendizaje, a los que llamaremos *método CORWM-AG* y *método CORCH-AG*.

2.3.4. Aplicación de los Algoritmos de Optimización Basados en Colonias de Hormigas a la Metodología COR

A continuación, presentamos los principales aspectos que componen el algoritmo de OCH basado en la metodología COR que proponemos:

- *Representación del problema* — Para usar algoritmos de OCH en la metodología COR, resulta conveniente considerar esta como un problema de optimización combinatoria con la capacidad de poder representarse mediante un grafo. De este modo, podemos afrontar el problema interpretando la metodología COR como la forma de asignar consecuentes ($B_j \in \mathcal{B}$) —es decir, etiquetas de la partición difusa de salida— a subespacios difusos n -dimensionales de entrada con ejemplos positivos (S_{s_+}) atendiendo a un criterio de optimalidad (error cuadrático medio).

Por lo tanto, estaríamos trabajando con una tarea de asignación, de modo que la representación del problema puede ser similar a la tradicionalmente empleada en el campo de la OCH para resolver el problema de la asignación cuadrática —en inglés, *quadratic assignment problem* (QAP)— (véase una introducción en el apéndice B.4) [BDT99, MC99], pero con algunas peculiaridades. Podemos establecer una analogía entre *subespacios* difusos de entrada con ejemplos positivos y localizaciones y entre etiquetas en el *consecuente* y unidades. Sin embargo, a diferencia del QAP, el *conjunto de consecuentes conectados con cada subespacio puede ser diferente y es posible asignar un mismo consecuente a más de un subespacio* (dos reglas lingüísticas pueden tener el mismo consecuente). Podemos deducir a partir de estas características que el orden seguido para seleccionar cada uno de los subespacios a los que se les irá asignando un consecuente no es determinante ya que una asignación no restringe al resto, es decir, el orden de la asignación es irrelevante.

Así, el grafo se construirá realizando los pasos descritos en la figura 2.12. Siguiendo estos pasos, el grafo correspondiente al ejemplo de aprendizaje con COR presentado en la figura 2.9 sería el que recoge la figura 2.13.

Para construir una solución completa, una hormiga irá recorriendo cada subespacio y eligiendo un consecuente con una probabilidad que dependerá de la cantidad de feromona $\tau_{s_+,j}$ y de la información heurística $\eta_{s_+,j}$. Como ya hemos comentado, el orden para recorrer los distintos subespacios es irrelevante.

1. *Establecer un nodo para cada subespacio* — Usar un nodo para cada subespacio de entrada difuso n -dimensional que contenga ejemplos positivos (S_{s_+}), obteniendo así un total de N_{S_+} nodos subespacio.
2. *Conectar los subespacios con consecuentes* — El nodo subespacio S_{s_+} estará conectado con el nodo consecuente B_j , donde $j \in \{1, \dots, |\mathcal{B}|\}$, si y sólo si este consecuente pertenece al conjunto de consecuentes candidatos del subespacio en cuestión ($B_j \in C^{s_+}$).

Figura 2.12.: Proceso de construcción del grafo para el algoritmo COR basado en OCH

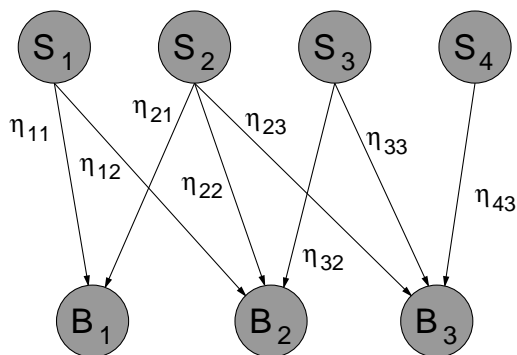


Figura 2.13.: Grafo del algoritmo de OCH correspondiente al ejemplo de la figura 2.9

- *Información heurística* — Para calcular la información heurística (η_{s_+j}) sobre la preferencia potencial de elegir un consecuente específico ($B_j \in \mathcal{B}$) en cada subespacio (S_{s_+}) emplearemos alguna de las funciones FVR , a elegir entre las tres introducidas en la sección 2.1.2.2. Puede observarse que en aquellas asignaciones donde el consecuente no pertenezca al conjunto de consecuentes candidatos del correspondiente subespacio, la información heurística será cero ($B_j \notin C^{s_+} \Rightarrow \eta_{s_+j} = 0$).
- *Inicialización de feromona* — El valor de feromona inicial en cada asignación se obtiene de la siguiente forma:

$$\tau_0 = \frac{\sum_{s_+=1}^{N_{S_+}} \max_{j=1}^{|\mathcal{B}|} \eta_{s_+,j}}{N_{S_+}},$$

es decir, la feromona inicial será el valor medio del camino construido tomando el mejor consecuente en cada subespacio de acuerdo a la información heurística (esto es, en definitiva, una asignación voraz² según el método WM o CH).

- *Función de adecuación* — Una vez más, emplearemos la función objetivo ECM propuesta para el caso del ES (sección 2.3.1).
- *Enfoque de OCH* — Consideraremos el algoritmo sistema de hormigas (SH) [DMC96] (en el apéndice B.4 puede consultarse su funcionamiento). Los únicos aspectos del algoritmo particulares a nuestra aplicación son los siguientes:
 - En la *regla de transición*, el conjunto de nodos alcanzables desde el nodo subespacio S_{s_+} será el conjunto de consecuentes candidatos, es decir,

$$J_k(s_+) = \{j \text{ tal que } \eta_{s_+,j} \neq 0\}.$$

- En la *regla de actualización*, el aporte de feromona de la hormiga k será

$$f(C(S_k)) = \frac{1}{\text{ECM}_k},$$

siendo ECM_k el error cuadrático medio resultante al diseñar el SBRD con la BR construida por la hormiga k .

De la combinación de esta técnica de búsqueda con los dos enfoques basados en COR propuestos en la sección anterior, surgen dos nuevos métodos de aprendizaje, a los que llamaremos *método CORWM-SH* y *método CORCH-SH*.

2.4. Estudio Experimental de la Metodología COR

En esta sección analizaremos el comportamiento de la metodología de aprendizaje propuesta al resolver cuatro problemas de modelado lingüístico diferentes: el

² Este término, proveniente del inglés *greedy*, se emplea dentro del campo de las metaheurísticas para designar a un tipo de algoritmos diseñados *ad hoc* que generan soluciones aceptables mediante un proceso constructivo basado en información heurística

modelado de la función tridimensional F_1 , el modelado de la función tridimensional F_2 , la estimación de la calidad del arroz y la estimación de la longitud de línea eléctrica de baja tensión necesaria en zonas rurales. En el apéndice C se detallan estas aplicaciones y las consideraciones que se han tomado en cuenta para realizar la experimentación.

Para comprobar comparativamente el funcionamiento de la metodología, incluiremos tres métodos de aprendizaje específicos de envoltura de ejemplos propuestos en la literatura, los dos métodos analizados en este capítulo (WM y CH en sus diferentes versiones: WM, WM₂, WM₃, CH₁, CH₂ y CH₃) y el método propuesto por Nozaki, Ishibuchi y Tanaka [NIT97] (*método NIT*) descrito en el capítulo anterior. Este método resuelve el problema de seleccionar localmente las reglas mediante el uso de reglas ponderadas con doble consecuente, así que puede considerarse como otra alternativa para favorecer la cooperación y su inclusión en el estudio comparativo resulta necesaria. Dado que el objetivo del presente análisis es la generación de BRs, en el método NIT consideraremos $\alpha = 1$ (sin factor de escala) para realizar una comparación más objetiva. Por otro lado, también incluiremos un método de generación más complejo que emplea un AG, el método propuesto por Thrift [Thr91] (*método T*) descrito en el capítulo anterior. A diferencia del resto, este algoritmo no establece un número de reglas fijo determinado por los subespacios con ejemplos positivos sino que busca el mejor conjunto de reglas entre todas las combinaciones de antecedentes posibles.

En la tabla 2.1 se recogen los distintos métodos considerados en este estudio experimental.

2.4.1. Tablas de Resultados y Valores de Parámetros

Las tablas 2.2 y 2.3 muestran los resultados obtenidos. En ellas, #R indica el número de reglas, ECM_{entr} y ECM_{prue} los valores del error cuadrático medio sobre los conjuntos de entrenamiento y prueba, y EMS el número de evaluaciones necesarias para obtener la mejor solución. Este último valor indica objetivamente la velocidad del algoritmo ya que el tiempo empleado será un múltiplo de dicho valor, que variará según la velocidad de cálculo del computador donde se realice la experimentación. En el problema del arroz, se incluye para cada valor la media obtenida con los diez modelos (véase la sección C.2.4 del apéndice C). Se muestran en **negrita** los mejores resultados para cada problema.

En la tabla 2.4 se recogen los valores de parámetros empleados por cada método

Tabla 2.1.: Métodos considerados en el estudio experimental de la metodología COR

Método	Enfoque	Técnica de búsqueda
WM	Guiado por ejemplos	—
WM ₂	Guiado por ejemplos	—
WM ₃	Guiado por ejemplos	—
CH ₁	Guiado por rejilla difusa	—
CH ₂	Guiado por rejilla difusa	—
CH ₃	Guiado por rejilla difusa	—
NIT	Guiado por rejilla difusa con dos reglas por subespacio	—
T	—	Algoritmo genético
CORWM-ES	Guiado por ejemplos	Enfriamiento simulado
CORWM-BT	Guiado por ejemplos	Búsqueda tabú
CORWM-AG	Guiado por ejemplos	Algoritmo genético
CORWM-SH	Guiado por ejemplos	Sistema de hormigas
CORCH-ES	Guiado por rejilla difusa	Enfriamiento simulado
CORCH-BT	Guiado por rejilla difusa	Búsqueda tabú
CORCH-AG	Guiado por rejilla difusa	Algoritmo genético
CORCH-SH	Guiado por rejilla difusa	Sistema de hormigas

en cada problema. Los parámetros TP , $\#G$, P_c y P_m de los métodos T y COR-AG indican respectivamente el tamaño de la población, el número de generaciones y las probabilidades de cruzar y mutar. En los métodos COR-SA, T_0 indica la temperatura inicial, $\#V$ el número máximo de vecinos generados en cada temperatura y $\#A$ el número máximo de aceptaciones antes de realizar el enfriamiento. En los métodos COR-BT, $\#It.$ indica el número de iteraciones, $\#V$ el número de vecinos a generar y los parámetros CI y CD indican los factores de convergencia (en número de iteraciones) de intensificación y diversificación, respectivamente. Por último, en los métodos COR-SH, el número de hormigas será el número de subespacios de entrada con ejemplos positivos en cada caso (N_{S_+}), el número de iteraciones será 50 y para el resto de parámetros (η , ρ , α y β) se muestran los valores con los que se obtuvo los mejores resultados.

Tabla 2.2.: Resultados obtenidos por los métodos basados en la metodología COR y el resto de métodos de comparación en las dos funciones tridimensionales

Método	F_1			F_2				
	#R	ECM _{entr}	ECM _{prue}	EMS	#R	ECM _{entr}	ECM _{prue}	EMS
WM	49	2,048137	2,287129	—	49	0,194386	0,044466	—
WM ₂	49	2,740901	2,505504	—	49	0,194386	0,044466	—
WM ₃	49	2,048137	2,287129	—	49	0,194386	0,044466	—
CH ₁	49	2,048137	2,287129	—	49	0,194386	0,044466	—
CH ₂	49	3,755082	3,393716	—	49	0,168581	0,076570	—
CH ₃	49	2,740901	2,505504	—	49	0,097397	0,043984	—
NIT	98	2,465487	1,768125	—	98	0,175382	0,061249	—
T	49	1,609890	1,193721	27.301	47	0,067518	0,032442	29.572
CORWM-ES	49	1,605482	1,132797	6.084	49	0,097397	0,043984	5.909
CORWM-BT	49	1,669564	1,275750	1.668	49	0,097397	0,043984	2.981
CORWM-AG	49	1,596661	1,392013	12.736	49	0,097397	0,043984	5.289
CORWM-SH	49	1,677016	1,657133	483	49	0,097397	0,043984	116
CORCH-ES	49	1,609891	1,213388	6.924	49	0,097397	0,043984	6.690
CORCH-BT	49	2,618359	2,188273	1.708	49	0,097397	0,043984	3.253
CORCH-AG	49	1,620743	1,601233	17.743	49	0,097397	0,043984	11.030
CORCH-SH	49	1,660622	1,419587	678	49	0,097397	0,043984	1

Tabla 2.3.: Resultados obtenidos por los métodos basados en la metodología COR y el resto de métodos de comparación en los problemas reales

Método	Arroz				Línea de baja tensión			
	#R	ECM _{entr}	ECM _{prue}	EMS	#R	ECM _{entr}	ECM _{prue}	EMS
WM	15	0,013284	0,013119	—	24	222,654	239,962	—
WM ₂	15	0,014841	0,015127	—	24	207,269	198,383	—
WM ₃	15	0,017049	0,017320	—	24	207,269	198,383	—
CH ₁	32	0,052208	0,053694	—	32	239,393	275,953	—
CH ₂	32	0,021293	0,021266	—	32	275,952	269,668	—
CH ₃	32	0,009101	0,010123	—	32	268,005	249,601	—
NIT	64	0,008626	0,009851	—	64	185,395	170,489	—
T	16	0,004949	0,005991	10,021	49	169,077	175,739	26,706
CORWM-ES	15	0,007979	0,008244	21	24	181,911	175,267	84
CORWM-BT	15	0,007979	0,008244	33	24	181,911	175,267	168
CORWM-AG	15	0,007979	0,008244	23	24	181,911	175,267	1,494
CORWM-SH	15	0,007979	0,008244	13	24	181,911	175,267	56
CORCH-ES	32	0,007076	0,008012	3,963	32	174,295	161,261	1,129
CORCH-BT	32	0,007213	0,008398	433	32	174,785	186,840	1,312
CORCH-AG	32	0,006845	0,007830	5,123	32	171,830	174,987	15,398
CORCH-SH	32	0,006943	0,007702	1,108	32	178,119	158,662	359

Tabla 2.4.: Valores de los parámetros considerados en cada método para cada problema

Método	F_1	F_2
	Valores de parámetros	Valores de parámetros
T	$TP = 61, \#G = 1000, P_c = 0,6, P_m = 0,2$	$TP = 61, \#G = 1000, P_c = 0,6, P_m = 0,2$
CORWM-ES	$T_0 = 40, \#V = 98, \#A = 49$	$T_0 = 40, \#V = 98, \#A = 49$
CORWM-BT	$\#It. = 150, \#V = 49, CI = 10, CD = 5$	$\#It. = 150, \#V = 49, CI = 10, CD = 5$
CORWM-AG	$TP = 61, \#G = 500, P_c = 0,6, P_m = 0,2$	$TP = 61, \#G = 500, P_c = 0,6, P_m = 0,2$
CORWM-SH	$\eta = FVR_1, \rho = 0,4, \alpha = 1, \beta = 1$	$\eta = FVR_3, \rho = 0,8, \alpha = 2, \beta = 1$
CORCH-ES	$T_0 = 40, \#V = 98, \#A = 98$	$T_0 = 40, \#V = 98, \#A = 49$
CORCH-BT	$\#It. = 150, \#V = 49, CI = 10, CD = 5$	$\#It. = 150, \#V = 49, CI = 10, CD = 5$
CORCH-AG	$TP = 61, \#G = 500, P_c = 0,6, P_m = 0,2$	$TP = 61, \#G = 500, P_c = 0,6, P_m = 0,2$
CORCH-SH	$\eta = FVR_1, \rho = 0,2, \alpha = 1, \beta = 2$	$\eta = FVR_3, \rho = 0, \alpha = 1, \beta = 12$

Método	Arroz	Línea de baja tensión
	Valores de parámetros	Valores de parámetros
T	$TP = 61, \#G = 500, P_c = 0,6, P_m = 0,2$	$TP = 61, \#G = 1000, P_c = 0,6, P_m = 0,4$
CORWM-ES	$T_0 = 30, \#V = 15, \#A = 15$	$T_0 = 5000, \#V = 24, \#A = 24$
CORWM-BT	$\#It. = 150, \#V = 49, CI = 10, CD = 5$	$\#It. = 150, \#V = 49, CI = 10, CD = 5$
CORWM-AG	$TP = 61, \#G = 50, P_c = 0,6, P_m = 0,2$	$TP = 61, \#G = 500, P_c = 0,6, P_m = 0,2$
CORWM-SH	$\eta = FVR_3, \rho = 0,6, \alpha = 1, \beta = 1$	$\eta = FVR_2, \rho = 0,6, \alpha = 1, \beta = 2$
CORCH-ES	$T_0 = 70, \#V = 32, \#A = 32$	$T_0 = 500, \#V = 32, \#A = 32$
CORCH-BT	$\#It. = 150, \#V = 49, CI = 10, CD = 5$	$\#It. = 150, \#V = 49, CI = 10, CD = 5$
CORCH-AG	$TP = 61, \#G = 500, P_c = 0,6, P_m = 0,2$	$TP = 61, \#G = 500, P_c = 0,6, P_m = 0,2$
CORCH-SH	$\eta = FVR_2, \rho = 0,6, \alpha = 1, \beta = 2$	$\eta = FVR_2, \rho = 0,6, \alpha = 1, \beta = 2$

2.4.2. Análisis de los Resultados Obtenidos

Vamos a realizar el análisis desde tres puntos de vista distintos. En primer lugar, estudiaremos el comportamiento de los métodos específicos de envoltura de ejemplos básicos, los métodos WM y CH, analizando los efectos que causa seguir el enfoque guiado por ejemplos o por rejilla difusa, respectivamente. A continuación, analizaremos la metodología COR frente al resto de métodos considerados. Por último, veremos las diferencias existentes entre las cuatro técnicas de búsqueda aplicadas a COR propuestas.

2.4.2.1. Métodos Guiados por Ejemplos frente a Métodos Guiados por Rejilla Difusa

Analizando los resultados obtenidos por las distintas variantes de los métodos WM y CH, podemos comprobar que el método WM se ve menos afectado por la función de evaluación empleada para valorar la calidad de cada consecuente, mostrando las diferentes variantes del método un comportamiento parecido en todos los problemas. Esto puede deberse al hecho de generar un número menor de consecuentes candidatos en cada regla, que hace que la valoración de cada uno de ellos no resulte tan importante. No sucede lo mismo en el método CH, donde podemos comprobar que la elección de dicha función parece crucial.

Resulta interesante observar que el efecto de la función de evaluación seleccionada difiere según el enfoque de aprendizaje en el problema del arroz. Mientras que en las variantes de WM (métodos guiados por ejemplos) se obtienen mejores resultados con una valoración de reglas egoísta (FVR_1 , método original WM), en los métodos CH (guiados por rejilla difusa) los mejores resultados se consiguen empleando un criterio más cooperativo (FVR_3 , variante CH₃). Esto puede estar relacionado con el número de reglas generadas, ya que en el método CH, al cohabitar más reglas, existe una mayor interdependencia.

Analizando los resultados obtenidos en el problema eléctrico, comprobamos que los métodos basados en WM se comportan mejor que las variantes de CH. Este hecho está relacionado con la naturaleza del problema, que contiene ruido debido a su procedencia. Los métodos basados en ejemplos tratan estos puntos espurios generando una regla para cada uno de ellos y resulta difícil que dichas reglas participen en el proceso de inferencia cuando se evalúan datos nuevos. Sin embargo, dichos puntos contribuyen a generar varias reglas en los métodos basados en rejilla, lo que incrementa la distorsión. Por otro lado, en las variantes del método WM, el uso de criterios promediados (valoración según FVR_2 y FVR_3 en los métodos WM₂ y WM₃, respectivamente) evita el excesivo dominio de las reglas candidatas generadas por estos ejemplos erróneos. Por tanto, los métodos basados en ejemplos que consideran criterios cooperativos para seleccionar las reglas candidatas son más robustos frente al ruido, obteniendo modelos más precisos.

2.4.2.2. Métodos Basados en COR frente al Resto

En vista de los resultados obtenidos, podemos apreciar que los métodos basados en COR mejoran significativamente la precisión de los modelos generados en comparación con métodos no cooperativos como las variantes de WM y CH. Es interesante destacar que, en el problema eléctrico, la consideración de cooperación entre las reglas lingüísticas reduce el efecto del ruido observado en los métodos no cooperativos, especialmente en las variantes de CH.

En la figura 2.14 mostramos cómo, a partir del conjunto de consecuentes candidatos generados en cada subespacio por el enfoque guiado por ejemplos para el problema F_1 (figura 2.14(a)), el método WM realiza la selección considerando la mejor regla en cada subespacio (figura 2.14(b)), mientras que el método CORWM-ES hace uso de un criterio cooperativo que tiene en cuenta el comportamiento global de las reglas (figura 2.14(c)). De esta forma, el segundo método selecciona un consecuente distinto del mejor local en las celdas mostradas en negrita e itálica, haciendo que la precisión del modelo obtenido sea notablemente mejor. En la figura 2.15 mostramos otro ejemplo del comportamiento seguido por el método WM y el CORWM-ES, esta vez para una partición de datos concreta en el problema del arroz. De nuevo, el hecho de usar sólo dos consecuentes distintos de los generados por el método WM (indicados en negrita e itálica) hace que el modelo obtenido por el método CORWM-ES sea significativamente más preciso.

Al comparar los resultados obtenidos por la metodología COR y el método NIT, observamos que el primero obtiene en todos los casos modelos lingüísticos con mayor grado de precisión. Mientras que el método NIT construye dos reglas en cada subespacio con los dos mejores consecuentes dando un factor de certeza a cada uno, los métodos basados en COR analizan qué reglas proporcionan la mejor precisión global y únicamente hacen uso de dichas reglas. Así, los modelos obtenidos por COR son más precisos y presentan una BR significativamente más simple, lo que favorece la interpretabilidad. En la figura 2.16 se puede observar la BR generada por el método NIT y el CORCH-SH en el problema eléctrico. A tenor de la experimentación realizada, podemos concluir que un conjunto cooperativo de reglas simples obtiene mejores resultados que un conjunto no cooperativo de reglas con doble consecuente ponderadas.

Frente al método T, sin embargo, la eficacia de la metodología COR resulta contradictoria según el problema donde se aplique. Si bien en el problema F_1 se obtienen modelos más precisos con COR, en las aplicaciones F_2 y arroz resulta claramente

(a) Conjuntos de consecuentes candidatos en cada subespacio siguiendo el enfoque guiado por ejemplos

	X_1					EG
	EP	MP	P	M	G	
EP	{G,M,G,EG}	{M,G,M,G}	{P,M,G}	{P,M}	{P,M,G}	{G,M,G,EG}
MP	{M,G,M,G}	{P,M,G}	{MP,P,M}	{MP,P}	{MP,P,M}	{M,G,M,G}
P	{P,M,G}	{MP,P,M}	{EP,MP,P}	{EP,MP}	{MP,P,M}	{P,M,G}
X_2						
M	{P,M}	{MP,P}	{EP,MP}	{EP}	{MP,P}	{P,M}
G	{P,M,G}	{MP,P,M}	{EP,MP}	{EP,MP}	{MP,P,M}	{P,M,G}
MG	{M,G,M,G}	{P,M,G}	{MP,P,M}	{MP,P}	{MP,P,M}	{M,G,M,G}
EG	{G,M,G,EG}	{M,G,M,G}	{P,M,G}	{P,M}	{P,M,G}	{G,M,G,EG}

(b) BR del método WM

	X_1					MG	EG
	EP	MP	P	M	G		
EP	MG	G	M	M	M	G	MG
MP	G	M	P	MP	P	M	G
P	M	P	MP	EP	MP	P	M
X_2							
M	M	MP	EP	EP	EP	MP	M
G	M	P	MP	EP	MP	P	M
MG	G	M	P	MP	P	M	G
EG	MG	G	M	M	M	G	MG

(c) BR del método CORWM-ES

	X_1					MG	EG
	EP	MP	P	M	G		
EP	EG	G	G	M	M	MG	EG
MP	G	M	MP	MP	P	P	G
P	G	MP	MP	EP	EP	P	M
X_2							
M	M	MP	EP	EP	EP	MP	M
G	M	P	EP	EP	MP	MP	G
MG	G	P	P	MP	MP	M	G
EG	EG	MG	M	M	G	G	EG

Figura 2.14.: Método WM frente al método CORWM-ES modelando F_1

2.4. Estudio Experimental de la Metodología COR

Regla	Sabor	Aspecto	Gusto	Pegajosidad	Dureza	Evaluación		
						Consecuentes candidatos	WM	CORWM-ES
R_1	Malo	Malo	Malo	Suelto	Blando	{Baja}	Baja	Baja
R_2	Malo	Malo	Malo	Suelto	Duro	{Baja,Alta}	Baja	Baja
R_3	Malo	Bueno	Malo	Suelto	Blando	{Baja}	Baja	Baja
R_4	Malo	Bueno	Bueno	Suelto	Blando	{Baja,Alta}	Baja	Alta
R_5	Malo	Bueno	Bueno	Pegajoso	Blando	{Baja}	Baja	Baja
R_6	Bueno	Malo	Malo	Suelto	Blando	{Baja}	Baja	Baja
R_7	Bueno	Malo	Malo	Suelto	Duro	{Baja}	Baja	Baja
R_8	Bueno	Malo	Bueno	Suelto	Blando	{Baja}	Baja	Baja
R_9	Bueno	Bueno	Malo	Suelto	Blando	{Baja,Alta}	Alta	Alta
R_{10}	Bueno	Bueno	Malo	Suelto	Duro	{Baja,Alta}	Baja	Alta
R_{11}	Bueno	Bueno	Malo	Pegajoso	Blando	{Alta}	Alta	Alta
R_{12}	Bueno	Bueno	Bueno	Suelto	Blando	{Alta}	Alta	Alta
R_{13}	Bueno	Bueno	Bueno	Suelto	Duro	{Baja,Alta}	Alta	Alta
R_{14}	Bueno	Bueno	Bueno	Pegajoso	Blando	{Baja,Alta}	Alta	Alta
R_{15}	Bueno	Bueno	Bueno	Pegajoso	Duro	{Alta}	Alta	Alta

Figura 2.15.: Método WM ($ECM_{entr/prue} = 0,014704 / 0,016700$) frente al método CORWM-ES ($ECM_{entr/prue} = 0,008232 / 0,007546$) en una partición de datos concreta del problema del arroz

vencedor el método T. No obstante, en el problema eléctrico, se aprecia cierto sobreaprendizaje en el método T debido al excesivo número de reglas empleadas, que aporta al modelo un alto grado de aproximación perdiendo generalización.

Este comportamiento se encuentra directamente relacionado con una de las principales características del método T: la capacidad de aprender el número de reglas. Aunque esta faceta ocasiona que el espacio de búsqueda se incremente notablemente (véase la figura 2.17), la capacidad de aproximación es mayor.

En efecto, analicemos las BRs generadas en el problema F_2 por el método T y cualquiera de los métodos basados en COR (la figura 2.18 muestra ambos modelos). Podemos observar que las BRs únicamente difieren en tres subespacios, en dos de los cuales el método T no considera ninguna regla —subespacios (P,EP) y (EG,EG)—, mientras que emplea un consecuente distinto en el tercero —subespacio (P,MP)—.

La solución que el método T genera para los subespacios (P,EP) y (P,MP) parece estar relacionado simplemente con haber encontrado una solución subóptima y

2. Metodología COR: Cooperación entre Reglas

		X1						
		EP	MP	P	M	G	MG	EG
EP	EP	0.7279	0.5402	0.9538	0.5061			
	MP	0.2721	0.4598	0.0462	0.4939			
MP	MP	0.7165	0.9403	0.8116	0.5168	0.5454		
	EP	0.2835	0.0597	0.1884	0.4832	0.4546		
P	MP	0.7101	0.5959	0.8317	0.8691	0.5454		
	P	0.2899	0.4041	0.1683	0.1309	0.4546		
X2 M	P	0.9235	0.5623	0.8353	0.9757	0.8974		0.6221
	MP	0.0765	0.4377	0.1647	0.0243	0.1026		0.3779
G	P	0.8028	0.5448	0.7767	0.7451	0.8974		0.6221
	M	0.1972	0.4552	0.2233	0.2549	0.1026		0.3779
MG	P	0.8110	0.6954	0.5309	0.7571			
	M	0.1890	0.3046	0.4691	0.2429			
EG	M	0.8827	0.8827					
	P	0.1173	0.1173					

		X1						
		EP	MP	P	M	G	MG	EG
EP	EP	EP	EP	MP	MP			
	MP	MP	MP	MP	P	P		
P	MP	MP	P	P	M	M		
	X2 M	MP	M	M	MG	MG		P
G	P	P	M	M	EG	MG		P
MG	MP	MP	G	G	M			
	EG	M	M					

Figura 2.16.: BRs generadas por los métodos NIT ($ECM_{entr/prue} = 185.395 / 170.489$) —izquierda— y CORCH-SH ($ECM_{entr/prue} = 178.119 / 158.662$) —derecha— en el problema eléctrico

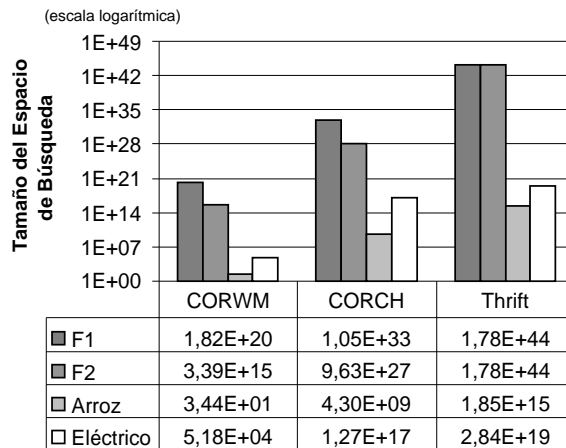


Figura 2.17.: Espacio de búsqueda manejado por los dos enfoques de la metodología COR y el método T

carece de importancia. Sin embargo, el hecho de eliminar la regla correspondiente al subespacio (EG,EG) confiere al modelo obtenido una gran precisión. En efecto, si probamos a eliminar «manualmente» la regla de dicho subespacio en el modelo generado por COR, la precisión obtenida es notablemente mayor ($ECM_{entr/prue} = 0,061158 / 0,031473$). Aún mayor si también eliminamos la regla del subespacio (EP,EP) ($ECM_{entr/prue} = 0,024919 / 0,032345$). Este comportamiento se debe a la naturaleza especial del problema F_2 que, como se describe en el apéndice C, contiene dos discontinuidades en los puntos $(0, 0)$ y $(1, 1)$ —correspondiente a los subespacios (EP,EP) y (EG,EG)—.

Este análisis, que en mayor o menor medida puede trasladarse al resto de aplicaciones, nos sugiere que la eliminación de aquellas reglas que distorsionan el comportamiento del modelo puede favorecer de forma significativa la cooperación entre las reglas y, por consiguiente, la precisión del modelo lingüístico obtenido.

2.4.2.3. Enfoques de Generación de Reglas y Técnicas de Búsqueda en los Métodos Basados en COR

CORWM o CORCH

En cuanto a la preferencia entre el enfoque guiado por ejemplos (CORWM) o el guiado por rejilla difusa (CORCH) en la metodología COR, los resultados de pre-

		X1						
		EP	MP	P	M	G	MG	EG
X2	EP	M	EG		EG	EG	EG	EG
	MP	EP	M	MG	MG	MG	EG	EG
	P	EP	P	M	G	MG	MG	EG
	M	EP	MP	P	M	G	MG	EG
	G	EP	MP	MP	P	M	G	EG
	MG	EP	EP	MP	MP	P	M	EG
	EG	EP	EP	EP	EP	EP	EP	

		X1						
		EP	MP	P	M	G	MG	EG
X2	EP	M	EG	EG	EG	EG	EG	EG
	MP	EP	M	G	MG	MG	EG	EG
	P	EP	P	M	G	MG	MG	EG
	M	EP	MP	P	M	G	MG	EG
	G	EP	MP	MP	P	M	G	EG
	MG	EP	EP	MP	MP	P	M	EG
	EG	EP	EP	EP	EP	EP	EP	M

Figura 2.18.: BRs generadas por el método T ($ECM_{entr/prue} = 0,067518 / 0,032442$) —izquierda— y la metodología COR ($ECM_{entr/prue} = 0,097397 / 0,043984$) —derecha— en el problema F_2

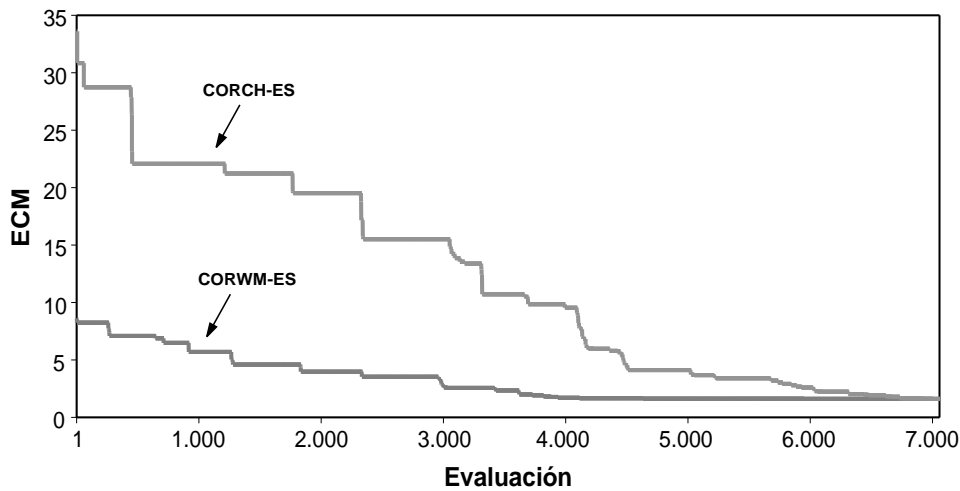


Figura 2.19.: Gráfica de evolución de los métodos CORWMM-ES y CORCH-ES en el problema F_1

cisión dependen de la aplicación considerada. En general, podemos observar que existe una relación entre el espacio de búsqueda generado por estos enfoques y la precisión obtenida. En la figura 2.17 mostramos el tamaño de los diferentes espacios de búsqueda según el enfoque seguido y el problema abordado.

CORWMM funciona mejor cuando el espacio de búsqueda es considerablemente grande (problema F_1) y la reducción que facilita dicho enfoque permite encontrar soluciones mejores. Por contra, si el espacio se restringe en exceso (problemas del arroz y línea de baja tensión), la gama de alternativas no ofrece soluciones muy precisas.

Por otro lado, gracias a trabajar en un espacio más confinado, el enfoque CORWMM obtiene soluciones aceptables con mayor rapidez. Este hecho se observa claramente cuando analizamos las gráficas de evolución, como mostramos en la figura 2.19 para los métodos CORWMM-ES y CORCH-ES en el problema F_1 . Efectivamente, podemos observar que el método CORCH-ES sólo obtiene una solución tan buena como la generada por el método CORWMM-ES al final de la ejecución del algoritmo.

En resumen, podemos decir que el enfoque CORWMM es preferible al CORCH cuando: el espacio de búsqueda es muy grande, el número de reglas en el primer caso es notablemente inferior o existen fuertes restricciones de tiempo. En el resto de situaciones, el enfoque CORCH parece funcionar mejor.

ES, BT, AGs o SH

Referente a la técnica empleada en la metodología COR, debemos comentar en primer lugar que la BT, si bien obtiene soluciones con rapidez, ofrece el peor comportamiento apreciándose una convergencia prematura que provoca caer en óptimos locales.

Entre las técnicas de AGs, ES y SH, comprobamos que los primeros obtienen soluciones sensiblemente mejores en términos de precisión. Sin embargo, el tiempo empleado para ello (definido por el valor EMS) es prohibitivo si consideramos uno de los principales objetivos que pretendíamos satisfacer con la metodología COR: la eficiencia. No obstante, la reducción del espacio de búsqueda ocasionado por nuestra metodología hace que los métodos CORWM-AG y CORCH-AG sean notablemente más rápidos que el método T, basado también en un AG.

Por otro lado, a partir de los resultados podemos indicar que la técnica de ES ofrece un mejor equilibrio entre precisión y tiempo requerido. Sin embargo, la técnica SH (métodos CORWM-SH y CORCH-SH) es sin duda la que mejor satisface los requisitos exigidos a la metodología, ya que obtiene soluciones con una buena precisión y de forma tremendamente rápida.

La figura 2.20 ilustra el comportamiento de las cuatro técnicas analizadas mediante las gráficas de evolución de los métodos CORCH-ES, CORCH-BT, CORCH-AG y CORCH-SH en una partición de datos concreta del problema del arroz. Mientras que el método CORCH-BT realiza un descenso gradual, los otros tres métodos muestran una fuerte bajada al comienzo, aunque el método CORCH-ES se estabiliza después de varias evaluaciones iniciales. Como podemos observar, el método CORCH-SH presenta el mejor comportamiento obteniendo buenas soluciones rápidamente.

La razón de la buena respuesta mostrada por la técnica SH está sin duda relacionada con el empleo de información heurística. Para comprobar este hecho, podemos analizar los resultados obtenidos por el método CORCH-SH con y sin información heurística (tabla 2.5). Se puede apreciar que al emplear dicho método sin considerar la información heurística disponible se presenta peor comportamiento en los siguientes términos:

- necesidad de un excesivo número de evaluaciones (en el problema F_1),
- obtención de peores resultados de precisión (en el problema F_2), o

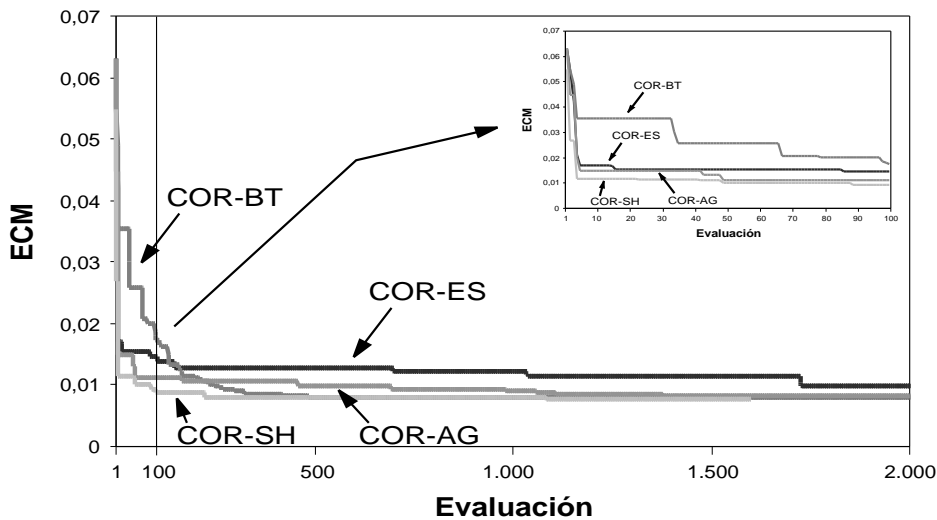


Figura 2.20.: Gráfica de evolución de los cuatro métodos que siguen el enfoque COR-CH en una partición del problema del arroz

- caída en óptimos locales (en los problemas del arroz y de la línea de baja tensión).

2.5. Extensiones a la Metodología COR

En la sección anterior hemos podido comprobar el buen funcionamiento de la metodología COR, así como que su combinación con la técnica de OCH es la que obtiene los mejores resultados, fundamentalmente en términos de eficiencia. No obstante, también ha quedado patente las deficiencias de la metodología en ciertas situaciones, esencialmente por el uso de determinados subespacios que degradan la calidad de la solución.

Por tanto, en esta sección nos centraremos en la combinación de COR con la técnica de OCH y propondremos tres extensiones concretas que hacen a la metodología más eficiente y eficaz:

- Empleo de un algoritmo de OCH más avanzado: el *sistema de colonias de hormigas* (SCH) [DG97].

Tabla 2.5.: Resultados obtenidos por el método CORCH-SH con y sin información heurística

Problema	Heurística	Iteraciones	ECM _{entr}	ECM _{prue}	EMS
F ₁	Sí	50	1,66062	1,41959	678
	No	50	10,84198	9,26706	1.873
	No	1.000	1,65561	1,46146	25.554
F ₂	Sí	50	0,09740	0,04398	1
	No	50	0,36866	0,25648	1.067
	No	1.000	0,11859	0,05465	16.027
Arroz	Sí	50	0,00694	0,00770	1.108
	No	50	0,00731	0,00817	1.372
	No	1.000	0,00719	0,00818	1.742
Eléctrico	Sí	50	178.119	158.662	359
	No	50	185.535	190.264	1.485
	No	1.000	182.291	183.457	2.102

- Aplicación de un proceso de *búsqueda local* al algoritmo de OCH.
- Adición del *consecuente candidato 'nulo'* en cada subespacio para eliminar reglas.

Las dos primeras extensiones estarán destinadas a mejorar el proceso de búsqueda para obtener soluciones de forma más rápida y eficaz gracias a realizar una exploración de las soluciones más optimizada. La tercera extensión será una variante de la metodología COR que permitirá eliminar aquellas reglas lingüísticas que perturban el buen comportamiento del modelo lingüístico obtenido, mejorando así su precisión e interpretabilidad.

Dado que dichas extensiones están destinadas a mejorar el proceso de búsqueda y eliminar reglas, en este caso es más recomendable emplear la *metodología con el enfoque guiado por rejilla difusa (CORCH)*, ya que de este modo se genera un espacio de búsqueda mayor y un número inicial de reglas más elevado. Por tanto, en esta sección nos limitaremos exclusivamente a dicho enfoque, aunque todas las propuestas planteadas también serían válidas par el enfoque guiado por ejemplos (CORWM).

2.5.1. Uso del Sistema de Colonias de Hormigas

En la experimentación realizada con el algoritmo SH hemos podido comprobar que, a pesar de su buen funcionamiento, en espacios de búsqueda grandes no se encuentran soluciones tan buenas como se desearía y que en determinados problemas el tiempo empleado es ligeramente alto. Este comportamiento puede optimizarse considerando otro algoritmo más avanzado dentro de la técnica de OCH.

En [DG97], los autores proponen un nuevo algoritmo de OCH, el *sistema de colonias de hormigas* (SCH), que mejora la búsqueda del SH acelerando la convergencia (*explotación*) al mismo tiempo que se favorece la diversidad (*exploración*). La explotación se mejora considerando una regla de actualización global donde sólo participa la *mejor solución* encontrada. Por otro lado, se favorece la exploración añadiendo una *regla de actualización local* que evita que las hormigas sigan caminos muy parecidos en la misma iteración, provocando así mayor diversidad. Además, se permite controlar mejor este equilibrio entre explotación y exploración con el uso de una *nueva regla de transición*. En el apéndice B.4 se detalla el funcionamiento del algoritmo. Emplearemos este algoritmo de OCH en la metodología COR con la intención de acelerar el tiempo necesario para diseñar la BR y encontrar mejores resultados.

Al aplicar este algoritmo de OCH en la metodología COR guiada por rejilla difusa, obtenemos un nuevo método: el *método CORCH-SCH*.

2.5.2. Uso de una Búsqueda Local dentro del Sistema de Colonias de Hormigas

Una de las formas más usuales de mejorar el rendimiento de los algoritmos de OCH es el uso de técnicas de búsqueda local [BHS99, MC99, SH00]. Esto se realiza usando algoritmos de búsqueda local para refinar las soluciones obtenidas después de una o varias iteraciones del algoritmo de OCH.

A pesar de que el uso de procedimientos de búsqueda local mejora la eficacia del algoritmo de OCH, se incrementa el número de evaluaciones en cada iteración y, por tanto, el tiempo de ejecución del método de aprendizaje, perdiendo así eficiencia. Además, debemos considerar que en nuestro caso, a diferencia de otras aplicaciones, el tiempo necesario para evaluar una solución vecina es mayor que el necesario para construir una nueva solución.

La búsqueda local se aplica generalmente a cada solución generada por cada hor-

Sean BLi y BLn dos valores previamente establecidos por el diseñador para definir respectivamente el número máximo de iteraciones y número de vecinos a crear en cada iteración. Hacer lo siguiente:

Sea $\mathbf{B}_{mejor} = \{B_{k^1}^1, \dots, B_{k^{s+}}^{s+}, \dots, B_{k^{Ns+}}^{Ns+}\}$ la solución correspondiente al mejor camino encontrado en la iteración actual del algoritmo de OCH.

Hacer $\mathbf{B}_{act} \leftarrow \mathbf{B}_{mejor}$.

Para $h = 1, \dots, BLi$ hacer:

Para $j = 1, \dots, BLn$ hacer:

- Obtener la solución \mathbf{B}'_j aplicando un *mecanismo de generación de vecino* a \mathbf{B}_{act} , $\mathbf{B}'_j \leftarrow N(\mathbf{B}_{act})$. Emplearemos el operador propuesto en la sección 2.3.1 para la técnica de ES: $\mathbf{B}'_j = \{B_{k^1}^1, \dots, B_{k^{s+}}^{s+}, \dots, B_{k^{Ns+}}^{Ns+}\}$.
- Si $j = 1$, hacer $\mathbf{B} \leftarrow \mathbf{B}'_1$. Si no, si \mathbf{B}'_j es mejor que \mathbf{B} , hacer $\mathbf{B} \leftarrow \mathbf{B}'_j$.
- Si \mathbf{B} es mejor que \mathbf{B}_{act} , hacer $\mathbf{B}_{act} \leftarrow \mathbf{B}$ y continuar. En otro caso, terminar el bucle.

Hacer que el mejor camino sea la solución optimizada, $\mathbf{B}_{mejor} \leftarrow \mathbf{B}_{act}$.

Figura 2.21.: Proceso de búsqueda local usado en el algoritmo SCH

miga. Sin embargo, dado que la metodología COR persigue realizar un aprendizaje rápido, en nuestro caso *el proceso de búsqueda local se aplicará solamente a la mejor solución generada en cada iteración*. Después de este proceso, se aplicará la regla de actualización global de feromona en la forma habitual. La búsqueda local propuesta consistirá en el algoritmo de ascensión de colinas descrito en la figura 2.21.

Al aplicar este proceso de búsqueda local al método extendido propuesto en la sección anterior, obtenemos un nuevo método: el *método CORCH-SCHBL*.

2.5.3. Extensión de la Metodología COR para Eliminar Reglas Lingüísticas

Uno de los problemas más usuales que surgen al definir una BR es el de emplear

- *reglas redundantes* cuyas acciones ya son cubiertas por otras,

- *reglas erróneas* que están mal definidas y perturban el rendimiento del sistema, o
- *reglas conflictivas* que distorsionan el comportamiento del sistema cuando coexisten con otras.

Además, el uso de un número alto de reglas entorpece la legibilidad del modelo.

Para afrontar estos problemas, se puede desarrollar un *proceso de reducción de reglas* combinando y/o seleccionando un subconjunto de ellas a partir de una BR dada, de forma que se reduzca el número de reglas mientras se mantiene —o incluso se mejora— la precisión del SBRD. Se han propuesto algunos métodos para buscar un conjunto optimizado de reglas, usualmente mediante AGs [CH97, CH00, HLV98a, INYT95, KKS00]. Por lo general, estas propuestas realizan la reducción con una fase de postprocesamiento una vez que la BR se ha obtenido.

En esta sección, proponemos una extensión de la metodología COR —a la que llamaremos *CORR*— para realizar el proceso de reducción al mismo tiempo que se realiza el aprendizaje con la intención de mejorar la precisión (la cooperación entre reglas puede mejorarse eliminando reglas erróneas y conflictivas) y la interpretabilidad (resulta más fácil comprender el funcionamiento de un modelo con un número menor de reglas) del modelo lingüístico aprendido.

Para hacer esto, podemos incluir un término adicional —el *consecuente nulo* (\mathcal{N})— al conjunto de consecuentes candidatos correspondiente a cada subespacio, es decir,

$$C^{s+} \leftarrow C^{s+} \cup \{\mathcal{N}\} \quad \text{y, por tanto,} \quad c_{s+} \leftarrow c_{s+} + 1.$$

De esta forma, si dicho consecuente se selecciona para un subespacio específico, esto implicará que la correspondiente regla lingüística no formará parte de la BR finalmente aprendida. La figura 2.22 ilustra gráficamente la forma de proceder de esta extensión en las técnicas de OCH aplicadas a la metodología COR.

Naturalmente, este ligero cambio en la metodología COR implica incrementar el espacio de búsqueda con los conocidos pros y contras que esto supone: se pueden obtener soluciones mejores pero crece la dificultad para encontrarlas. La recomendación de su uso dependerá de la naturaleza del problema.

En cuanto a la información heurística de los nuevos enlaces al consecuente nulo en cada subespacio, consideraremos el valor medio de la información heurística del

2. Metodología COR: Cooperación entre Reglas

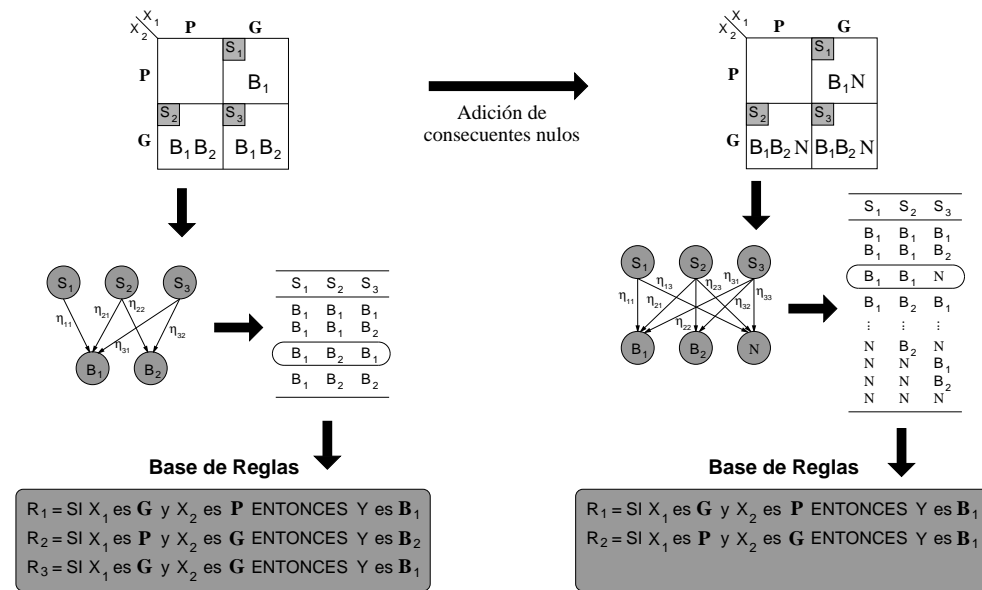


Figura 2.22.: Extensión permitiendo que la metodología COR elimine reglas lingüísticas mediante el uso de consecuentes nulos (\mathcal{N}) en cada subespacio. Dentro de los algoritmos de OCH, esto supone añadir al grafo un nuevo nodo consecuente con todos los subespacios conectados con él

resto de consecuentes candidatos usados en el correspondiente subespacio, es decir,

$$\eta_{s_+, c_{s_+}} = \frac{\sum_{j=1}^{c_{s_+}-1} \eta_{s_+, j}}{c_{s_+} - 1}.$$

Consideraremos, por tanto, tres nuevos métodos de aprendizaje que surgen de la combinación de la metodología extendida CORR con los tres algoritmos de OCH propuestos: el método *CORRCH-SH*, el método *CORRCH-SCH* y el método *CORRCH-SCHBL*.

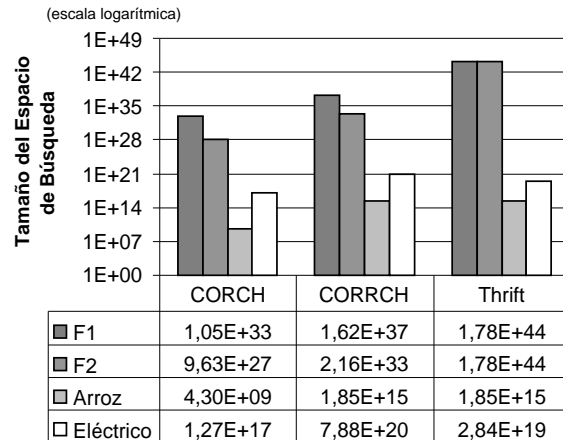


Figura 2.23.: Espacio de búsqueda manejado por CORCH, CORRCH y el método T

2.6. Estudio Experimental de las Extensiones a la Metodología COR

En esta sección realizaremos un nuevo estudio experimental considerando las extensiones a la metodología COR propuestas en la sección anterior. Emplearemos las mismas aplicaciones contempladas en el anterior estudio (sección 2.4). Para facilitar la comparación, incluiremos los resultados obtenidos por los métodos T y CORCH-SH.

La tabla 2.6 muestra los resultados obtenidos. En la tabla 2.7 indicamos los valores de parámetros empleados en los nuevos métodos propuestos.

Analizando los resultados obtenidos observamos que, en cuanto a la extensión consistente en considerar el algoritmo SCH para la búsqueda combinatoria, el método CORCH-SCH sólo obtiene mejoras representativas en términos de precisión frente al método CORCH-SH en el problema F_1 . En el resto de problemas, los resultados de precisión son similares, aunque el método CORCH-SH requiere un número menor de evaluaciones. Este comportamiento parece estar relacionado con el espacio de búsqueda asociado a cada uno de los problemas, tal como se indica en la figura 2.23. Por tanto, en vista de los resultados obtenidos, el uso del algoritmo SCH, una extensión del SH, sólo resulta de utilidad desde el punto de vista de la precisión cuando el problema a resolver implica un espacio de búsqueda muy grande.

Por otro lado, la inclusión del proceso de búsqueda local en el algoritmo SCH

Tabla 2.6.: Resultados obtenidos por las extensiones sobre la metodología COR y el resto de métodos de comparación en los cuatro problemas considerados

Método	F_1					F_2				
	#R	ECM _{entr}	ECM _{prue}	EMS	EMS	#R	ECM _{entr}	ECM _{prue}	EMS	EMS
T	49	1,609890	1,193721	27.301	27.301	47	0,067518	0,032442	29.572	29.572
CORCH-SH	49	1,660622	1,419587	678	678	49	0,097397	0,043984	1	1
CORCH-SCH	49	1,621177	1,207625	1.925	1.925	49	0,097397	0,043984	1	1
CORCH-SCHBL	49	1,596661	1,392013	1.390	1.390	49	0,097397	0,043984	1	1
CORRCH-SH	49	1,680405	1,502762	2.331	2.331	47	0,024919	0,032345	2.189	2.189
CORRCH-SCH	49	1,614917	1,339201	953	953	46	0,024919	0,032345	458	458
CORRCH-SCHBL	49	1,596661	1,392013	2.058	2.058	46	0,024919	0,032345	765	765

Método	Arroz					Línea de baja tensión				
	#R	ECM _{entr}	ECM _{prue}	EMS	EMS	#R	ECM _{entr}	ECM _{prue}	EMS	EMS
T	16	0,004949	0,005991	10.021	10.021	49	169.077	175.739	26.706	26.706
CORCH-SH	32	0,006943	0,007702	1.108	1.108	32	178.119	158.662	359	359
CORCH-SCH	32	0,006873	0,007815	1.212	1.212	32	179.029	161.618	780	780
CORCH-SCHBL	32	0,006898	0,007813	1.668	1.668	32	174.692	175.837	325	325
CORRCH-SH	17	0,005338	0,005642	1.229	1.229	26	182.134	159.222	359	359
CORRCH-SCH	17	0,005954	0,007792	691	691	27	181.662	169.672	1.464	1.464
CORRCH-SCHBL	19	0,005715	0,006438	1.027	1.027	29	173.823	160.753	417	417

Tabla 2.7.: Valores de los parámetros considerados en cada método para cada problema en la segunda experimentación

	F_1	F_2
Método	Valores de parámetros	Valores de parámetros
CORCH-SCH	$\eta = FVR_1, \rho = 0,6, \alpha = 2, \beta = 2, q_0 = 0,4$	$\eta = FVR_3, \rho = 0,0, \alpha = 1, \beta = 10, q_0 = 0$
CORCH-SCHBL	$\eta = FVR_3, \rho = 0,4, \alpha = 2, \beta = 2, q_0 = 0,2, BL_i = 10, BL_n = 49$	$\eta = FVR_3, \rho = 0,0, \alpha = 1, \beta = 10, q_0 = 0, BL_i = 0, BL_n = 0$
CORRCH-SH	$\eta = FVR_1, \rho = 0,2, \alpha = 1, \beta = 1$	$\eta = FVR_1, \rho = 0,4, \alpha = 1, \beta = 1$
CORRCH-SCH	$\eta = FVR_1, \rho = 0,6, \alpha = 1, \beta = 2, q_0 = 0,8$	$\eta = FVR_2, \rho = 0,2, \alpha = 1, \beta = 2, q_0 = 0,2$
CORRCH-SCHBL	$\eta = FVR_1, \rho = 0,8, \alpha = 2, \beta = 2, q_0 = 0,4, BL_i = 10, BL_n = 49$	$\eta = FVR_1, \rho = 0,2, \alpha = 1, \beta = 2, q_0 = 0,8, BL_i = 10, BL_n = 49$

	Arroz	Línea de baja tensión
Método	Valores de parámetros	Valores de parámetros
CORCH-SCH	$\eta = FVR_3, \rho = 0,2, \alpha = 1, \beta = 1, q_0 = 0,2$	$\eta = FVR_2, \rho = 0,6, \alpha = 2, \beta = 2, q_0 = 0,2$
CORCH-SCHBL	$\eta = FVR_3, \rho = 0,2, \alpha = 1, \beta = 1, q_0 = 0,2, BL_i = 10, BL_n = 32$	$\eta = FVR_2, \rho = 0,6, \alpha = 1, \beta = 1, q_0 = 0,2, BL_i = 10, BL_n = 32$
CORRCH-SH	$\eta = FVR_2, \rho = 0,2, \alpha = 1, \beta = 2$	$\eta = FVR_3, \rho = 0,6, \alpha = 1, \beta = 2$
CORRCH-SCH	$\eta = FVR_1, \rho = 0,2, \alpha = 1, \beta = 1, q_0 = 0,2$	$\eta = FVR_1, \rho = 0,6, \alpha = 1, \beta = 2, q_0 = 0,6$
CORRCH-SCHBL	$\eta = FVR_1, \rho = 0,2, \alpha = 1, \beta = 1, q_0 = 0,2, BL_i = 10, BL_n = 32$	$\eta = FVR_3, \rho = 0,6, \alpha = 1, \beta = 2, q_0 = 0,4, BL_i = 10, BL_n = 32$

(métodos CORCH-SCHBL y CORRCH-SCHBL) mejora de manera significativa la precisión de los modelos obtenidos en la mayoría de los casos. Además, en muchas situaciones se acelera el proceso de búsqueda con un número de evaluaciones menor. Otra ventaja interesante de emplear una optimización con búsqueda local es que hace al algoritmo más robusto frente a variaciones de los parámetros, como analizaremos en la sección 2.7.

En lo que respecta a la extensión CORR, que permite eliminar reglas lingüísticas que no cooperan adecuadamente, los resultados de precisión se mejoran notablemente (como puede observarse en los problemas F_2 , arroz y eléctrico). Por ejemplo, como se muestra en la figura 2.24 para el problema eléctrico y los métodos CORCH-SH y CORRCH-SCHBL, el hecho de eliminar tres reglas y variar dos consecuentes favo-

rece la acción cooperativa del conjunto de reglas lingüísticas y se mejora la precisión. Además, como ya se ha mencionado, los modelos obtenidos ganan en interpretabilidad.

No obstante, la metodología CORR no mejora la precisión en el problema F_1 comparado con la metodología COR. Esto se debe a que el número óptimo de reglas en este problema parece ser 49 debido a la buena distribución de ejemplos (véase el apéndice C.2.1), por lo que la capacidad de eliminar reglas incrementa el espacio de búsqueda innecesariamente.

Por otro lado, mientras que el método T y los basados en CORR obtienen un número de reglas similar en el problema del arroz, en el eléctrico CORR obtiene modelos precisos con un reducido número de reglas mientras que T emplea demasiadas reglas, lo que le conduce a sobreaprender ligeramente el problema. Esta diferencia entre los dos enfoques está relacionada con los espacios de búsqueda que manejan ambos métodos (como se indica en la figura 2.23) y con la facultad de los algoritmos de OCH de considerar adecuadamente la información heurística disponible.

2.7. Estudio de la Sensibilidad de los Parámetros en los Métodos de OCH Basados en COR y CORR

Hemos realizado un estudio sobre la sensibilidad de los algoritmos de OCH para analizar el efecto producido al alterar los valores de los parámetros y así poder recomendar los valores más adecuados para obtener resultados óptimos. La tabla 2.8 recoge las conclusiones deducidas para cada algoritmo y cada problema, donde las columnas *Sen.* muestran los parámetros que mayor influencia tienen en la precisión de los modelos obtenidos y las columnas *Recom.* indican los valores recomendados para un comportamiento óptimo (teniendo en cuenta tanto la precisión como la eficiencia).

Como se puede observar, los parámetros no tienen una influencia significativa en tres casos distintos:

- En primer lugar, el uso de la búsqueda local hace a los algoritmos notablemente más robustos.
- Además, en el problema eléctrico, el alto riesgo de caer en óptimos locales supone encontrar resultados similares a partir de diferentes valores de parámetros, menguando así su influencia.

2.7. Sensibilidad de los Parámetros en los Métodos de OCH Basados en COR y CORR

		X1						
		EP	MP	P	M	G	MG	EG
X2	EP	EP	EP	MP	MP			
	MP	MP	MP	MP	P	P		
	P	MP	P	P	M	M		
	M	MP	M	M	MG	MG		P
	G	P	M	M	EG	MG		P
	MG	MP	G	G	M			
	EG	M	M					

		X1						
		EP	MP	P	M	G	MG	EG
X2	EP	EP	EP	MP	MP			
	MP	MP	MP	MP	P	P		
	P	MP	P	P	M	M		
	M	MP	M		EG	MG		P
	G	P	M	M	EG	MG		P
	MG	MP	G	M	M			
	EG							

Figura 2.24.: BRs generadas por los métodos CORCH-SH ($ECM_{entr/prue} = 178.119 / 158.662$) —izquierda— y CORRCH-SCHBL ($ECM_{entr/prue} = 173.823 / 160.753$) —derecha— en el problema eléctrico

Tabla 2.8.: Sensibilidad a los parámetros y algunos valores recomendados para los diferentes métodos de OCH basados en COR y CORR

Método	F_1		F_2		Arroz		Baja tensión	
	Sen.	Recom.	Sen.	Recom.	Sen.	Recom.	Sen.	Recom.
COR-SH	η α	$\eta = FV/R_1$ $\alpha = 1$	—	$\eta = FV/R_3$	α	$\eta = FV/R_{2,3}$ $\alpha = 1$	—	$\eta = FV/R_3$
COR-SCH	η p α	$\eta = FV/R_1$ $p = 0,2$ $\alpha = 1$	—	$\eta = FV/R_3$	η p α	$\eta = FV/R_3$ $p = 0,2$ $\alpha = 1$	—	$\eta = FV/R_{2,3}$ $\alpha = 1$
COR-SCHBL	—	$\eta = FV/R_1$ $q_0 \leq 0,6$	—	$\eta = FV/R_3$	η q_0	$\eta = FV/R_3$ $q_0 = 0,8$	—	$\eta = FV/R_3$
CORR-SH	η α	$\eta = FV/R_1$ $\alpha = 1$	η α	$\eta = FV/R_1$ $\alpha = 1$	α	$\eta = FV/R_2$ $\alpha = 1$	η β	$\eta = FV/R_3$ $\beta = 2$
CORR-SCH	η p q_0	$\eta = FV/R_1$ $p = 0,8$ $q_0 = 0,8$	α p	$\alpha = 1$ $p = 0,2$	η p α	$\eta = FV/R_1$ $p = 0,2$ $\alpha = 1$	—	$q_0 \leq 0,6$
CORR-SCHBL	—	$\eta = FV/R_1$ $q_0 \leq 0,8$	—	$\alpha = 1$ $p = 0,2$	—	$\alpha = 1$ $p = 0,2$	—	$\eta = FV/R_{2,3}$ $q_0 = 0,6$

- Finalmente, la buena información heurística disponible en el problema F_2 para la metodología COR también implica que los parámetros no sean decisivos.

Por otro lado, en el resto de casos, podemos observar que generalmente la información heurística (η_{s+j}) y el factor de ponderación para el rastro de feromona (α) afectan significativamente a los resultados obtenidos. Resulta también interesante verificar que el algoritmo SCH es más sensible a cambios del parámetro de evaporación de feromona (ρ) que el SH.

Podemos destacar algunas reglas generales sobre los valores usados para cada parámetro:

1. En general, la información heurística con el mejor comportamiento en el algoritmo voraz (variantes del método CH) es la más recomendable. Sin embargo, cuando el problema tiene un alto riesgo de caer en óptimos locales, es preferible usar una heurística distinta de esta.
2. Hacer $\alpha = 1$ para dar menos importancia a la feromona y evitar que las buenas soluciones dominen en exceso.
3. Se recomienda un valor alto para el factor de evaporación de feromona ρ cuando se traten espacios de búsqueda muy grandes (mayor exploración).
4. Para los algoritmos SCH y SCHBL, se recomienda un valor alto de q_0 (mayor explotación).

2.8. Sumario

Hemos dedicado este capítulo a la propuesta de una nueva metodología de aprendizaje de reglas lingüísticas difusas con las siguientes características:

- Aprende de modo inductivo la representación simbólica y el número de reglas que componen el SBRD.
- Obtiene modelos con buena precisión favoreciendo la cooperación entre las reglas.
- Los modelos lingüísticos generados gozan de la máxima interpretabilidad.

2. Metodología COR: Cooperación entre Reglas

- La metodología es muy flexible y puede aplicarse con cualquier técnica de optimización combinatoria.
- El reducido espacio de búsqueda generado permite obtener buenas soluciones con rapidez.

Tabla 2.9.: Notación empleada en este capítulo

Notación	Significado
n	Número de variables de entrada
i	Índice de variables de entrada, ($1 \leq i \leq n$)
\mathcal{A}_i	Conjunto de términos lingüísticos de la i -ésima variable de entrada
\mathcal{B}	Conjunto de términos lingüísticos de la variable de salida
k	Índice de términos lingüísticos de la variable de salida, ($1 \leq k \leq \mathcal{B} $)
E	Conjunto de pares de datos entrada-salida (ejemplos)
N	Número de ejemplos, $N = E $
l	Índice de ejemplos, ($1 \leq l \leq N$)
x_i^l	Valor de la i -ésima variable de entrada del l -ésimo ejemplo
y^l	Valor de la variable de salida del l -ésimo ejemplo
e^l	Vector $(n+1)$ -dimensional que contiene el l -ésimo ejemplo, $e^l = (x_1^l, \dots, x_n^l, y^l)$
A_i^l	Conjunto difuso que mejor envuelve a x_i^l , $A_i^l \in \mathcal{A}_i$
B^l	Conjunto difuso que mejor envuelve a y^l , $B^l \in \mathcal{B}$
RC^l	Regla candidata obtenida a partir del l -ésimo ejemplo, $RC^l = \text{SI } X_1 \text{ es } A_1^l \text{ y } \dots \text{ y } X_n \text{ es } A_n^l \text{ ENTONCES } Y \text{ es } B^l$
N_S	Número de subespacios de entrada difusos multidimensionales, $N_S = \prod_{i=1}^n \mathcal{A}_i $
s	Índice de subespacios multidimensionales, ($1 \leq s \leq N_S$)
A_i^s	Término lingüístico usado en el subespacio S_s para la i -ésima variable de entrada, $A_i^s \in \mathcal{A}_i$
S_s	s -ésimo subespacio de entrada difuso n -dimensional, $S_s = (A_1^s, \dots, A_i^s, \dots, A_n^s)$
E'_s	Conjunto de ejemplos considerados en el subespacio S_s
e_{l^s}	l^s -ésimo ejemplo situado en el subespacio S_s , $e_{l^s} \in E'_s$
C^s	Conjunto de etiquetas lingüísticas entre los términos de la variable de salida (consecuentes candidatos) consideradas para los ejemplos incluidos en E'_s
c_s	Número de consecuentes candidatos en el subespacio S_s , $c_s = C^s $
k^s	Índice de consecuentes candidatos en el subespacio S_s , ($1 \leq k^s \leq c_s$)
$B_{k^s}^s$	k^s -ésimo consecuente candidato en el subespacio S_s , $B_{k^s}^s \in C^s$
$R_{k^s}^s$	Regla en el subespacio S_s formada con el consecuente $B_{k^s}^s$
s_+	Índice de subespacios de entrada difusos multidimensionales con ejemplos positivos, es decir, $E'_{s_+} \neq \emptyset$. La notación anterior puede restringirse a aquellos subespacios con ejemplos positivos empleando este índice
N_{S_+}	Número de subespacios de entrada difusos multidimensionales con ejemplos positivos
$ C $	Cardinalidad del conjunto C
$\mu_A(x)$	Grado de pertenencia del valor x a la etiqueta lingüística (conjunto difuso asociado) A
$VE_T(R, e)$	Valor de envoltura de la regla $R = \text{SI } X_1 \text{ es } A_1 \text{ y } \dots \text{ y } X_n \text{ es } A_n \text{ ENTONCES } Y \text{ es } B$ sobre el ejemplo $e = (x_1, \dots, x_n, y)$ usando al t -norma T , $VE_T(R, e) = T(\mu_{A_1}(x_1), \dots, \mu_{A_n}(x_n), \mu_B(y))$

3. Ajuste de las Estructuras Profundas de las Reglas Lingüísticas Difusas

Una palabra mal colocada estropea el más bello pensamiento.

VOLTAIRE (1694–1778), ESCRITOR Y FILÓSOFO FRANCÉS

En el capítulo 1 pudimos comprobar que el hecho de extender el proceso de modelado añadiendo grados de libertad adicionales para modificar la BD o la BR permitía obtener modelos lingüísticos con una gran precisión. Se demostró además que resultaba ser un enfoque tan eficaz que la combinación de un proceso de aprendizaje sencillo con un procedimiento posterior de ajuste superaba en términos de precisión a otros métodos de derivación más complejos.

Estos resultados tan contundentes nos motivan a presentar un método de ajuste que mejore aún más los grados de precisión alcanzados pero sin sacrificar en exceso la interpretabilidad de los modelos obtenidos. Para ello, como veremos a lo largo de este capítulo, realizaremos diferentes tipos de variaciones e incidiremos en dos niveles de especificidad distintos. Por otro lado, impondremos restricciones para evitar perder legibilidad en exceso.

El capítulo se organiza de la siguiente manera. En primer lugar, distinguiremos dos estructuras distintas que definen las reglas lingüísticas a dos niveles de especificidad diferentes. Posteriormente, analizaremos cómo pueden ajustarse convenientemente estas estructuras. Finalmente, introduciremos un método basado en AGs que realiza estas tareas de ajuste y validaremos su eficacia con un estudio experimental.

3.1. Estructuras Profundas de las Reglas Lingüísticas Difusas

Una regla lingüística difusa puede representarse a dos niveles de descripción diferentes definiendo dos estructuras distintas [Zad94]:

- *Estructura superficial* — Es una descripción menos específica que comprende la definición de la regla en su forma simbólica como una relación entre las variables lingüísticas de entrada y salida. La figura 3.1(a) muestra la estructura superficial de una regla lingüística. En un SBRD lingüístico, la estructura superficial de cada regla difusa viene codificada en la BR.
- *Estructura profunda* — Es una descripción más específica que se compone de la estructura superficial junto con el significado de las funciones de pertenencia asociadas a los términos lingüísticos implicados en la regla. La figura 3.1(b) ilustra la estructura profunda de una regla lingüística difusa. En un SBRD lingüístico, la estructura profunda está contenida en la BC (BR+BD).

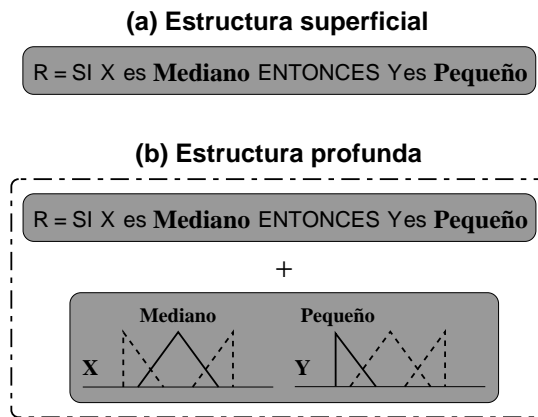


Figura 3.1.: Dos formas diferentes de definir una regla lingüística difusa: (a) *estructura superficial* = representación simbólica, (b) *estructura profunda* = representación simbólica + forma de las funciones de pertenencia

3.2. Ajuste de Estructuras Profundas en Sistemas Basados en Reglas Difusas

Como hemos comentado en los capítulos anteriores, uno de los problemas más importantes en las aplicaciones de la Lógica Difusa es la derivación automática de estas estructuras superficiales y profundas a partir de información numérica que represente el comportamiento de un sistema. Para realizar esta tarea, podemos distinguir entre dos enfoques distintos:

- *Proceso de aprendizaje* — Se refiere a la tarea de obtener directamente las estructuras superficiales [Thr91, WM92] o profundas [SR00] a partir de los datos disponibles.
- *Proceso de ajuste* — Se asume la existencia de una definición previa de las estructuras profundas —a partir de procesos de aprendizaje o información de expertos— y se ajustan dichas estructuras con *ligeras* modificaciones para incrementar el rendimiento del sistema (por ejemplo, [Jan93, Kar91]).

Algunas propuestas realizan un diseño de las estructuras de superficie [BKC96, CH97, KS95] con una búsqueda combinatoria para asignar términos lingüísticos en las representaciones simbólicas. Este enfoque, que en algunas ocasiones es incorrectamente considerado como ajuste de reglas difusas [BKC96], es en realidad un proceso de aprendizaje ya que las modificaciones a las estructuras superficiales se realizan como un medio para encontrar la solución final, y no como un fin.

Tradicionalmente, el proceso de ajuste se ha empleado para refinar las estructuras profundas cambiando exclusivamente los significados de las funciones de pertenencia [BKC96, CH97, HLV95, Jan93, Kar91, SKG93]. Algunos autores emplean expresiones de las funciones de pertenencia no lineales [CdJH98]. Por otro lado, recientes contribuciones realizan un ajuste de las estructuras superficiales alterando las representaciones simbólicas [CdJH98, GP99] con modificadores lingüísticos [Zad75].

No obstante, hasta ahora no se han considerado propuestas que combinen estos dos enfoques. En este capítulo nos proponemos introducir un método basado en AGs que realiza un ajuste *completo* de las estructuras profundas (incluyendo las de superficie) usando diferentes formas de cambiar el significado de los términos lingüísticos junto con una integración de modificadores lingüísticos en las reglas difusas.

En esta sección, introduciremos la forma de ajustar un SBRD redefiniendo sus estructuras profundas. Para realizar esto, en los dos subapartados siguientes explicaremos cómo ajustar las estructuras de superficie (representaciones simbólicas) usando modificadores lingüísticos y cómo adaptar la parte de las estructuras profundas que contienen las funciones de pertenencia. La acción combinada de ambos procesos realizará un ajuste completo de las estructuras profundas que definen el SBRD.

3.2.1. Ajuste de la Estructura Superficial usando Modificadores Lingüísticos

En la sección 1.4.2.1 del capítulo 1 vimos que la estructura de una regla lingüística podía hacerse más flexible incluyendo operadores que matizaran el significado de cada uno de los términos lingüísticos incluidos en ella (tales como *muy*, *más-o-menos*, *ligeramente*, *extremadamente*, etc). Estos operadores debían realizar una modificación semántica clara cuya acción pudiera ser fácilmente interpretable. Para ello, propusimos emplear modificadores lingüísticos.

Esta extensión al ML facilita de forma intuitiva procesos de ajuste en el sentido que hemos introducido en la sección anterior, es decir, como mecanismos que realizan una modificación ligera de estructuras previamente definidas. En efecto, esos matices que los modificadores lingüísticos aportan al significado de cada etiqueta no son más que un ajuste de las mismas. Además, podemos decir que dichas modificaciones afectan únicamente a la estructura superficial de la regla dado que estamos alterando su representación simbólica independientemente del significado global de los términos lingüísticos empleados. Por esta razón, la interpretabilidad de las reglas ajustadas se conserva en un algo grado. Algunas propuestas que realizan este tipo de ajuste con modificadores lingüísticos son [CdJH98, GP99].

Por ejemplo, a partir de la regla lingüística previamente definida:

SI X_1 es alto y X_2 es bueno **ENTONCES** Y es pequeño,

podemos obtener la siguiente regla lingüística ajustada:

SI X_1 es *muy* alto y X_2 es bueno **ENTONCES** Y es *más-o-menos* pequeño.

En realidad, este enfoque de ajuste no define un nuevo significado para lo que se conoce como *términos primarios* —alto, bueno y pequeño en nuestro ejemplo—

, sino que los usa como generadores cuyo significado se especifica en el contexto. En otras palabras, gracias a la *semántica de gramática atribuida* [Zad75] asociada a las variables lingüísticas, las funciones de pertenencia finales se calculan a partir del conocimiento de las funciones de pertenencia de los términos primarios.

Por supuesto, el hecho de emplear modificadores lingüísticos tendrá una influencia significativa en el comportamiento del SBRD ya que variará el grado de emparejamiento del antecedente de la regla así como el conjunto difuso de salida obtenido al aplicar la implicación en el proceso de inferencia.

En general, podemos distinguir tres tipos de modificadores lingüísticos:

- *Escalado no lineal* [Zim96] — Se realiza con funciones que operan sobre los grados de pertenencia de los términos lingüísticos elevándolos a la potencia de algún factor (véase la figura 3.2(a)). Por ejemplo, el operador ‘*mu*y’ eleva al cuadrado los grados de pertenencia del término lingüístico cuyo significado está modificando, es decir, $\mu_T^{mu}(x) = (\mu_T(x))^2$. Si se eleva a la potencia de un factor positivo menor que 1, el efecto será el de dilatar el conjunto difuso y, por el contrario, si el factor es superior a 1, el efecto será el de contraer el conjunto difuso.
- *Ensanchamiento/reducción* [MSG00] — Estos modificadores varían los conjuntos soporte y núcleo de los conjuntos difusos agrandándolos o reduciéndolos, pero intentando mantener un centro de gravedad similar al original (véase la figura 3.2(b)). El efecto que produce sobre los conjuntos difusos, aunque es similar al anterior (en términos de dilatación y contracción), resulta algo más intenso y se realiza con variaciones lineales.
- *Desplazamiento* [Lak73] — Este tercer tipo de modificadores varía las funciones de pertenencia desplazándolas a lo largo sus dominios (véase la figura 3.2(c)). Su efecto sobre los términos lingüísticos es más intenso que los anteriores.

También se han propuesto formas de combinar algunos de estos enfoques [Nov96].

3.2.2. Ajuste de la Estructura Profunda Con Variaciones Lineales y no Lineales

Ajustar la estructura profunda de una regla, además de alterar la estructura de superficie, implica adaptar la caracterización de las funciones de pertenencia asociadas

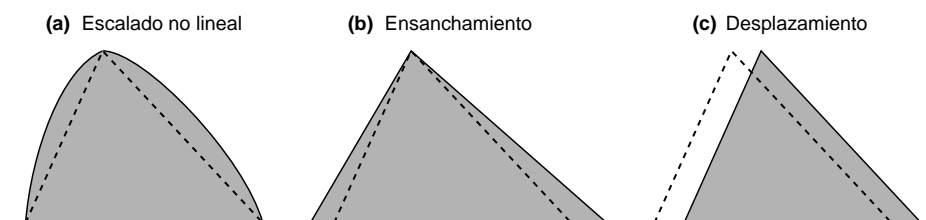


Figura 3.2.: Tipos de modificadores lingüísticos

a los términos lingüísticos primarios considerados en el sistema. Así, el significado de los términos lingüísticos se cambia a partir de una definición previa, una BD inicial en un SBRD.

Para cambiar el significado de las funciones de pertenencia —es decir, sus formas— debemos alterar los parámetros que la definen. Como vimos en la sección 1.4.1 del capítulo 1, podemos distinguir principalmente entre dos tipos de parámetros, lineales y no lineales, que determinan dos tipos diferentes de ajuste (véase la figura 1.4 en la página 25):

- Por un lado, podemos realizar *variaciones lineales* alterando los parámetros básicos que definen las funciones de pertenencia.
- Por otro lado, podemos realizar *variaciones no lineales* añadiendo un factor de escala no lineal a las funciones de pertenencia y alterando su grado.

El segundo enfoque (ajuste no lineal) tiene una importante limitación con respecto al primero (ajuste lineal): no altera el soporte y el núcleo de los conjuntos difusos. Además, cuando se consideran conjuntos difusos simétricos, su centro de gravedad no varía. En cambio, el grado de pertenencia de un valor al conjunto difuso se incrementa de una forma no lineal conforme se acerca al núcleo.

Por tanto, los dos enfoques producen efectos significativamente diferentes en el proceso de inferencia del SBRD y, dado que cada uno de ellos incide en un aspecto distinto del mismo, puede considerarse que sus acciones no son contradictorias sino complementarias.

Como ya analizamos en el capítulo 1, el proceso de ajuste de la semántica corre el riesgo de provocar algunos efectos no deseables como son la pérdida de interpretabilidad del modelo y el sobreaprendizaje del problema. Para resolver estos inconvenien-

tes, se suelen imponer restricciones para mantener una integridad semántica. Algunas de las propiedades que definen una semántica íntegra son las siguientes [Val99]:

- *Envoltura* — Cualquier elemento del universo de discurso debería pertenecer a al menos un término lingüístico. Alternativamente, se puede considerar un criterio más restrictivo estableciendo un grado de cubrimiento mínimo que debería exigirse a la semántica para cualquier valor del universo de discurso.

En el ajuste, esta propiedad se puede mantener empleando intervalos de variación que aseguren que dos funciones de pertenencia consecutivas estén solapadas en algún grado.

- *Normalidad* — Todo valor del universo de discurso de la variable debería tener un grado de pertenencia igual a 1 en alguna función de pertenencia. Esto es, todos los conjuntos difusos deberían ser normales.

En el ajuste, esta propiedad se puede mantener fácilmente forzando a que los puntos modales de las etiquetas de los extremos estén contenidos en el universo de discurso.

- *Distinguibilidad* — Cada término lingüístico debería tener un significado semántico y el conjunto difuso asociado debería definir claramente un rango del universo de discurso. En definitiva, las funciones de pertenencia deberían ser claramente diferentes.

Para mantener esta propiedad, se pueden establecer distancias mínimas entre los puntos modales de las funciones de pertenencia o evitar que el solapamiento entre los conjuntos difusos supere un umbral.

Es interesante destacar que, aunque generalmente los procesos de derivación de la BD consideran estas propiedades durante su diseño, en algunos casos se permite violar las restricciones y se emplea un procedimiento *a posteriori* para subsanar aquellas situaciones donde no se cumplan las propiedades contempladas [EV00].

Como se puede observar, si partimos de una semántica inicial íntegra, estas restricciones sólo pueden violarse cuando el ajuste realiza variaciones lineales, mientras que en el caso de las variaciones no lineales dichas propiedades siempre se cumplen. Por contra, en este segundo caso, puede aparecer otro tipo de pérdida de interpretabilidad relacionada con una excesiva deformación de las funciones de pertenencia. En

estos casos, se imponen restricciones tales como considerar un intervalo (por ejemplo, $[0, 5]$) o un conjunto (por ejemplo, $\{0, 5, 1, 2, 5\}$) de variación para el factor de escala que determina la alteración no lineal.

3.3. Un Método para el Ajuste Completo de las Estructuras Profundas

En esta sección vamos a proponer un proceso de ajuste basado en AGs (véase el apéndice B) para adaptar conjuntamente las funciones de pertenencia mediante variaciones lineales y no lineales y la estructura superficial de la regla difusa usando modificadores lingüísticos. El ajuste supone la existencia de una definición previa de la BC (BR+BD) que se puede obtener o bien a partir de un proceso de derivación o bien a partir de expertos.

En la figura 3.3 se muestran las distintas posibilidades de ajuste que permite el método propuesto. Consideraremos funciones de pertenencia triangulares. Realizaremos variaciones lineales alterando, dentro de los correspondientes intervalos de variación, los tres parámetros que las definen. Consideraremos también un factor de escala no lineal (dentro del intervalo $[0, 5]$) para cada función de pertenencia. Como modificadores lingüísticos en las estructuras de superficie, emplearemos los operadores ‘*mu*’ (que eleva la función de pertenencia al cuadrado) y ‘*más-o-menos*’ (que calcula la raíz cuadrada de la función de pertenencia).

El proceso consta de los siguientes componentes:

- Consideraremos un **AG generacional** con el procedimiento de muestro universal estocástico de Baker [Bak87], junto con un criterio elitista consistente en asegurar que la nueva población contenga la mejor solución encontrada en la generación anterior.
- El objetivo (**función de adaptación**) será minimizar la función ECM (definida en el capítulo anterior, sección 2.3.1).
- Emplearemos un **esquema de codificación** triple ($C_P + C_A + C_L$) para ajustar las funciones de pertenencia (C_P y C_A) —es decir, los significados de los términos lingüísticos en las estructuras profundas— y las representaciones simbólicas (parte C_L) —es decir, las estructuras superficiales—. C_P codificará los parámetros que definen la forma lineal de las funciones de pertenencia, C_A los

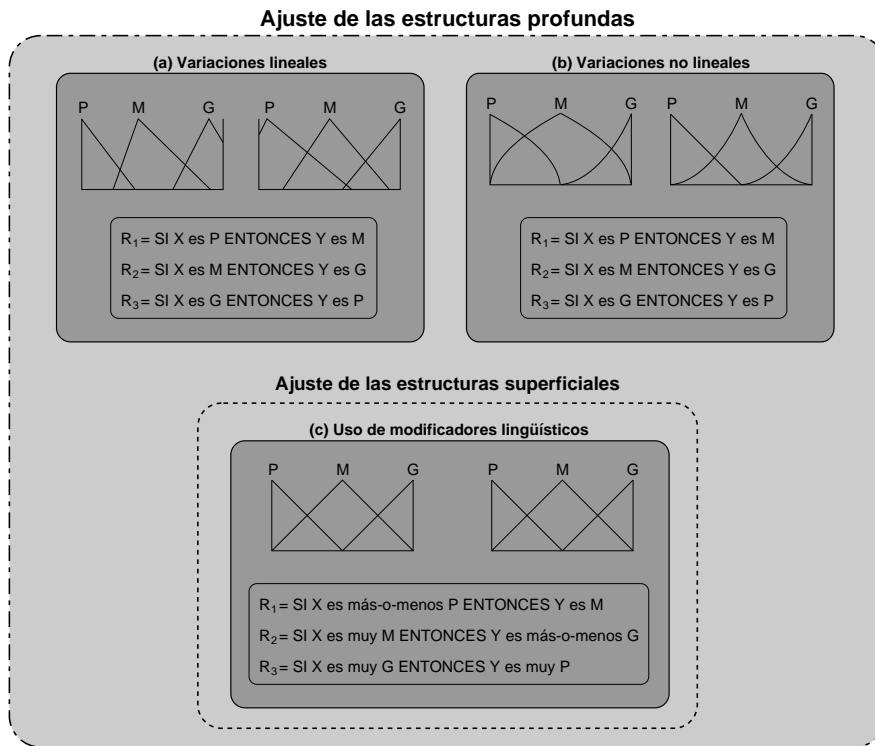


Figura 3.3.: Tipos de ajuste considerados en el método propuesto: (a) ajuste de las funciones de pertenencia de la BD con variaciones lineales, (b) ajuste de las funciones de pertenencia de la BD con variaciones no lineales y (c) ajuste de las representaciones simbólicas de las reglas difusas con modificadores lingüísticos

parámetros α que definen la forma no lineal y C_L los modificadores lingüísticos incluidos en las diferentes reglas. La figura 3.4 muestra este esquema gráficamente.

- Para la parte C_P , se usará una tupla de tres valores reales para cada función de pertenencia triangular, de modo que la BD se codifica en una subcadena del cromosoma construida uniendo las funciones de pertenencia empleadas en variable lingüísticas. Usaremos un intervalo de variación en cada gen de la parte C_P para mantener conjuntos difusos con un

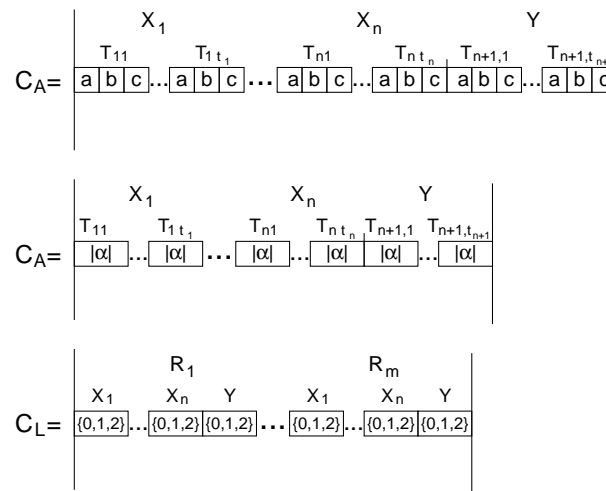


Figura 3.4.: Esquema de codificación para el ajuste de SBRDs, donde n es el número de variables de entrada, T_{ij} el j -ésimo término lingüístico de la i -ésima variable (considerando $n + 1$ como la variable de salida), t_i el número de términos lingüísticos de la i -ésima variable y m el número de reglas lingüísticas difusas

significado claro [CH97]. En la figura 3.5 se indica un ejemplo con los intervalos de variación considerados. Para calcularlos, se analizan los puntos medios entre parámetros homólogos de etiquetas consecutivas en la partición difusa inicial.

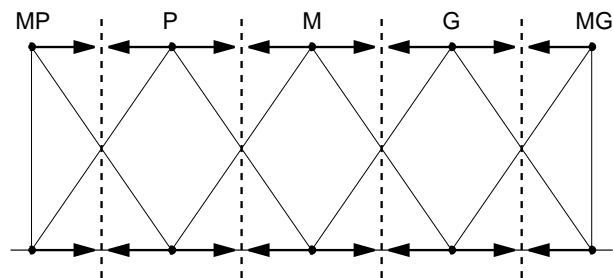


Figura 3.5.: Intervalos de variación considerados

- Para la *parte* C_A , se usará una subcadena del cromosoma con valores reales que codifican el valor del parámetro α para cada función de per-

tenencia. Cada gen toma cualquier valor en el intervalo $[-1, 1]$ con la siguiente correspondencia entre el alelo y el valor real:

$$\alpha = \begin{cases} 1 + c_{ij}^A, & \text{si } c_{ij}^A \in [-1, 0] \\ 1 + 4 \cdot c_{ij}^A, & \text{si } c_{ij}^A \in]0, 1] \end{cases},$$

siendo c_{ij}^A el gen asociado a la función de pertenencia del j -ésimo término lingüístico de la i -ésima variable.

- Para la *parte* C_L , el esquema de codificación genera cadenas de números enteros de longitud $m \cdot (n + 1)$ (donde m es el número de reglas y n el número de variables de entrada). Cada gen puede tomar cualquier valor del conjunto $\{0, 1, 2\}$ con la siguiente correspondencia con el modificador lingüístico empleado:

$$c_{ij}^L = 0 \longleftrightarrow \text{se usa el operador 'muy'},$$

$$c_{ij}^L = 1 \longleftrightarrow \text{no se usa ningún operador},$$

$$c_{ij}^L = 2 \longleftrightarrow \text{se usa el operador 'más-o-menos'}.$$

siendo c_{ij}^L el gen asociado al término lingüístico usado en la j -ésima variable de la i -ésima regla.

- Al generar la **población inicial**, se combinará parte de la información original contenida en la BC y BR iniciales (aprendida automáticamente o proporcionada por expertos) con valores aleatorios. Haciendo uso de este conocimiento facilitaremos la convergencia del AG.

Para considerar los valores originales en la parte C_P incluiremos directamente los valores reales, es decir, el valor de cada uno de los parámetros que definen cada una de las funciones de pertenencia contenidas en la BD inicial.

Para la parte C_A , los valores originales dependerán de si estos parámetros fueron considerados en la BC inicial o no. Si es así, los parámetros α se codificarán con el esquema comentado anteriormente; si no, se usará el alelo 0 (que fenotípicamente significa $\alpha = 1$).

Para la parte C_L , los modificadores usados en la BC inicial se codifican con el esquema mencionado anteriormente. Si no se consideraron modificadores lingüísticos en la BC inicial, se emplearán alelos 1.

Consideraremos los siguientes cuatro pasos para generar la población inicial:

1. Incluir un cromosoma que representa la BD y BR iniciales. Por tanto, los genes en las partes C_P , C_A y C_L codificarán directamente los valores correspondiente a la BC original.
 2. Generar un tercio de la población con la parte C_P aleatoria (dentro del intervalo de variación para cada gen) y con los alelos en las partes C_A y C_L codificando los valores originales.
 3. Generar otro tercio de la población con los valores originales en C_P , alelos aleatorios (dentro del intervalo $[-1, 1]$) en C_A y los valores originales en C_L .
 4. El resto de cromosomas se generan con los valores originales de la BD en los segmentos C_P y C_A , y alelos aleatorios (dentro del conjunto $\{0, 1, 2\}$) en C_L .
- El operador de **cruce** dependerá de la parte del cromosoma donde se aplique:
 - En los *segmentos* C_P y C_A , se considerará el operador de cruce max-min-aritmético [HLV98a]. En el apéndice B.3 se detalla su funcionamiento.
 - En el *segmento* C_L , se usará un cruce multipunto en dos puntos estándar.

Después de recombinar cada parte, se seleccionarán los dos mejores cromosomas entre los ocho (*cuatro* segmentos en C_P y C_A combinados con *dos* segmentos distintos en C_L) descendientes obtenidos para reemplazar a sus padres.

- El operador de **mutación** también dependerá de la parte del cromosoma donde se aplique:
 - En los *segmentos* C_P y C_A , se usará la mutación no uniforme de Michalewicz [Mic96] dentro del intervalo permitido para cada gen. En el apéndice B.3 se detalla su funcionamiento.
 - En el *segmento* C_L , el operador de mutación cambia el gen al alelo 1 cuando se deba mutar un gen con alelos 0 ó 2, y aleatoriamente a 0 ó 2 cuando se deba mutar un gen con alelo 1.

3.3. Un Método para el Ajuste Completo de las Estructuras Profundas

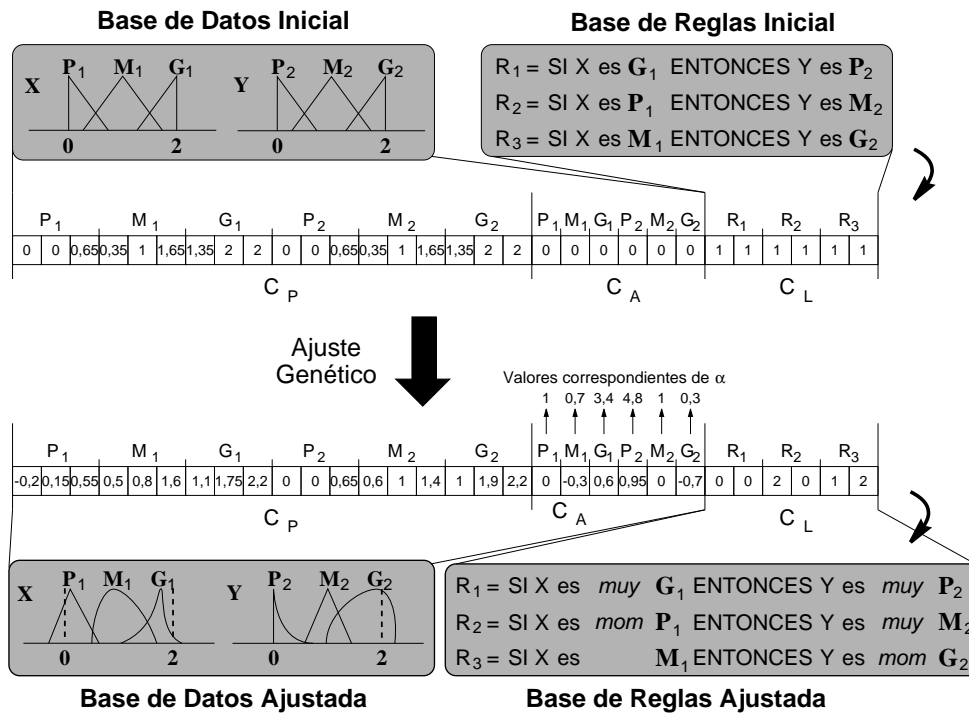


Figura 3.6.: Ejemplo de proceso de ajuste genético para un SBRD con una entrada y una salida, tres términos lingüísticos distintos para cada variable y tres reglas lingüísticas. P significa *pequeño*, N *normal*, G *grande* y *mom* ‘más-o-menos’

La figura 3.6 ilustra el proceso de ajuste genético propuesto. El AG se puede usar de diferentes formas dependiendo de los segmentos del cromosoma que se adapten, realizando así diferentes procesos de ajuste. Los más interesantes son un proceso de ajuste de la estructura profunda realizando variaciones lineales y no lineales en las funciones de pertenencia —segmentos C_P y C_A — (*método PA-Ajus*) y un proceso de ajuste de las estructuras profunda y de superficie con cambios lineales de las funciones de pertenencia y modificadores lingüísticos —partes C_P y C_L — (*método PL-Ajus*).

Otro proceso de ajuste interesante sería considerar las tres partes del cromosoma para ajustar las estructuras de superficie con modificadores lingüísticos y las estructuras profundas cambiando las funciones de pertenencia de forma lineal y no lineal

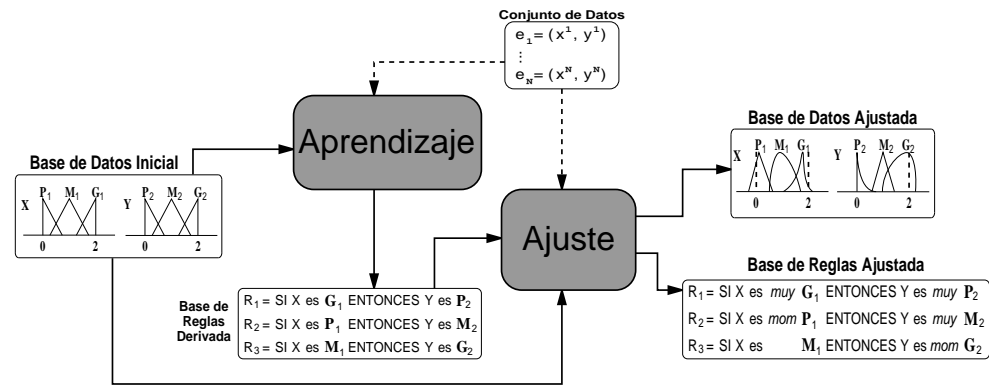


Figura 3.7.: El proceso de ajuste se realiza en dos etapas: en la primera, empleamos un método de aprendizaje para derivar la BR a partir de una BD inicial; después, el método de ajuste variará la BR obtenida previamente y/o la BD inicial

(*método PAL-Ajus*). Sin embargo, aunque este enfoque disfruta de la máxima flexibilidad (y potencial precisión), genera modelos lingüísticos menos interpretables. Además, una excesiva flexibilidad corre el riesgo de sobreajustar el problema, con la consecuente falta de predicción.

3.4. Estudio Experimental

El estudio experimental se centrará en aplicar diferentes procesos de ajuste a modelos previamente generados por distintos métodos de derivación. En nuestro caso, consideraremos dos métodos para aprender la BR inicial: el método WM y el método T. Ambos métodos se describieron en la sección 1.5.1.1 del capítulo 1. Por tanto, estos métodos actuarán como el módulo de aprendizaje mostrado en la figura 3.7, donde se indica gráficamente el modelo de derivación en dos etapas (primero aprendizaje y luego ajuste) que seguiremos en este estudio experimental.

Aplicaremos los métodos considerados a cuatro problemas: el modelado de las funciones F_1 y F_2 y la resolución de los problemas reales de la estimación de la calidad del arroz cocido y la estimación del costo de mantenimiento de líneas eléctricas de media tensión en zonas urbanas. En el apéndice C se puede consultar una descripción de estas aplicaciones.

Como BD inicial consideraremos una partición uniformemente distribuida para cada variable con funciones de pertenencia triangulares. La figura 2.1 (página 62) muestra un ejemplo de la partición considerada con cinco términos lingüísticos.

Para realizar un análisis riguroso, consideraremos todas las combinaciones posibles entre los tres enfoques de ajuste propuestos en este capítulo, es decir: ajuste de la estructura profunda con variaciones lineales de la semántica, con variaciones no lineales de la semántica o con modificadores lingüísticos en las estructuras superficiales de las reglas difusas. La tabla 3.1 resume los diferentes procesos de ajuste considerados. Como comentábamos en la sección anterior, los enfoques más interesantes parecen ser el PA-Ajus, PL-Ajus y PAL-Ajus.

Tabla 3.1.: Procesos de ajuste considerados en el estudio experimental

Método	Variaciones lineales de la semántica	Variaciones no lineales de la semántica	Variaciones de la estructura superficial con modif. ling.
P-Ajus	✓		
A-Ajus		✓	
L-Ajus			✓
PA-Ajus	✓	✓	
PL-Ajus	✓		✓
AL-Ajus		✓	✓
PAL-Ajus	✓	✓	✓

3.4.1. Tablas de Resultados y Valores de Parámetros

Las tablas 3.2 y 3.3 recogen los resultados obtenidos por los métodos de aprendizaje y ajuste analizados, siendo $\#R$ el número de reglas y ECM_{entr} y ECM_{prue} los valores del error cuadrático medio sobre los conjuntos de entrenamiento y prueba, respectivamente. Se indican en **negrita** los mejores resultados en cada problema y con cada método de aprendizaje.

Los valores de parámetros empleados en los AGs fueron los siguientes: un tamaño de población de 61 individuos, 2.000 generaciones para las funciones tridimensionales y el problema eléctrico y 1.000 para el arroz, 0,6 como probabilidad de cruce, 0,2 como probabilidad de mutación, 0,35 para el factor de ponderación en el cruce max-min-aritmético y 5 para el factor de ponderación en la mutación no uniforme.

Tabla 3.2.: Resultados obtenidos por los métodos de ajuste propuestos en las dos funciones tridimensionales

	F_1				F_2			
	WM (#R=49)		Thrift (#R=49)		WM (#R=49)		Thrift (#R=47)	
	2,048137	2,255929	1,596661	1,392013	0,194386	0,044467	0,067518	0,032442
Ajuste	ECM_{entr}	ECM_{prue}	ECM_{entr}	ECM_{prue}	ECM_{entr}	ECM_{prue}	ECM_{entr}	ECM_{prue}
P	0,281021	0,286958	0,585879	0,487147	0,037259	0,026799	0,026351	0,023980
A	0,922672	0,642659	1,493767	1,129385	0,091332	0,146879	0,055241	0,038520
L	0,632174	0,419765	1,155316	0,888639	0,071104	0,028436	0,029455	0,019749
PA	0,208114	0,231909	0,376250	0,395177	0,038014	0,080719	0,024852	0,023569
PL	0,255422	0,283536	0,345697	0,333120	0,019731	0,022026	0,013075	0,022166
AL	0,680960	0,498452	1,083085	0,827633	0,023271	0,023291	0,020639	0,018271
PAL	0,237614	0,250579	0,281956	0,281612	0,009473	0,012994	0,009257	0,018448

Tabla 3.3.: Resultados obtenidos por los métodos de ajuste propuestos en los dos problemas reales

	Arroz			Línea de media tensión				
	WM (#R=15)		Thrifty (#R=16)	WM (#R=66)		Thrifty (#R=534)		
	ECM _{entr}	ECM _{prue}	ECM _{entr}	ECM _{entr}	ECM _{prue}	ECM _{entr}		
Ajuste	0,013284	0,013119	0,004949	0,005991	71.294	80.934	34.063	42.116
P	0,001134	0,002339	0,000965	0,002712	16.066	22.653	12.032	16.530
A	0,002194	0,001731	0,001460	0,002525	40.331	47.073	21.566	24.130
L	0,001758	0,001731	0,000936	0,001916	22.882	26.013	6.560	6.679
PA	0,000990	0,002488	0,000841	0,002137	17.447	20.333	11.188	14.650
PL	0,000622	0,001840	0,000583	0,002322	6.946	13.509	3.979	5.191
AL	0,001611	0,001825	0,000669	0,002197	18.693	22.459	5.222	7.119
PAL	0,000575	0,001890	0,000573	0,002034	5.271	11.005	4.483	4.958

3.4.2. Análisis de los Resultados Obtenidos

Antes de todo, en vista de los resultados obtenidos, podemos apreciar que los procesos de ajuste mejoran significativamente —tanto en aproximación (ECM_{entr}) como en generalización (ECM_{prue})— los modelos derivados por los métodos de aprendizaje.

En términos generales, podemos decir que al realizar un ajuste completo de las estructuras profundas —es decir, incluyendo las de superficie— (como hacen los métodos PL-Ajus, AL-Ajus y PAL-Ajus), se obtienen modelos difusos significativamente más precisos. Es particularmente notable los excelentes resultados de precisión generados por el método PAL-Ajus que combina efectos de ajuste macroscópicos y microscópicos [BKC96] con dos formas de cambiar las funciones de pertenencia.

Por otro lado, centrándonos en los métodos P-Ajus y A-Ajus, el primero obtiene claramente modelos más precisos. Sin embargo, en problemas como el arroz, donde existe un alto riesgo de sobreajustar el conjunto de datos, el método P-Ajus obtiene un grado de predicción peor debido a su mayor capacidad de aproximación. En efecto, el hecho de variar el conjunto soporte de los conjuntos difusos asociados a los términos lingüísticos en el método P-Ajus (lo que no es posible en el método A-Ajus) permite al modelo difuso obtenido aproximarse mejor al conjunto de entrenamiento.

El método L-Ajus resuelve este inconveniente en cierto grado. Aunque, en cualquier caso, este método realiza modificaciones en las funciones de pertenencia similares al método A-Ajus, tiene mejor capacidad de aproximación gracias a ajustar las estructuras superficiales. Así, el método L-Ajus supera al A-Ajus y muestra un mejor equilibrio entre aproximación y generalización que el método P-Ajus.

Cuando se combinan dos enfoques de ajuste, los resultados de precisión se mejoran notablemente. Podemos observar que el proceso PA-Ajus consigue un grado de aproximación incluso mejor que el método P-Ajus sin perder su capacidad de predicción gracias a integrar variaciones no lineales de las funciones de pertenencia. Además, se sigue manteniendo un buen grado de interperabilidad, como lo demuestra la figura 3.8 para el método WM+PA-Ajus en el problema F_1 .

Por otro lado, como hemos comentado, cuando se realizan procesos de ajuste a dos niveles de especificidad, no sólo se mejora de precisión notablemente sino que los modelos lingüísticos obtenidos siguen manteniendo un buen grado de interpretabilidad, como puede observarse por ejemplo para los métodos T+PL-Ajus (figura 3.9) y T+AL-Ajus (figura 3.10 y tabla 3.4) en el problema F_2 y arroz, respectivamente.

Finalmente, en cuanto al comportamiento del método PAL-Ajus, que agrupa los

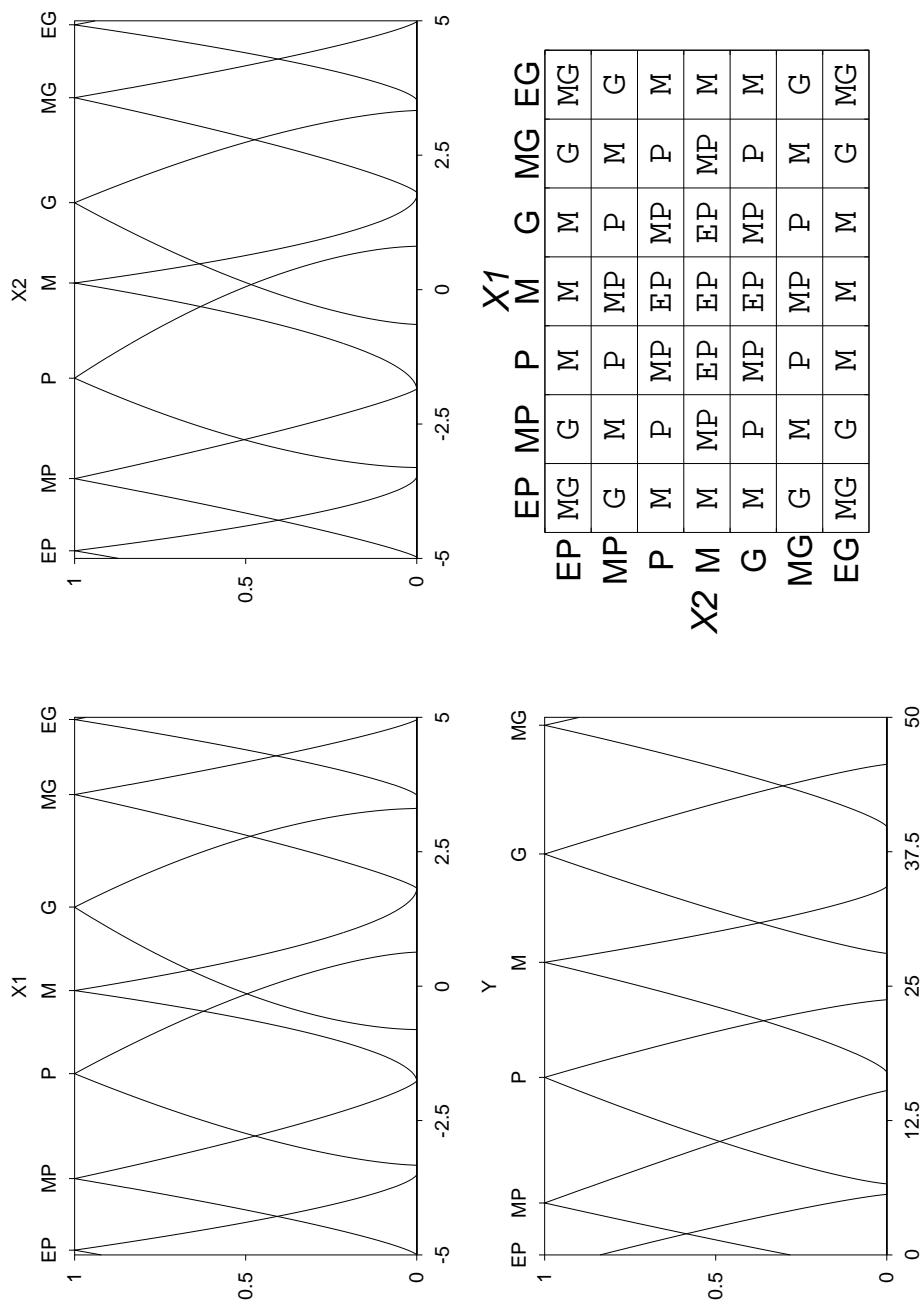


Figura 3.8.: Estructuras profundas generadas por el método WM+PA-Ajus para el problema F₁. #R=49, $ECM_{entr/prue}$ antes del ajuste = 2,048137 / 2,255929, $ECM_{entr/prue}$ después del ajuste = 0,208114 / 0,231909

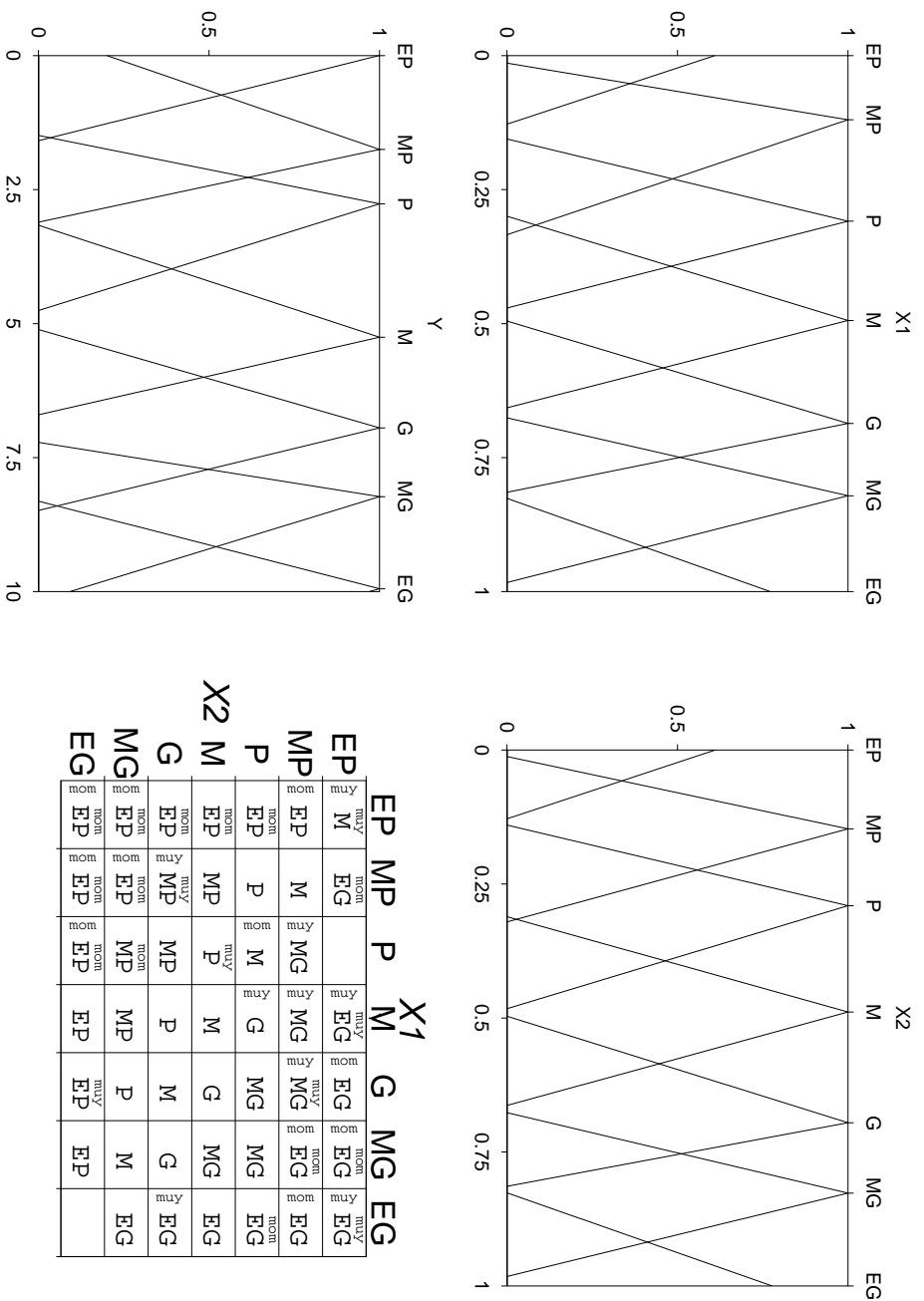


Figura 3.9.: Estructuras profundas generadas por el método T+PL-Ajus para el problema F₂. #R=47, ECM_{entr/prue} antes del ajuste = 0,067518 / 0,032442, ECM_{entr/prue} después del ajuste = 0,013075 / 0,022166

tres tipos de ajuste considerados, podemos apreciar que en la mayoría de los casos obtiene los mejores grados de aproximación y generalización entre todos los métodos considerados. En efecto, gracias a realizar procesos de ajuste de diferente naturaleza (variaciones lineales y no lineales) y especificidad (estructuras superficiales y profundas), se evita sobreaprender el problema en exceso obteniendo así unos grados de generalización muy buenos.

Además, debido a las restricciones impuestas en el proceso de ajuste de las estructuras profundas (intervalos de variación para los enfoques lineal y no lineal) y superficiales (uso de sólo dos modificadores lingüísticos), la interpretabilidad de los modelos obtenidos sigue siendo buena. La figura 3.11 y la tabla 3.5 muestran el modelo lingüístico generado por el método WM+PAL-Ajus en el problema eléctrico. Podemos observar que se mantiene una legibilidad adecuada. Aunque el efecto de las variaciones no lineales en este modelo parece imperceptible en términos de interpretabilidad, sí favorece notablemente la precisión. En efecto, si eliminamos las variaciones no lineales (hacemos $\alpha = 1$) en la BD generada (figura 3.11), obtenemos una precisión de $ECM_{entr/prue} = 13.646 / 21.529$, que resulta notablemente peor.

3.5. Sumario

En este capítulo hemos presentado un método de ajuste de modelos lingüísticos previamente definidos que ha demostrado obtener unos grados de precisión muy buenos conservando en gran medida la interpretabilidad. Esto se consigue gracias a considerar las siguientes características para cada aspecto:

- *Precisión* — Para ganar en precisión, se ha realizado un ajuste a dos niveles de especificidad diferentes, obteniendo efectos macroscópicos y microscópicos. Por otro lado, se han combinado variaciones lineales con no lineales, lo cual mejora la precisión al incidir en diferentes aspectos en el proceso de inferencia.
- *Interpretabilidad* — Se han considerado intervalos de variación adecuados que limitan la libertad del ajuste, consiguiendo que los modelos ajustados sigan manteniendo una buena interpretabilidad. Por otro lado, hemos empleado modificadores lingüísticos para aportar un significado claro al efecto obtenido. El conjunto de modificadores está igualmente limitado a sólo dos elementos.

3. Ajuste de las Estructuras Profundas de las Reglas Lingüísticas Difusas

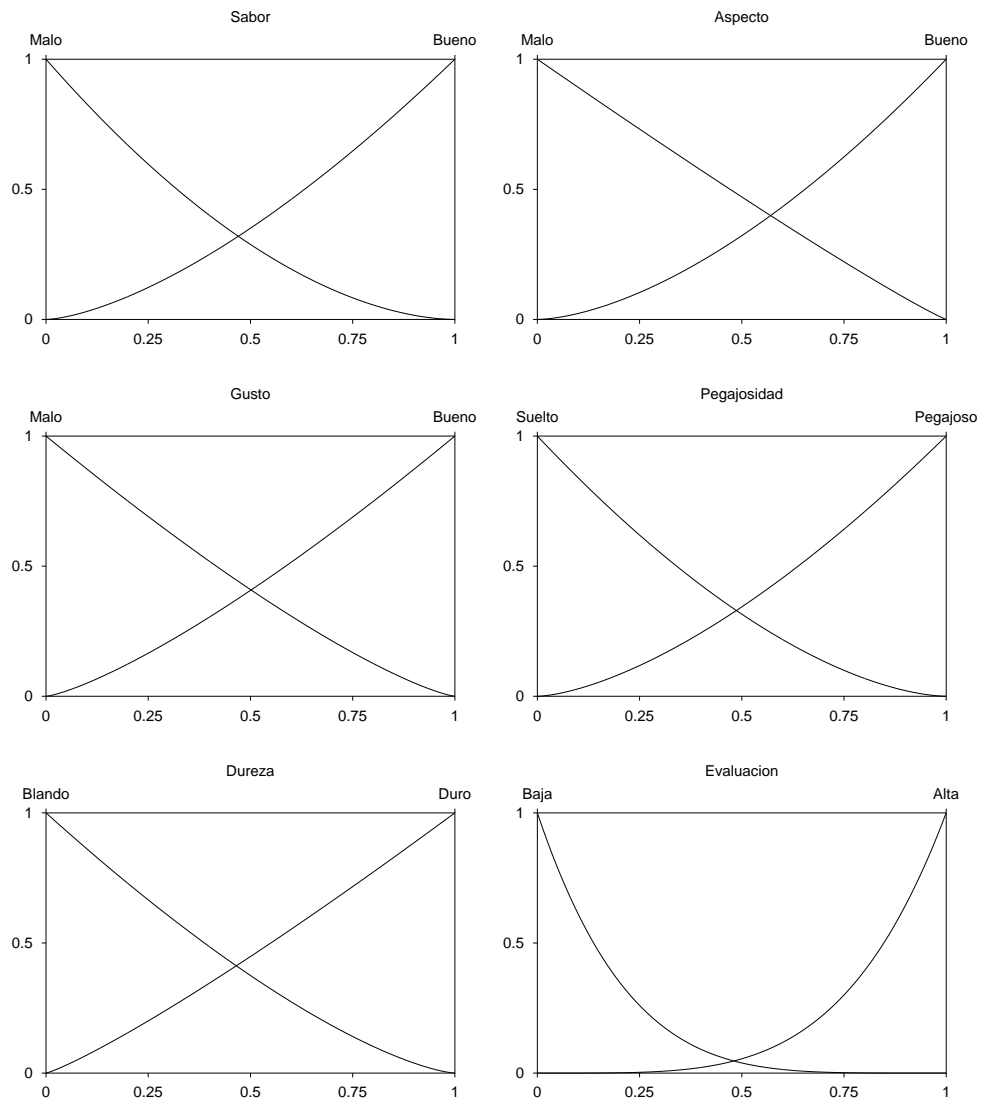


Figura 3.10.: BD (semántica de las estructuras profundas de las reglas difusas) generada por el método T+AL-Ajus para una partición específica del conjunto de datos del problema del arroz. #R=15, $ECM_{entr/prue}$ antes del ajuste = 0,005130 / 0,004110, $ECM_{entr/prue}$ después del ajuste = 0,000686 / 0,001297

Regla	Sabor	Aspecto	Gusto	Pegajosidad	Dureza	Evaluación
R ₁	Malo	muy Malo	muy Malo	Suelto	Blando	muy Baja
R ₂	Malo	muy Malo	muy Malo	Suelto	Duro	muy Baja
R ₃	Malo	muy Malo	Bueno	Suelto	Blando	Alta
R ₄	muy Malo	mom Bueno	muy Malo	Suelto	Blando	muy Baja
R ₅	mom Malo	mom Bueno	muy Malo	Suelto	Duro	Baja
R ₆	mom Malo	muy Bueno	Bueno	Pegajoso	Blando	Alta
R ₇	Bueno	Malo	Malo	Suelto	Blando	Baja
R ₈	Bueno	Malo	Malo	Suelto	Duro	Baja
R ₉	mom Bueno	Malo	Malo	Pegajoso	Blando	mom Baja
R ₁₀	muy Bueno	mom Malo	muy Bueno	Pegajoso	Blando	Alta
R ₁₁	Bueno	mom Malo	mom Bueno	Pegajoso	Duro	Alta
R ₁₂	muy Bueno	muy Bueno	muy Malo	Pegajoso	Duro	Alta
R ₁₃	Bueno	Bueno	Bueno	Suelto	Blando	Alta
R ₁₄	muy Bueno	mom Bueno	muy Bueno	Pegajoso	Blando	Alta
R ₁₅	muy Bueno	Bueno	muy Bueno	Pegajoso	Duro	mom Alta

Tabla 3.4.: BR (estructuras superficiales de las reglas difusas) generada por el método T+AL-Ajus para una partición específica del conjunto de datos del problema del arroz. #R=15, $ECM_{entr/prue}$ antes del ajuste = 0,005130 / 0,004110, $ECM_{entr/prue}$ después del ajuste = 0,000686 / 0,001297

3. Ajuste de las Estructuras Profundas de las Reglas Lingüísticas Difusas

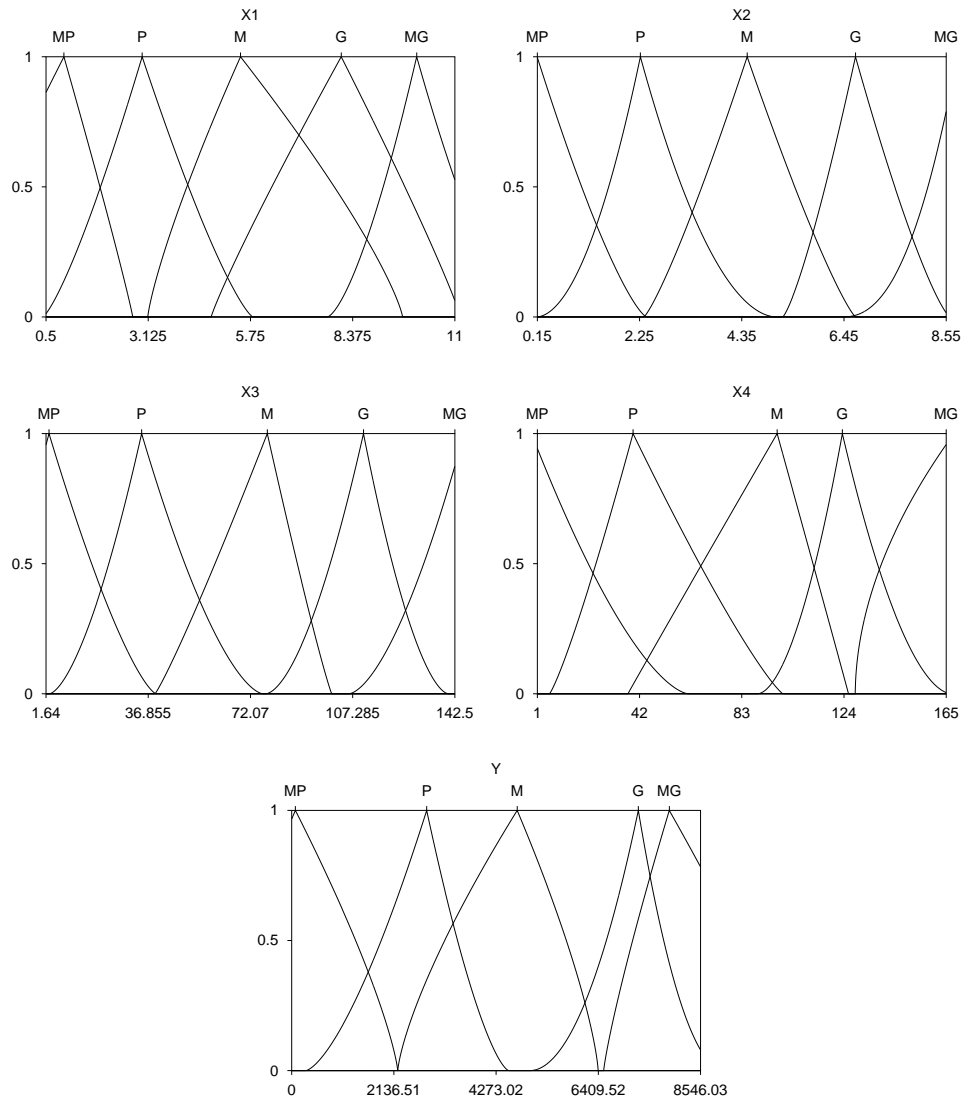


Figura 3.11.: BD (semántica de las estructuras profundas de las reglas difusas) generada por el método WM+PAL-Ajus para el problema de la línea de media tensión. #R=66, $ECM_{entr/prue}$ antes del ajuste = 71.294 / 80.934, $ECM_{entr/prue}$ después del ajuste = 5.271 / 11.005

Reg.	X ₁	X ₂	X ₃	X ₄	Y	Reg.	X ₁	X ₂	X ₃	X ₄	Y
R ₁	MP	mom	MP	MP	MP	R ₃₄	G	muy	P	MP	P
R ₂	MP	mom	MP	P	muy	R ₃₅	G	mom	muy	mom	mom
R ₃	mom	MP	muy	mom	MP	R ₃₆	muy	muy	P	M	M
R ₄	MP	muy	mom	MP	MP	R ₃₇	muy	muy	P	mom	mom
R ₅	MP	muy	muy	P	muy	R ₃₈	G	M	mom	MP	P
R ₆	mom	mom	MP	MP	MP	R ₃₉	G	mom	P	M	M
R ₇	P	mom	muy	mom	mom	R ₄₀	G	mom	P	M	M
R ₈	muy	muy	MP	MP	mom	R ₄₁	G	mom	P	G	M
R ₉	muy	MP	muy	MP	MP	R ₄₂	muy	muy	M	M	M
R ₁₀	mom	MP	mom	P	mom	R ₄₃	mom	muy	M	M	G
R ₁₁	mom	P	mom	MP	muy	R ₄₄	G	muy	M	MP	G
R ₁₂	mom	muy	mom	muy	muy	R ₄₅	G	G	M	M	M
R ₁₃	mom	mom	P	MP	MP	R ₄₆	G	G	M	M	G
R ₁₄	mom	mom	muy	P	P	R ₄₇	G	G	M	M	G
R ₁₅	muy	mom	mom	muy	muy	R ₄₈	G	mom	M	G	mom
R ₁₆	mom	muy	muy	MP	muy	R ₄₉	mom	G	G	M	G
R ₁₇	muy	M	M	M	MP	R ₅₀	muy	G	G	M	mom
R ₁₈	mom	mom	M	muy	P	R ₅₁	G	G	G	mom	mom
R ₁₉	mom	P	mom	MP	MP	R ₅₂	mom	muy	MG	P	M
R ₂₀	M	P	mom	MP	P	R ₅₃	mom	muy	MG	M	G
R ₂₁	mom	muy	mom	M	mom	R ₅₄	mom	mom	G	G	mom
R ₂₂	muy	muy	muy	MP	mom	R ₅₅	mom	MG	MG	mom	MG
R ₂₃	muy	mom	muy	P	P	R ₅₆	muy	MG	MG	M	MG
R ₂₄	muy	muy	muy	mom	M	R ₅₇	muy	MG	MG	G	muy
R ₂₅	M	muy	muy	MP	MP	R ₅₈	muy	G	P	P	mom
R ₂₆	mom	M	M	P	P	R ₅₉	MG	muy	P	M	M
R ₂₇	mom	muy	M	mom	mom	R ₆₀	muy	MG	P	G	muy
R ₂₈	M	M	M	M	M	R ₆₁	mom	MG	M	M	M
R ₂₉	M	M	M	M	M	R ₆₂	muy	MG	muy	muy	mom
R ₃₀	mom	muy	G	M	MP	R ₆₃	mom	MG	mom	MG	mom
R ₃₁	M	mom	M	mom	M	R ₆₄	muy	MG	G	P	mom
R ₃₂	muy	mom	G	muy	P	R ₆₅	MG	muy	M	G	muy
R ₃₃	muy	muy	G	muy	M	R ₆₆	muy	MG	M	MG	MG

Tabla 3.5.: BR (estructuras superficiales de las reglas difusas) generada por el método WM+PAL-Ajus para el problema de la línea de media tensión. #R=66, $ECM_{entr/prue}$ antes del ajuste = 71.294 / 80.934, $ECM_{entr/prue}$ después del ajuste = 5.271 / 11.005

4. Hibridación de Diferentes Mejoras al Modelado Lingüístico

¡Señores, no es cuestión de romper estructuras, sino de saber qué hacer con los pedazos!

QUINO (1932–), HUMORISTA Y DIBUJANTE ARGENTINO

A lo largo de esta memoria hemos estudiado diferentes posibilidades de extender el ML para ganar en precisión manteniendo un buen grado de comprensibilidad. Concretamente, en los dos capítulos anteriores hemos planteado cuatro mecanismos para mejorar el ML haciéndolo más flexible desde distintos puntos de vista:

- *Mejora C* — Aprendizaje de reglas induciendo cooperación con la metodología COR.
- *Extensión P* — Aprendizaje de los parámetros que definen las funciones de pertenencia de la BD.
- *Extensión A* — Aprendizaje de los factores de escala no lineales (α) de las funciones de pertenencia de la BD.
- *Extensión L* — Aprendizaje de los modificadores lingüísticos empleados en cada variable lingüística para cada regla.

Cabe señalar que, si bien las diferentes extensiones consideradas individualmente presentan ciertas deficiencias en algún contexto particular, su adecuada combinación podría resultar en una poderosa herramienta de modelado. De hecho, en el capítulo anterior hemos podido comprobar cómo la combinación de las extensiones P, A y L

en su forma más simple mejoraba notablemente el comportamiento del proceso de modelado.

En un sentido más general, las técnicas de *Lógica Difusa* y *Computación Evolutiva* empleadas en esta memoria forman parte del área conocida como *Computación Flexible* (en inglés, *Soft Computing*) o, en otros sectores, Inteligencia Computacional. Según Goldberg [Gol00], estas técnicas son individualmente muy poderosas pero «*el futuro de nuestro campo está en la integración de lo mejor de cada uno de sus exponentes*».

A pesar de que hoy en día es posible encontrar combinaciones tales como sistemas neuro-difusos, neuro-genéticos, difuso-genéticos... la cuidadosa integración de estos métodos en sistemas inteligentes autónomos requiere más que simples combinaciones [Gol99]. En efecto, la integración del poder de abstracción de los Sistemas Difusos y el poder innovador de los AGs [Gol98, Gol00] precisan una sofisticación de diseño que va más allá de la simplicidad de «*poner todo junto*», siendo necesario un mayor sentido de la compacidad de las hibridaciones.

Esto nos anima a realizar un estudio más profundo de la hibridación de las distintas extensiones al ML para explotar en mejor medida las interdependencias existentes. Al abordar este estudio, arrancamos con dos interrogantes:

- Por un lado, cabe plantearse la conveniencia de dividir el proceso en diferentes tareas secuenciales o, por contra, realizarlo todo simultáneamente.
- Por otro lado, podemos afrontar cada tarea con los métodos tradicionales de la Computación Evolutiva o bien considerar nuevos enfoques que estén mejor preparados para tratar la hibridación de componentes. Por ejemplo, podríamos considerar la coevolución cooperativa (CC) [PJ00].

En las secciones siguientes analizaremos los procesos de derivación secuencial o simultánea y los algoritmos coevolutivos cooperativos como herramientas para realizar una hibridación efectiva. Posteriormente, propondremos cinco modos distintos de fusionar convenientemente las cuatro mejoras del ML consideradas. Finalizaremos el capítulo con un estudio experimental y algunas conclusiones.

4.1. Derivación Secuencial o Simultánea

Para agrupar los procesos de derivación de los distintos componentes podemos distinguir, en general, entre dos enfoques:

- *Derivación secuencial* — El diseño se divide en subprocesos (etapas) encadenados secuencialmente. Entre las etapas más comunes se encuentran el aprendizaje del número de términos lingüísticos [CHV01, Glo96, IM96], aprendizaje de la BR [CH97, KKK94, SR00], la reducción de reglas [CH97], la compactación de reglas [SR00], el ajuste de las funciones de pertenencia [BKC96, CH97, HLV95, Kar91, KKK94] y la fusión de funciones de pertenencia [EV00].
- *Derivación simultánea* — El modelo lingüístico se diseña en un único proceso. Para ello, se suelen fundir varios procedimientos de derivación como, por ejemplo, aprendizaje de BR y BD [KS95, LM94, MMH97, PS01, WHT98] o reducción de reglas y ajuste de las funciones de pertenencia [GJ99].

La derivación secuencial presenta la principal ventaja de reducir el espacio de búsqueda al trabajar con espacios más confinados en cada una de las fases. Para escindir adecuadamente los componentes a diseñar es recomendable que entre ellos exista la menor relación posible. Así, una de las formas más comunes de proceder se compone de dos etapas: en primer lugar, emplear un proceso de aprendizaje de la BR que considere fija una definición preliminar de la BD y, posteriormente, realizar un ajuste de la BD para optimizar el rendimiento del modelo [CH97, KKK94, SR00].

Por otro lado, como apuntan algunos autores [LM94], la derivación secuencial corre el riesgo de generar soluciones subóptimas, dado que se optimiza cada componente por separado. Por contra, cuando se aprende el modelo completo en un único paso, se considera adecuadamente la fuerte dependencia existente entre los diferentes componentes. No obstante, la complejidad para realizar el proceso se incrementa ya que el espacio de búsqueda que debe manejarse es notablemente superior.

Como vemos, ambos enfoques tienen pros y contras que deben ser cuidadosamente analizados. En este capítulo propondremos métodos que combinen las cuatro mejoras analizadas siguiendo estas dos líneas de derivación.

4.2. Algoritmos Coevolutivos Cooperativos

Dentro de la teoría de la Computación Evolutiva [Mic96] (AGs, estrategias de evolución, etc.), ha surgido recientemente un nuevo paradigma, los algoritmos coevolutivos [Par95]. Estos algoritmos están recibiendo un interés creciente gracias a su habilidad para desenvolverse en problemas descomponibles con espacios de búsqueda muy grandes.

Los algoritmos coevolutivos se componen de dos o más especies (poblaciones) que interactúan permanentemente entre ellas mediante una función de adaptación conjunta. De ese modo, a pesar de que cada especie tiene su propio esquema de codificación y operadores de reproducción, a la hora de evaluar un individuo, su bondad se calcula considerando algunos individuos de las otras especies. Esta coevolución hace más fácil encontrar buenas soluciones en problemas complejos.

Se pueden considerar diferentes tipos de interacciones entre las especies según las dependencias existentes entre los subcomponentes de la solución. En general, podemos mencionar dos tipos distintos de interacción:

- *Algoritmos coevolutivos competitivos* [RB97] — Aquellos donde cada especie compite contra el resto (por ejemplo, obtener exclusividad de un recurso limitado). En este caso, el incremento de la adecuación de un individuo de una especie implica una disminución de la adecuación en el resto de las especies, es decir, el éxito ajeno supone el fracaso personal.
- *Algoritmos coevolutivos cooperativos o simbióticos* [PJ00] — Aquellos donde todas las especies cooperan para construir una solución al problema. En ese caso, la adecuación de un individuo dependerá de su capacidad para colaborar con individuos de otras especies.

La CC es recomendable cuando el problema a resolver tiene las siguientes características [PS01]:

1. el espacio de búsqueda es complejo,
2. el problema, por definición, se puede descomponer,
3. se manejan diferentes tipos de valores y conceptos, y
4. hay una fuerte interdependencia entre los componentes de la solución.

En los algoritmos cooperativos coevolutivos, la evolución de las distintas especies se realiza según el esquema mostrado en la figura 4.1. En cada generación, y para cada población, se selecciona un conjunto de cooperadores. Para evaluar un individuo en una determinada especie, se combina su información con los cooperadores del resto de especies, formando así una solución al problema. El valor de adaptación para ese individuo será el resultado de agregar la evaluación de cada una de las soluciones generadas.

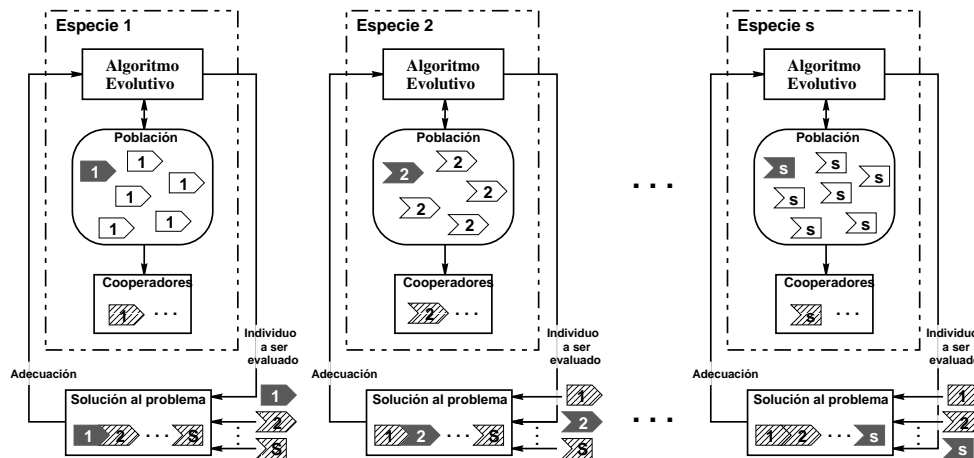


Figura 4.1.: Coevolución cooperativa

Por tanto, este tipo de interacción es ideal para nuestros propósitos, ya que podemos definir las extensiones al ML analizadas como subcomponentes del problema. De hecho, Peña-Reyes y Sipper ya han propuesto un método de derivación con esta filosofía [PS01] en el que dos especies evolucionan ajustando la BD y aprendiendo la BR al mismo tiempo.

4.3. Hibridaciones Propuestas

En esta sección nos proponemos estudiar diferentes formas de hibridar las cuatro mejoras al ML propuestas en los capítulos anteriores: *mejora C* (aprendizaje de reglas con COR), *extensión P* (aprendizaje de los parámetros de las funciones de pertenencia de la BD), *extensión A* (aprendizaje de los factores de escala no lineales α de las funciones de pertenencia de la BD) y *extensión L* (aprendizaje de los modificadores lingüísticos empleados en cada variable lingüística para cada regla). Para ello, realizaremos distintas combinaciones considerando básicamente los dos aspectos mencionados: realizar una derivación secuencial o simultánea y emplear AGs o CC. Entre todas las combinaciones posibles destacaremos sólo las que parecen ser más sensatas.

Así, un buen enfoque para realizar la derivación secuencial sería efectuar primero el proceso de aprendizaje de la BR (lo cual supone un efecto macroscópico) y

posteriormente ajustar tanto esta BR como la BD inicial (que implica efectos microscópicos). En cuanto a la CC, atenderemos a dos criterios para decidir cómo dividir los subcomponentes en dos grupos: por un lado distinguiremos entre efectos macroscópicos (C) y microscópicos (PAL) y, por otro, entre aprendizaje de la BR (CL) y de la BD (PA).

De esta forma, consideraremos los siguientes cinco métodos como representación de las diferentes hibridaciones posibles (la figura 4.2 recoge gráficamente estas combinaciones):

1. *Método C+PAL* — Derivación secuencial con AGs. Se compone de un proceso de aprendizaje de las estructuras superficiales (C) y un ajuste posterior de las estructuras profundas (PAL) con un AG.
2. *Método C+PA-L* — Derivación secuencial con CC. Se compone de un proceso de aprendizaje de las estructuras superficiales (C) y un ajuste posterior de las estructuras profundas con CC, una especie para la semántica de las estructuras profundas (PA) y otra para las estructuras superficiales (L).
3. *Método CPAL* — Derivación simultánea con AGs. Se trata de un proceso de aprendizaje de las estructuras profundas incluyendo en un único cromosoma los cuatro mecanismos de flexibilización considerados.
4. *Método C-PAL* — Derivación simultánea con CC. Es un proceso de aprendizaje de las estructuras profundas, una especie para aprendizaje de estructuras superficiales (C) y otra para ajuste de estructuras profundas (PAL).
5. *Método CL-PA* — Derivación simultánea con CC. En este proceso de aprendizaje de las estructuras profundas una especie se emplea para las estructuras superficiales (CL) y la otra para la semántica de las estructuras profundas (PA).

Dado que el método C+PAL no es más que la combinación secuencial de la metodología COR descrita en el capítulo 2 con el proceso de ajuste introducido en el capítulo 3, no será necesario describir de nuevo el comportamiento de este método. Introduciremos los aspectos más relevantes del resto de métodos en los siguientes subapartados.

En la sección 2.3.3 del capítulo 2 (página 79) y en la sección 3.3 del capítulo 3 (página 120) se puede encontrar una descripción más detallada de las representaciones y operadores de recombinación mencionados a continuación.

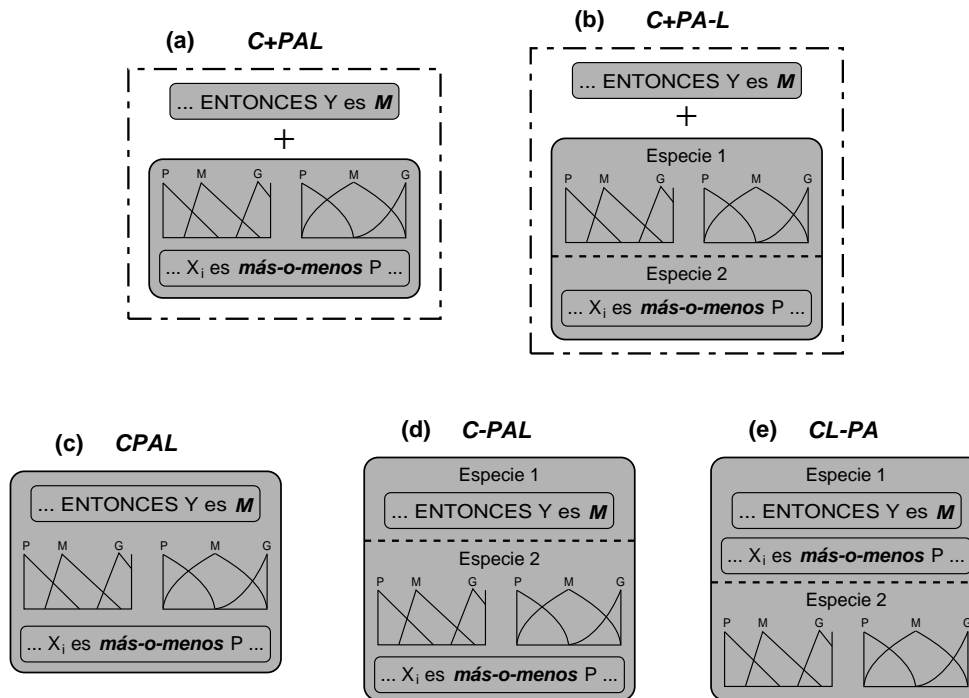


Figura 4.2.: Hibridaciones analizadas

4.3.1. Método CPAL

El método CPAL integra en un AG los cuatro mecanismos de extensión del ML analizados. El esquema de **codificación** se compone de cuatro partes que codifican respectivamente los consecuentes de las reglas en representación entera (C), los parámetros de las funciones de pertenencia en representación real (P), los factores de escala no lineal de las funciones de pertenencia en representación real (A) y los modificadores lingüísticos de las reglas en representación entera (L).

Se seguirá un esquema de AG **generacional**. Consideraremos el procedimiento de **selección** por muestreo universal estocástico de Baker (con probabilidades de selección según el esquema de orden lineal) junto con un mecanismo elitista consistente en sustituir el peor individuo de la población actual por el mejor de la población anterior en el caso de que este último no haya sobrevivido tras los procesos de cruce y mutación.

La **población inicial** contendrá un cromosoma con los valores originales y cuatro

conjuntos de individuos distintos, cada uno de ellos con cromosomas que contienen valores originales en una de las partes y valores aleatorios en las otras tres.

Para realizar el proceso de **cruce**, se aplicará el operador multipunto en dos puntos en las partes C y L. Esto se realiza escogiendo una subcadena para cada parte e intercambiando la información contenida en los padres, obteniendo así dos descendientes. Por otro lado, a las partes P y A se le aplica el operador de cruce max-min-aritmético obteniendo cuatro descendientes. Se seleccionarán los dos mejores individuos entre las ocho combinaciones posibles.

En el proceso de **mutación**, una vez establecido que un cromosoma ha de ser mutado según una determinada probabilidad, el operador selecciona un gen de cada parte y lo altera convenientemente. En el caso de las partes C, P y A, la mutación consiste en sustituir el alelo actual por otro distinto seleccionado aleatoriamente en el correspondiente dominio (un conjunto de valores en la parte C y un intervalo real en las partes P y A). En la parte L, la mutación altera el gen al alelo más cercano de forma aleatoria. Es decir, si los valores posibles son $\{0,1,2\}$ y el gen a mutar contiene el alelo 0 ó 2, el nuevo valor será 1; si por el contrario contiene el valor 1, el nuevo alelo será 0 ó 2 escogido aleatoriamente.

4.3.2. Métodos Coevolutivos: PA-L, C-PAL y CL-PA

Como hemos comentado anteriormente, el problema de derivar una BC apropiada para un SBRD se puede descomponer en varias tareas. Nuestros algoritmos coevolutivos se compondrán de dos especies que cooperarán para construir la solución completa, conteniendo cada una de ellas algunos de los mecanismos de flexibilización contemplados en este capítulo. En los dos subapartados siguientes introduciremos el esquema de interacción cooperativa considerada entre las dos especies (común a todos los métodos) y los operadores evolutivos usados en cada método.

4.3.2.1. Esquema de Interacción Cooperativa Considerado

Sea F_{ij} el SBRD obtenido mediante la composición de los subcomponentes codificados en los cromosomas i y j de la especie 1 y 2, respectivamente. El objetivo será minimizar el error cuadrático medio:

$$ECM_{ij} = \frac{1}{2 \cdot N} \sum_{l=1}^N \left(F_{ij}(x^l) - y^l \right)^2,$$

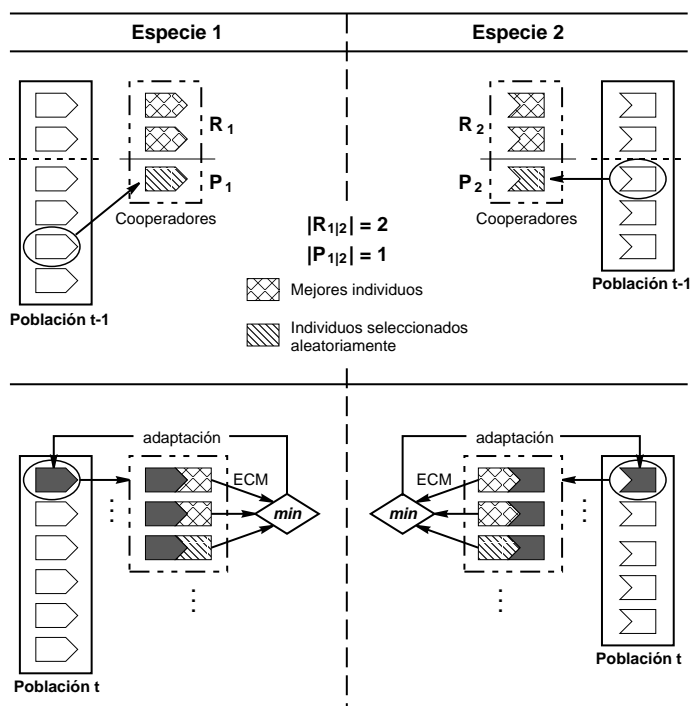


Figura 4.3.: Esquema de interacción considerado en el método de aprendizaje cooperativo coevolutivo

donde $F_{ij}(x^l)$ es la salida obtenida a partir del SBRD diseñado.

Cada individuo de las especies 1 ó 2 se evalúa con la correspondiente **función de adaptación** f_1 o f_2 , que se definen de la siguiente forma:

$$f_1(i) = \min_{j \in R_2 \cup P_2} ECM_{ij},$$

$$f_2(j) = \min_{i \in R_1 \cup P_1} ECM_{ij},$$

siendo i y j los individuos de las especies 1 y 2 respectivamente, R_1 y R_2 conjuntos formados por los individuos con el mayor grado de adaptación en la generación previa de las especies 1 ó 2 respectivamente, y P_1 y P_2 conjuntos de individuos seleccionados aleatoriamente de la población previa de las especies 1 ó 2 respectivamente. La figura 4.3 ilustra gráficamente el esquema de interacción propuesto.

Mientras que los conjuntos $R_{1|2}$ permiten a los mejores individuos influir en el

proceso guiando la búsqueda a través de buenas soluciones, los conjuntos $P_{1|2}$ introducen diversidad en la búsqueda. El uso combinado de ambos tipos de conjuntos brinda al algoritmo un buen equilibrio entre explotación ($R_{1|2}$) y exploración ($P_{1|2}$). El diseñador será quien defina previamente la cardinalidad de los conjuntos $R_{1|2}$ y $P_{1|2}$.

4.3.2.2. Operadores Evolutivos

Se seguirá un esquema de AG **generacional** en ambas especies. Consideraremos el procedimiento de selección por muestreo universal estocástico de Baker junto con un mecanismo elitista que asegura la selección del mejor individuo de la generación anterior. El resto de aspectos a considerar en cada especie dependerá del método de derivación empleado:

- *Método PA-L:*
 - *Especie 1: Ajuste de la semántica de las estructuras profundas (PA)* — Se emplea un esquema de codificación con representación real compuesto por dos subcadenas, una de ellas codifica los parámetros que definen cada función de pertenencia y la otra el factor de escala no lineal correspondiente (α). Se considera el cruce max-min-aritmético y la mutación aleatoria. El operador de mutación se aplica a las dos partes del cromosoma por separado, es decir, se mutan dos genes. La población inicial contiene un cromosoma con los valores originales de los parámetros y todos los factores de escala establecidos al valor 1. La mitad de la población se genera con valores de los parámetros de las funciones de pertenencia aleatoriamente y valores 1 para los factores de escala. Finalmente, el resto de la población se genera con los valores originales para los parámetros de las funciones de pertenencia y valores aleatorios para los factores de escala.
 - *Especie 2: Ajuste de las estructuras superficiales (L)* — Se trata de un esquema de codificación con representación multivaluada (tres valores) que define el modificador lingüístico empleado en cada variable para cada regla. Los valores posibles son usar el modificador ‘muy’, usar el modificador ‘más-o-menos’ o no usar ningún modificador. Se emplea el cruce multipunto en dos puntos. La mutación altera aleatoriamente el valor del gen al alelo más cercano. La población inicial se crea con un

cromosoma que no contiene ningún modificador y con el resto de individuos generados aleatoriamente.

- *Método C-PAL:*

- *Especie 1: Aprendizaje de las estructuras superficiales (C)* — El esquema de codificación de tipo entero representa el consecuente a emplear en cada subespacio de entrada difuso considerado, se permite el empleo de un consecuente adicional que indica la eliminación de la correspondiente regla lingüística. Se considera el cruce multipunto en dos puntos y el operador de mutación aleatorio uniforme. La población inicial consta de un cromosoma que contiene el mejor consecuente para cada subespacio según el algoritmo voraz empleado (métodos WM y CH en la metodología COR) y el resto de individuos generados aleatoriamente.
- *Especie 2: Aprendizaje de las estructuras profundas (PAL)* — Se considera un esquema de codificación compuesto por tres subcadenas para codificar respectivamente los parámetros de las funciones de pertenencia (representación real), los factores de escala no lineal de las funciones de pertenencia (representación real) y los modificadores lingüísticos de las reglas (representación entera). En las dos primeras partes se emplea el operador de cruce max-min-aritmético y la mutación no uniforme. En la tercera parte se considera el cruce en dos puntos y la mutación al alelo más cercano. Para generar los dos descendientes en cada cruce de cromosomas, se obtienen los dos mejores individuos entre las ocho combinaciones posibles (cuatro del cruce max-min-aritmético con dos del cruce multipunto). El operador de mutación altera un gen de cada parte del cromosoma. La población inicial contiene un cromosoma con los valores originales y tres conjuntos de individuos distintos, cada uno de ellos con cromosomas que contienen valores originales en una de las subcadenas y valores aleatorios en las otras dos.

- *Método CL-PA:*

- *Especie 1: Aprendizaje de las estructuras superficiales (CL)* — La representación de cromosomas consta de dos subcadenas de tipo entero, la

primera contiene los consecuentes a emplear en los diferentes subespacios de entrada difusos (incluyendo el consecuente nulo) y la segunda los modificadores lingüísticos a emplear en cada variable para cada regla. Se considera el operador de cruce en dos puntos y la mutación al alelo más cercano. Ambos operadores se aplican a las dos partes del cromosoma por separado, es decir, se intercambia una secuencia de cada subcadena en el cruce y se altera un gen de cada subcadena en la mutación. La población inicial contiene un cromosoma con los mejores consecuentes según el algoritmo voraz y sin emplear modificadores lingüísticos; una mitad de la población tendrá consecuentes aleatorios sin modificadores lingüísticos en las reglas; y, por último, el resto de la población contendrá los mejores consecuentes según el algoritmo voraz y modificadores lingüísticos generados aleatoriamente.

- *Especie 2: Aprendizaje de la semántica de las estructuras profundas (PA)*
— Similar a la especie 1 del método PA-L descrito con anterioridad.

4.4. Estudio Experimental

En este estudio experimental analizaremos el comportamiento de las hibridaciones propuestas. Emplearemos cuatro aplicaciones (véase el apéndice C): las funciones F_1 y F_2 , el problema del arroz y el de la estimación de costo de mantenimiento en líneas eléctricas de media tensión.

4.4.1. Tabla de Resultados y Valores de Parámetros

La tabla 4.1 recoge los resultados obtenidos por los métodos analizados, siendo $\#R$ el número de reglas y ECM_{entr} y ECM_{prue} los valores del error cuadrático medio sobre los conjuntos de entrenamiento y prueba, respectivamente. Se indican en **negrita** los mejores resultados en cada problema.

A continuación se detallan los valores de parámetros y enfoques de la metodología COR que se han considerado en cada método para cada problema:

- *Método C+PAL* — Los métodos basados en COR empleados en la primera etapa para cada problema fueron los siguientes: CORCH-SCHBL en F_1 , CORRCH-SH en F_2 y arroz, y CORRWM-SH en la línea de media tensión.

Tabla 4.1.: Resultados obtenidos por las diferentes hibridaciones propuestas en las cuatro aplicaciones consideradas

Método	F_1			F_2		
	#R	ECM_{entr}	ECM_{prue}	#R	ECM_{entr}	ECM_{prue}
C+PAL	49	0,213164	0,222263	47	0,005992	0,010902
C+PA-L	49	0,250022	0,280623	47	0,006428	0,012898
CPAL	49	0,244005	0,300322	45	0,006745	0,011838
C-PAL	49	0,216077	0,221788	45	0,006492	0,010727
CL-PA	49	0,179944	0,203371	45	0,008155	0,014808

Método	Arroz			Línea de media tensión		
	#R	ECM_{entr}	ECM_{prue}	#R	ECM_{entr}	ECM_{prue}
C+PAL	17	0,000558	0,001923	49	8.559	12.584
C+PA-L	17	0,000562	0,002052	49	9.708	11.967
CPAL	25	0,000479	0,001942	50	4.751	8.422
C-PAL	25	0,000504	0,002065	48	7.004	11.277
CL-PA	26	0,000561	0,001992	51	9.181	12.432

Estos métodos fueron los que mejores resultados obtuvieron en el estudio experimental realizado en el capítulo 2. Los valores de parámetros para los tres primeros problemas pueden consultarse en dicho capítulo; para el cuarto problema se han considerado los siguientes valores de parámetros: 50 iteraciones, $\eta_{sj} = FVR_1$, $\rho = 0,6$, $\alpha = 1$ y $\beta = 2$.

En la segunda etapa, los valores de parámetros considerados fueron los siguientes: 61 individuos; 2.000 generaciones en F_1 , F_2 y línea de media tensión y 1.000 generaciones en el arroz; 0,6 y 0,2 para las probabilidades de cruce y mutación respectivamente; 0,35 y 5 respectivamente para los factores de ponderación del operador de cruce max-min-aritmético y de mutación no uniforme en las partes P y A.

- *Método C+PA-L* — Los métodos basados en COR empleados en la primera etapa para cada problema coinciden con los del método anterior.

En la segunda etapa, los valores de parámetros considerados fueron los si-

guientes: 62 individuos (31 para cada especie); 2.000 generaciones en F_1 , F_2 y línea de media tensión y 1.000 generaciones en el arroz; 0,6 y 0,2 para las probabilidades de cruce y mutación respectivamente en ambas especies; 0,35 para el factor de ponderación en el operador de cruce max-min-aritmético de la especie 1; se consideraron los dos mejores individuos ($|R_{1|2}| = 2$) y dos individuos aleatorios ($|P_{1|2}|=2$) de cada especie como cooperadores para calcular la adecuación conjunta.

- *Método CPAL* — 61 individuos; 2.000 generaciones en F_1 , F_2 y línea de media tensión y 1.000 generaciones en el arroz; 0,6 y 0,2 para las probabilidades de cruce y mutación respectivamente en ambas especies; 0,35 para el factor de ponderación en el operador de cruce max-min-aritmético de las partes P y A.

Los enfoques de la metodología COR según el problema fueron los siguientes: CORCH en F_1 , CORRCH en F_2 y arroz, y CORRWM en la línea de media tensión. Para seleccionar el enfoque de la metodología COR más adecuado en cada caso hemos atendido a la naturaleza de cada problema.

En primer lugar, debemos comentar que, debido al empleo de técnicas de búsqueda adecuadas, nos interesará generar un espacio de búsqueda lo suficientemente enriquecido como para poder acceder a buenas soluciones. Por esta razón, hemos seguido el enfoque guiado por rejilla difusa (CH). El problema eléctrico constituye una excepción ya que en este caso, debido al alto número de variables de entrada y términos lingüísticos considerados, nos interesa seguir un enfoque más restrictivo que permita generar modelos con un número razonable de reglas en términos de interpretabilidad, de modo que hemos escogido el enfoque guiado por ejemplos (WM).

Por otro lado, en general nos interesará dotar a los métodos con la capacidad de eliminar aquellas reglas que no cooperen adecuadamente (enfoque CORR). En este caso, la excepción la constituye el problema F_1 donde, como hemos podido comprobar en los capítulos anteriores, gracias a la buena distribución en el espacio de entrada del conjunto de entrenamiento y a la ausencia de discontinuidades, no resulta necesario eliminar reglas lingüísticas, por lo que seguiremos el enfoque básico (COR).

- *Métodos C-PAL y CL-PA* — 62 individuos (31 para cada especie); 2.000 generaciones en F_1 , F_2 y línea de media tensión y 1.000 generaciones en el arroz;

0,6 y 0,2 para las probabilidades de cruce y mutación respectivamente en ambas especies; 0,35 para el factor de ponderación en el operador de cruce max-min-aritmético de la especie 2; se consideraron los dos mejores individuos ($|R_{1|2}| = 2$) y dos individuos aleatorios ($|P_{1|2}|=2$) de cada especie como cooperadores para calcular la adecuación conjunta. Se siguieron los mismos enfoques de la metodología COR descritos en el método anterior.

4.4.2. Análisis de los Resultados Obtenidos

En vista de los resultados obtenidos, lo primero que podemos destacar es el buen comportamiento de todas las hibridaciones y la alta calidad de los modelos lingüísticos generados. Si comparamos estos resultados con los grados de precisión alcanzados por los métodos analizados en el capítulo 1 (tablas 1.2 y 1.3, páginas 40 y 41) para los dos problemas comunes en ambos estudios experimentales (F_2 y arroz), vemos que incluso los peores resultados de las hibridaciones consideradas en este capítulo superan a los mejores resultados obtenidos en el capítulo 1. La tabla 4.2 recoge un resumen de estos resultados. Además de obtener modelos más precisos con las hibridaciones analizadas, la interpretabilidad alcanzada es significativamente más alta ya que se genera menor número de reglas (comparado con el método M-L) y se mantiene una semántica global (comparado con el método de MD P-FCS1).

Tabla 4.2.: Mejores resultados obtenidos por los métodos de ML y MD analizados en el capítulo 1 y por el método PAL-Ajus del capítulo 3

Aplicación	Método	#R	ECM _{entr}	ECM _{prue}
F_2	M-L	96	0,011865	0,014518
Arroz	P-FCS1	51	0,000596	0,009055
F_1	WM+PAL-Ajus	49	0,208114	0,231909
F_2	T+PAL-Ajus	47	0,009257	0,018448
Arroz	T+PAL-Ajus	16	0,000573	0,002034
Línea de media tensión	T+PAL-Ajus	534	4.483	4.958

También podemos comprobar en la tabla 4.2 que el ajuste PAL obtiene resultados más prometedores cuando se ajustan modelos obtenidos con la metodología COR (C+PAL) en lugar de los obtenidos por otros métodos, como los analizados en el capítulo anterior (WM+PAL-Ajus y T+PAL-Ajus).

La hibridación C+PAL obtiene modelos más precisos en los problemas F_2 y arroz y notablemente más simples (menor número de reglas) en el problema eléctrico. En el problema F_1 , sin embargo, el uso de la metodología COR como primera fase del proceso de derivación no supone mejoras significativas.

Centrándonos en los cinco métodos analizados, podemos observar que, en general, la derivación simultánea (métodos CPAL, C-PAL y CL-PA) alcanza unos grados de precisión ligeramente superiores a la derivación secuencial (métodos C+PAL y C+PA-L). Como hemos comentado, esto se debe a que se considera adecuadamente la fuerte interdependencia existente entre los diferentes componentes. Sólo en el problema F_2 se invierte dicho comportamiento. Este hecho parece estar claramente relacionado con la naturaleza del problema. Como vimos en el capítulo 2, una vez que se consideran adecuadamente las discontinuidades existentes en este problema, el aprendizaje de las reglas que mejor cooperan resulta ser lo suficientemente eficaz como para que pueda separarse el proceso de derivación en dos etapas. De esta forma, la segunda fase trabaja con un espacio de búsqueda más reducido que le permite alcanzar mejores soluciones.

En cuanto a la técnica de aprendizaje empleada (AGs básicos o CC), los resultados son más contradictorios. Sólo en el problema F_1 la CC demuestra ser más adecuada, mientras que en el resto de casos los AGs básicos muestran un comportamiento similar o incluso mejor. Este hecho nos sugiere que, aunque la CC parece ser muy adecuada para abordar problemas de hibridación de componentes, diseñar un esquema de interacción eficaz resulta muy complejo. Además, para considerar adecuadamente las interdependencias de componentes con la CC sería necesario dividir en más especies el proceso evolutivo, lo cual provoca un crecimiento geométrico de la complejidad, inabordable con las propuestas coevolutivas actuales.

Por otro lado, si analizamos los modelos lingüísticos generados podremos comprobar que el buen grado de precisión alcanzado no supone perder la capacidad de comprensibilidad en exceso. En la figura 4.4 podemos observar el modelo obtenido por el método CL-PA en el problema F_1 , en el que las semánticas y modificadores lingüísticos empleados mantienen una interesante simetría que permite interpretar el comportamiento del modelo con facilidad. Igualmente, el modelo obtenido por el método C+PAL en el problema F_2 muestra un resultado de interpretabilidad similar (véase la figura 4.5).

En la figura 4.6 y tabla 4.3 se representa el modelo obtenido (BD y BR respectivamente) por el método C-PAL para una partición de datos concreta en el problema

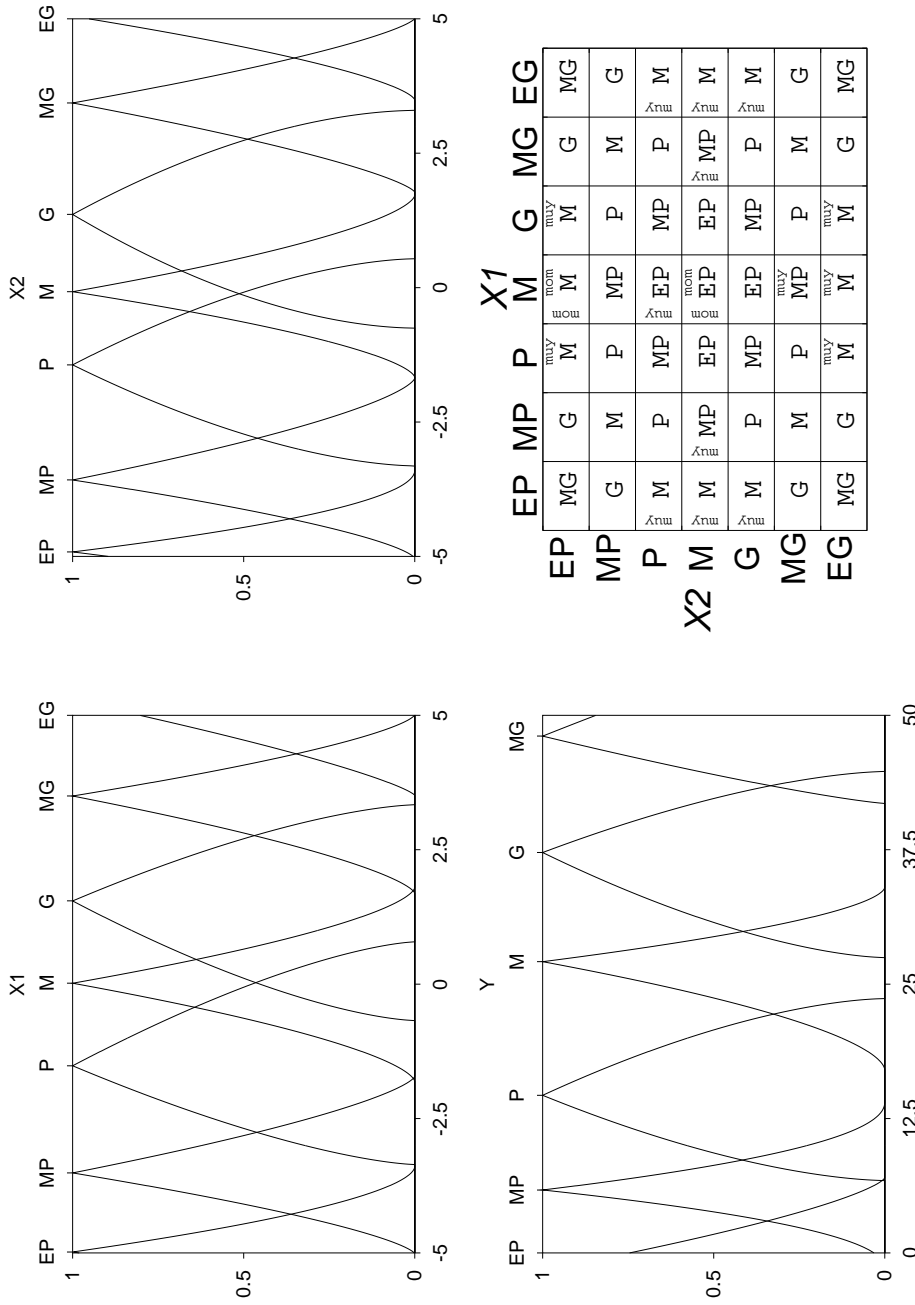


Figura 4.4.: Modelo lingüístico generado por el método CL-PA para el problema F₁

(#R=49, $ECM_{entr/prue} = 0,179944 / 0,203371$)

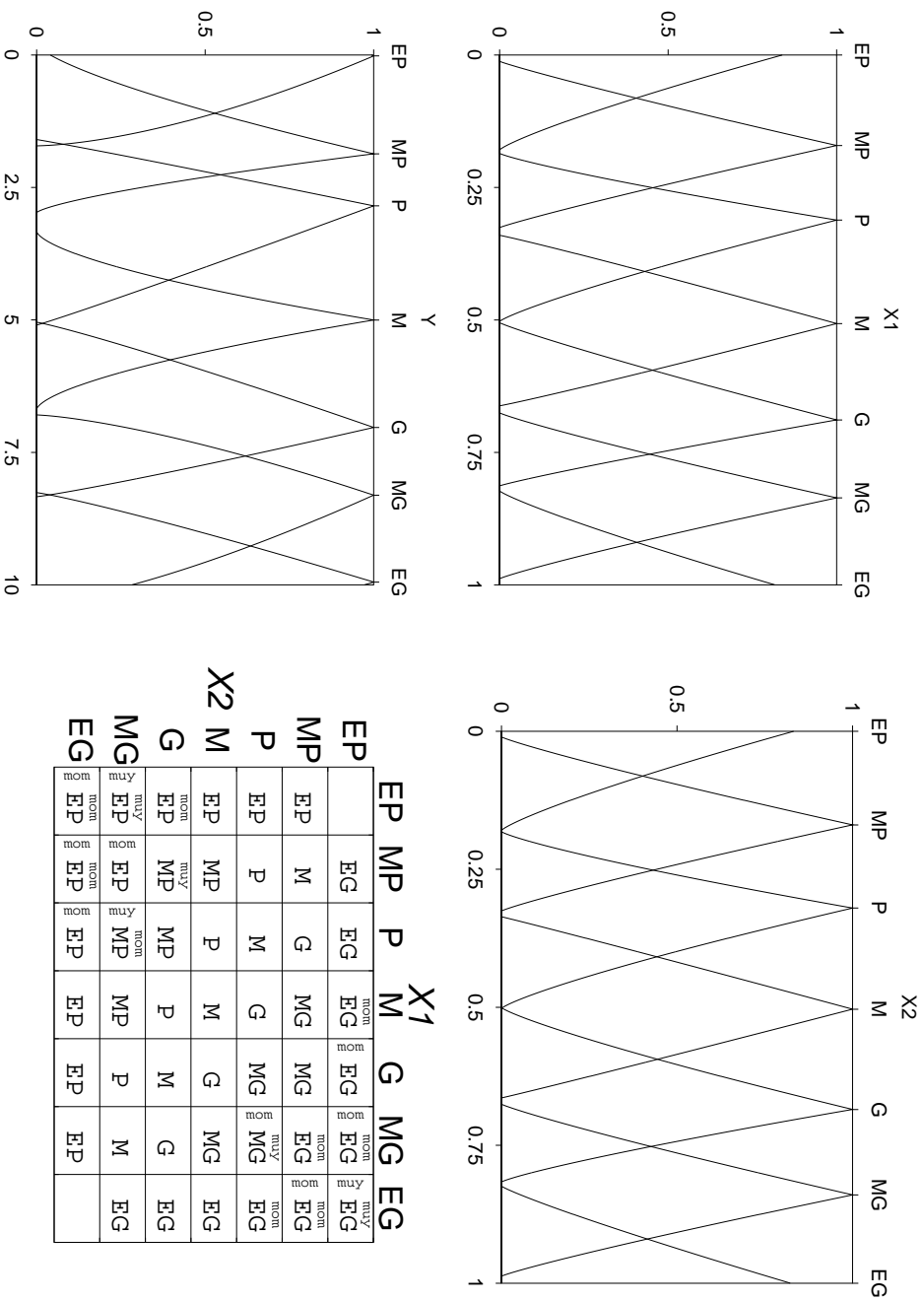


Figura 4.5.: Modelo lingüístico generado por el método C+PAL para el problema F₂
 (#R=47, ECM_{entr/prue} = 0,005992 / 0,010902)

del arroz. En este caso, si bien los modificadores lingüísticos empleados no parecen entorpecer en exceso la legibilidad de la BR generada, la semántica obtenida sí muestra cierta pérdida de interpretabilidad.

Este hecho puede deberse a dos circunstancias relacionadas entre sí. Por una parte, la gran capacidad de aproximación de las hibridaciones propuestas conlleva que, en este problema, los modelos generados estén excesivamente adaptados al conjunto de entrenamiento y se obtengan BDs intrincadas. Esto queda claramente patente con el efecto de sobreaprendizaje existente en este caso, obteniendo un grado de generalización (ECM_{prue}) ligeramente peor. Por otra parte, el uso de sólo dos términos lingüísticos para cada variable hace que sea necesario flexibilizar en mayor medida las formas de las funciones de pertenencia para alcanzar el alto grado de aproximación mencionado.

Estas circunstancias, sobreaprendizaje y pérdida de interpretabilidad, pueden solucionarse imponiendo restricciones ligeramente más estrictas (intervalos de variación más acotados) que eviten la excesiva aproximación, así como incrementando el número de términos lingüísticos considerados.

Finalmente, en la figura 4.7 y tabla 4.4 se muestra el modelo lingüístico generado (BD y BR respectivamente) por el método CPAL en el problema de la estimación de costo de mantenimiento en líneas eléctricas de media tensión. Como podemos observar, no sólo se alcanza un alto grado de precisión sino que, además, el modelo obtenido goza de una interpretabilidad satisfactoria.

4.5. Sumario

Hemos dedicado este capítulo a abordar un análisis que actualmente está cobrando un gran interés: la hibridación de componentes basados en Lógica Difusa dentro de la Computación Evolutiva.

Para ello, se han considerado los cuatro mecanismos de flexibilización del ML propuestos en los capítulos anteriores y se han planteado cinco combinaciones distintas para considerar adecuadamente la estrecha relación existente entre ellos. Tras efectuar un estudio experimental hemos obtenido las siguientes conclusiones:

- En general, el hecho de realizar las diferentes tareas al mismo tiempo mejora el grado de precisión alcanzado ya que se considera adecuadamente la interdependencia entre los componentes.

4. Hibridación de Diferentes Mejoras al Modelado Lingüístico

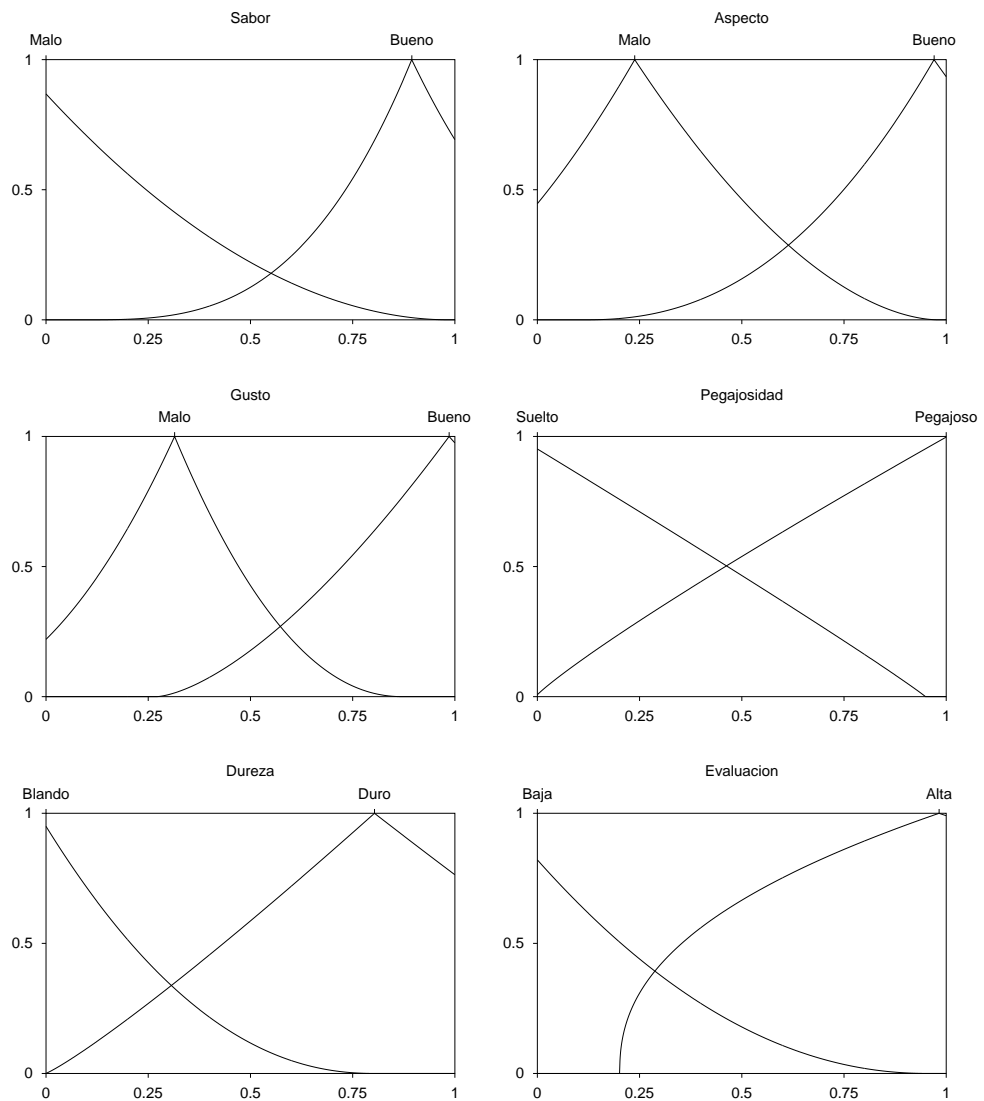


Figura 4.6.: BD generada por el método C-PAL para una partición específica del conjunto de datos del problema del arroz
 (#R=25, $ECM_{entr/prue} = 0.000443 / 0.001950$)

Regla	Sabor	Aspecto	Gusto	Pegajosidad	Dureza	Evaluación
R1	Malo	Malo	Malo	Suelto	Duro	Baja
R2	Malo	Malo	Malo	Pegajoso	Blando	Baja
R3	mom	muy	muy	Suelto	Blando	mom
R4	mom	muy	muy	Suelto	Duro	Alta
R5	muy	mom	mom	Pegajoso	Blando	Baja
R6	mom	muy	mom	Pegajoso	Duro	Baja
R7	Malo	Bueno	Malo	Suelto	Blando	Baja
R8	Malo	Bueno	Malo	Pegajoso	Blando	Alta
R9	mom	mom	mom	Suelto	Blando	Alta
R10	mom	muy	Bueno	Suelto	Duro	Baja
R11	muy	Bueno	Bueno	Pegajoso	Blando	Alta
R12	Malo	Bueno	Bueno	Pegajoso	Duro	Alta
R13	Bueno	muy	Malo	Suelto	Blando	Baja
R14	Bueno	muy	Malo	Suelto	Duro	Baja
R15	muy	muy	Malo	Pegajoso	Blando	Baja
R16	muy	mom	Malo	Pegajoso	Duro	Alta
R17	Bueno	muy	Bueno	Suelto	Blando	Alta
R18	Bueno	muy	Bueno	Suelto	Duro	Alta
R19	Bueno	Malo	Bueno	Pegajoso	Blando	Alta
R20	Bueno	mom	Malo	Suelto	Duro	Baja
R21	Bueno	muy	Malo	Pegajoso	Blando	Baja
R22	mom	mom	Malo	Pegajoso	Duro	Baja
R23	muy	Bueno	Bueno	Suelto	Duro	Alta
R24	Bueno	Bueno	Bueno	Pegajoso	Blando	Alta
R25	Bueno	Bueno	Bueno	Pegajoso	Duro	Alta

Tabla 4.3.: BR generada por el método C-PAL para una partición específica del conjunto de datos del problema del arroz (#R=25, $ECM_{entr/prue} = 0.000443 / 0.001950$)

4. Hibridación de Diferentes Mejoras al Modelado Lingüístico

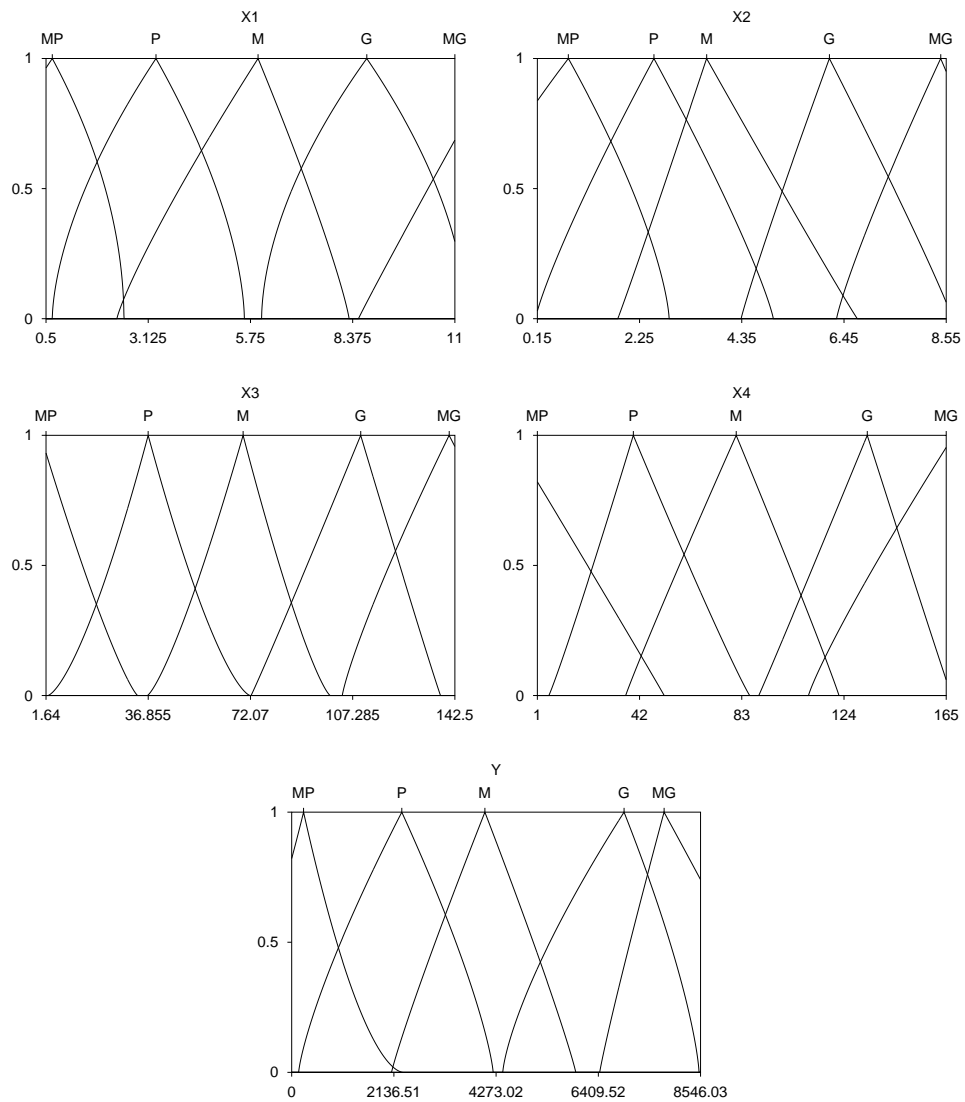


Figura 4.7.: BD generada por el método CPAL para el problema de la línea de media tensión ($\#R=50$, $ECM_{entr/prue} = 4.751 / 8.422$)

Regla	X ₁	X ₂	X ₃	X ₄	Y	Regla	X ₁	X ₂	X ₃	X ₄	Y
R ₁	MP	MP	MP	MP	MP	R ₂₆	G	mom	P	P	P
R ₂	muy	mom	mom	P	P	R ₂₇	muy	mom	muy	mom	mom
R ₃	muy	MP	MP	mom	muy	R ₂₈	mom	mom	M	M	M
R ₄	MP	P	mom	P	MP	R ₂₉	G	M	M	M	G
R ₅	MP	P	mom	muy	MP	R ₃₀	G	M	M	G	G
R ₆	P	MP	MP	MP	mom	R ₃₁	muy	M	P	mom	P
R ₇	P	MP	MP	MP	mom	R ₃₂	mom	M	P	mom	mom
R ₈	mom	MP	mom	P	mom	R ₃₃	mom	M	P	M	mom
R ₉	mom	P	mom	MP	mom	R ₃₄	mom	M	P	M	mom
R ₁₀	mom	P	mom	P	mom	R ₃₅	G	M	P	M	mom
R ₁₁	P	P	muy	mom	MP	R ₃₆	mom	G	G	muy	muy
R ₁₂	P	P	MP	muy	P	R ₃₇	muy	G	G	muy	G
R ₁₃	P	M	mom	MP	M	R ₃₈	G	G	M	P	M
R ₁₄	P	M	mom	mom	muy	R ₃₉	G	muy	M	mom	M
R ₁₅	mom	M	muy	muy	mom	R ₄₀	G	muy	M	mom	G
R ₁₆	M	mom	P	P	muy	R ₄₁	G	MG	MG	P	MG
R ₁₇	M	P	MP	MP	MP	R ₄₂	G	MG	MG	M	MG
R ₁₈	mom	P	MP	MP	mom	R ₄₃	G	MG	G	muy	M
R ₁₉	mom	P	MP	muy	P	R ₄₄	mom	MG	mom	muy	mom
R ₂₀	M	M	mom	mom	M	R ₄₅	mom	MG	mom	mom	mom
R ₂₁	M	M	M	M	mom	R ₄₆	mom	MG	mom	M	G
R ₂₂	mom	M	muy	mom	muy	R ₄₇	mom	P	muy	mom	P
R ₂₃	mom	M	mom	mom	muy	R ₄₈	mom	P	mom	mom	MG
R ₂₄	mom	M	muy	mom	mom	R ₄₉	muy	MG	M	muy	G
R ₂₅	M	muy	M	M	mom	R ₅₀	muy	MG	M	muy	MG

Tabla 4.4.: BR generada por el método CPAL para el problema de la línea de media tensión (#R=50, ECM_{entr/prue} = 4.751 / 8.422)

4. *Hibridación de Diferentes Mejoras al Modelado Lingüístico*

- El empleo de técnicas de búsqueda más avanzadas tales como la CC para tratar las hibridaciones planteadas no muestra ventajas notables. Aunque en principio parecen herramientas adecuadas para tales propósitos, su diseño parece ser bastante complejo. Así, resulta crucial definir un esquema de interacción apropiado y extender la coevolución a más de dos especies, lo cual supone un crecimiento de complejidad difícil de tratar.
- Si la hibridación de las extensiones al ML se realiza convenientemente, los modelos lingüísticos generados presentan un alto grado de precisión conservando una buena interpretabilidad.

Comentarios Finales

La paciencia es amarga, pero su fruto es dulce.

JEAN-JACQUES ROUSSEAU (1712–1778), FILÓSOFO FRANCÉS

Dedicaremos esta sección a resumir brevemente los resultados obtenidos y a destacar las conclusiones que esta memoria puede aportar. Además, comentaremos algunos aspectos sobre trabajos futuros que siguen la línea aquí expuesta y sobre otras líneas que se pueden derivar.

A. Resultados Obtenidos y Conclusiones

Hemos presentado diferentes métodos para derivar modelos lingüísticos a partir de datos numéricos. Esto se ha realizado bajo la perspectiva de obtener modelos precisos manteniendo un alto grado de comprensibilidad. Los siguientes apartados contienen las conclusiones deducidas durante esta labor.

Modelado Lingüístico frente a Modelado Difuso

Partiendo del hecho de que mayor flexibilidad del modelo no implica mayor precisión pues la complejidad para realizar el proceso de modelado crece, hemos lanzado la hipótesis de que se podría obtener un mejor comportamiento manteniendo un buen equilibrio entre la libertad del modelo (flexibilidad del modelo) y la dificultad que supone en el proceso de derivación (simplicidad del modelado).

De esta forma, lejos de la resignada opinión presente en algunos sectores de la literatura especializada que enclava al ML como herramientas con buena legibilidad

pero mala precisión, hemos demostrado que es posible flexibilizarlo para mejorar notablemente la precisión conservando su buena interpretabilidad.

Para hacer esto, hemos introducido dos tipos diferentes de extensiones [CCH00a]:

- Mejora en el *proceso de derivación*, usando mecanismos de generación sofisticados o definiendo las funciones de pertenencia.
- Extensión en la *estructura del modelo*, empleando reglas con modificadores lingüísticos, reglas con varios consecuentes o reglas ponderadas.

Mediante un profundo estudio experimental donde se han analizado diferentes propuestas existentes en la literatura, se han obtenido resultados muy interesantes combinando métodos de aprendizaje lingüísticos sencillos, que no emplean mecanismos de generación de reglas complejos, con un proceso de refinamiento *a posteriori* (o bien ajustando las funciones de pertenencia o bien añadiendo modificadores lingüísticos).

Esto nos ha motivado para proponer métodos que realicen procesos de aprendizaje *grosso modo* obteniendo rápidamente modelos con una precisión aceptable, combinados con métodos de ajuste que ofrezcan al modelo la precisión finalmente deseada.

Aprendizaje Rápido de Modelos Lingüísticos con Buena Precisión y Máxima Interpretabilidad

Se ha elaborado una nueva metodología para generar modelos lingüísticos eficaces y simples, la metodología COR [CCH00c, CCH00f]. Esta se basa en considerar la cooperación entre las reglas lingüísticas en el proceso de generación sacando partido del razonamiento interpolado que desarrollan los SBRDs. La forma de actuar está compuesta de dos etapas: generación del conjunto de consecuentes candidatos en cada subespacio y búsqueda combinatoria de los consecuentes que mejor cooperan.

Siguiendo esta metodología, hemos desarrollado varios métodos concretos empleando diferentes técnicas de búsqueda tales como el ES, la BT, los AGs y la OCH [CCH01]. Por otro lado, hemos propuesto algunas extensiones a esta metodología que permite obtener modelos aún más eficaces y de forma aún más rápida [CCH00d]. Entre todos los métodos propuestos, los basados en la OCH han dado los mejores resultados combinando eficiencia y precisión [CCH00d, CCH00e, CCH00b].

El estudio realizado nos permite concluir que la consideración de reglas cooperativas globales mejora el comportamiento de los modelos lingüísticos y que la generación de BCs obteniendo primero un conjunto de reglas candidatas y buscando después la mejor combinación de estas resulta una forma muy adecuada de llevar a cabo dicho proceso de aprendizaje. Además, el uso de reglas cooperativas hace que los modelos generados sean más robustos frente al ruido.

Ajuste de Modelos Lingüísticos

Acorde con la filosofía de derivación mencionada con anterioridad, también hemos propuesto un método de ajuste que, partiendo de un modelo previamente generado, realiza alteraciones en él para mejorar su precisión manteniendo una buena interpretabilidad [CCdJH01b, CCdJH01c].

Para ganar en precisión, se ha realizado un ajuste a dos niveles de especificidad diferentes, obteniendo efectos macroscópicos y microscópicos. Por otro lado, se han combinado variaciones lineales con no lineales, lo cual mejora la precisión al incidir en diferentes aspectos en el proceso de inferencia.

Para mantener la interpretabilidad, se han considerado intervalos de variación adecuados que limitan la libertad del ajuste, consiguiendo que los modelos ajustados sigan manteniendo una buena interpretabilidad. Por otro lado, hemos empleado modificadores lingüísticos para aportar un significado claro al efecto obtenido. El conjunto de modificadores está igualmente limitado a sólo dos elementos.

Los resultados obtenidos nos dirigen a pensar que la combinación de ajuste a dos niveles mejora significativamente la precisión de los modelos lingüísticos.

Hibridación de las Mejoras al Modelado Lingüístico

Después de comprobar el buen funcionamiento de la metodología de aprendizaje y el proceso de ajuste, nos planteamos la combinación adecuada de ambos procedimientos para considerar adecuadamente la relación existente entre ellos.

En este sentido, hemos realizado un estudio de hibridación de componentes aunando la lógica difusa y la computación evolutiva. Esta consideración disfruta actualmente de una buena aceptación en el ámbito científico. Para combinar adecuadamente las mejoras al ML propuestas con anterioridad hemos distinguido entre procesos de derivación secuencial (diseño de la BC por etapas) y simultánea (diseño completo de la BC en un único paso), y entre AGs básicos y algoritmos coevolutivos cooperativos.

Se ha podido comprobar que las diferentes formas de hibridación afectan significativamente a los resultados obtenidos y que, cuando esta se realiza convenientemente, se mejora notablemente la precisión de los modelos lingüísticos manteniendo un grado de interpretabilidad aceptable. Podemos concluir igualmente que, en general, el hecho de realizar las distintas mejoras al ML en un único paso mejora el grado de precisión alcanzado ya que se considera adecuadamente la fuerte interdependencia existente entre los diferentes componentes.

Por otro lado, los algoritmos coevolutivos cooperativos no han obtenido resultados destacables en la experimentación realizada. No obstante, estas técnicas de búsqueda más avanzadas parecen ser muy adecuadas para tareas de hibridación. Esto nos sugiere que resulta necesario un estudio más profundo en este novedoso paradigma de búsqueda abordándolo principalmente desde dos puntos de vista. En primer lugar, se deberían considerar más de dos especies para tratar adecuadamente aquellos problemas altamente descomponibles como nuestro caso. Esto supone un crecimiento geométrico de la complejidad difícil de tratar. Además, es necesario diseñar esquemas de interacción más complejos entre las especies que consideren adecuadamente la interdependencia existente entre los componentes y faciliten la escalabilidad a más especies.

B. Líneas de Investigación Futuras

A continuación, consideraremos algunas posibles extensiones sobre los métodos propuestos en esta memoria y discutiremos varias líneas de trabajo inmediatas.

Mejoras en el Método de Ajuste

Para mantener un mayor grado de interpretabilidad nos planteamos imponer nuevas restricciones al proceso de ajuste de los modelos lingüísticos, que pueden extenderse al método de derivación simultánea basada en algoritmos coevolutivos cooperativos. Entre otras, podemos establecer las siguientes restricciones:

- Considerar el porcentaje de uso de los modificadores lingüísticos en las estructuras superficiales penalizando a aquellas soluciones que impliquen usar estos con excesiva frecuencia.
- Realizar variaciones lineales más restrictivas. Por ejemplo, forzando a que la

semántica ajustada sea *fuerte* en todo momento, es decir, la suma de los grados de pertenencia de cualquier valor del dominio a cada una de las etiquetas sea 1. Esto puede realizarse ajustando el universo de discurso mediante factores de escala y distribuyendo la semántica uniformemente.

Además, podríamos realizar el ajuste con otras metaheurísticas distintas de la computación evolutiva. Sería particularmente interesante realizarlo con algoritmos de OCH, aunque esto supondría manejar entornos continuos y estas metaheurísticas aún no se encuentran muy desarrolladas en este campo. Como continuación de la investigación, nos proponemos abordar este problema empleando funciones de distribución de probabilidad para combinar la información heurística y memorística.

Mejoras en la Coevolución Cooperativa

Para mejorar el algoritmo coevolutivo cooperativo proponemos dos tipos de consideraciones:

- Por un lado, deberemos mejorar el esquema de interacción entre las especies para considerar de forma más adecuada las interdependencias existentes. Además, debería considerarse el uso de más de dos especies para descomponer el problema en mayor grado. No obstante, esto supone un crecimiento geométrico que debe ser cuidadosamente tratado.
- Por otro lado, sería muy interesante extender el concepto de coevolución, hasta ahora empleado en la computación evolutiva, a otras metaheurísticas tales como la OCH. De esta forma, la interacción seguiría siendo la misma pero optimizaríamos la evolución de las especies considerando la información heurística existente.

Proceso de Derivación más Completo

En el capítulo 1 hemos introducido varias extensiones posibles al ML flexibilizando la estructura del modelo. En concreto, sugeríamos el uso de modificadores lingüísticos, varios consecuentes por regla y reglas ponderadas. Aunque en esta memoria sólo nos hemos centrado en el primer aspecto, sería interesante profundizar la investigación en el resto de alternativas, manteniendo los mismos criterios de precisión e interpretabilidad considerados.

Además, dada la evidente descomposición del problema, el tratamiento de las extensiones comentadas con coevolución cooperativa podría ser muy interesante. Conforme aumentemos los grados de libertad, tendremos que aplicar restricciones más severas en aras de mantener la interpretabilidad y acotar el espacio de búsqueda.

Por otra parte, también nos planteamos extender el proceso de derivación para definir automáticamente otros conceptos tales como el número de términos lingüísticos de cada variable [CHV01], número de variables [CCdJH01a], semánticas iniciales extraídas a partir de la información disponible, participación más activa de la información de expertos cuando se posea, etc.

Mecanismos para Evitar el Sobreaprendizaje

Sin duda, uno de los problemas fundamentales con los que hay que lidiar continuamente dentro del área del Modelado de Sistemas es el sobreaprendizaje. Este es sin duda un efecto no deseable, ya que la excesiva aproximación conduce a una mala predicción que invalida la utilidad del modelo para representar con fidelidad el sistema a modelar.

En esta memoria hemos tratado de manejar adecuadamente este inconveniente empleando *estructuras* que incidan con suavidad en el modelo, asegurándonos así que se realiza una adecuada generalización. No obstante, el problema podría plantearse desde otra interesante perspectiva proveyendo al *método de derivación* con herramientas que eviten este sobreaprendizaje. En esta línea, una posibilidad consistiría en alternar entre diferentes subconjuntos de entrenamiento durante el proceso de búsqueda.

Interpretación Lingüística a Posteriori

Esta memoria se ha centrado continuamente en alcanzar el equilibrio adecuado entre interpretabilidad y precisión. Para ello, hemos seguido el camino hacia el equilibrio ideal en un único sentido buscando las formas de flexibilizar sin excedernos. Otro enfoque interesante sería permitir obtener modelos muy flexibles, para posteriormente aplicar un proceso que interprete lingüísticamente el resultado. Esto supone establecer qué reglas son importantes, qué significado posee cada conjunto difuso derivado, por qué resulta necesario que determinado componente tenga una determinada forma. . .

Apéndices

A. Sistemas Basados en Reglas Difusas

*Con el conocimiento se acrecientan
las dudas.*

JOHANN WOLFGANG VON GOETHE
(1749–1832), ESCRITOR ALEMÁN

Este apéndice está dedicado a introducir los SBRDs lingüísticos y aproximativos empleados respectivamente para el ML y el MD. 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.

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 $A = A_1 \times \dots \times A_n$ (donde el símbolo ‘ \times ’ denota el producto cartesiano) tenemos

SI X es A ENTONCES Y es B ,

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

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

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

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

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

$$f^* = A_1 \times B_1 + \dots + A_r \times B_r,$$

o, de forma más compacta

$$f^* = \sum_{i=1}^r A_i \times B_i.$$

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 de un grafo difuso sin restricciones.

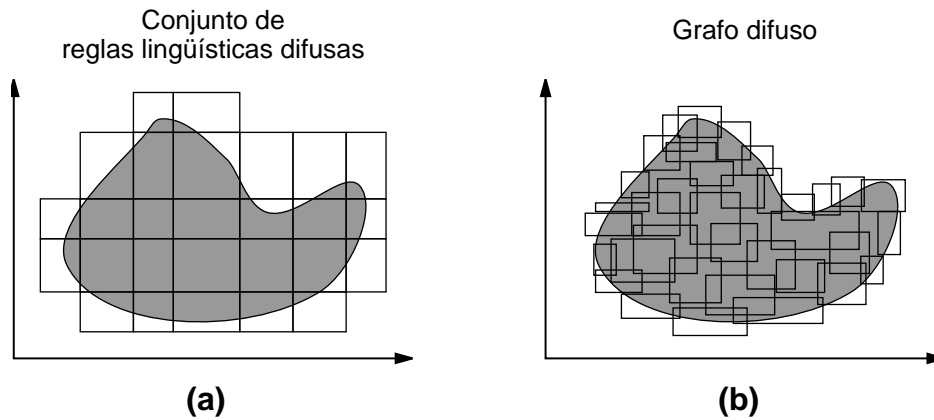


Figura A.1.: Representación aproximada de relaciones: (a) considerando grafos difusos restringidos, (b) considerando grafos difusos sin restricciones

Los grados 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 ML. Los segundos, por contra, consideran como principal requisito la aproximación al modelo, de forma que son más adecuados para MD. 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 *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

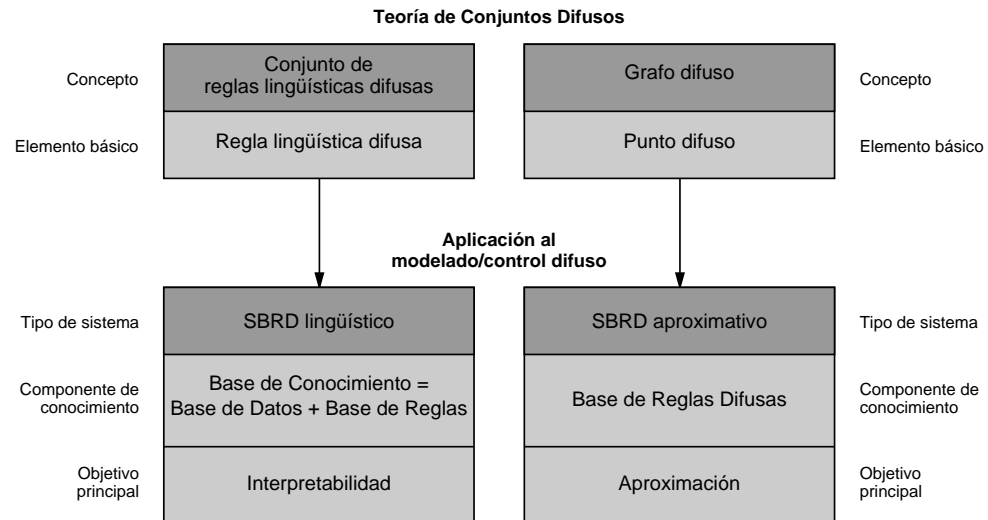


Figura A.2.: Paralelismo entre SBRDs lingüísticos (obtenidos a partir de un conjunto de reglas lingüísticas difusas) y SBRDs aproximativos (obtenidos a partir de un grafo difuso sin restricciones)

- SBRDs con *partición enrejada* 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.

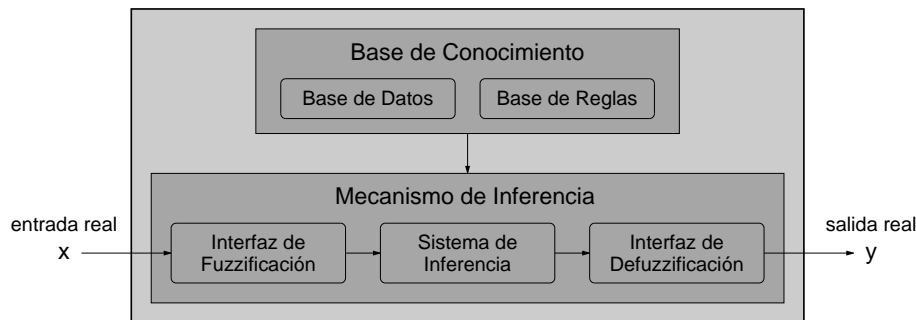


Figura A.3.: Estructura básica de un SBRD lingüístico

La figura A.3 muestra la estructura general de los SBRDs lingüísticos. Tal como comentaremos en la sección A.4, en la que se estudiará detenidamente cada uno de los componentes de este tipo de sistemas, la interfaz de fuzzificación establece una aplicación entre valores precisos en el dominio 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 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 conoci-

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

miento 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 ML. 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 1.4 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 MD 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 Base de Reglas Difusas (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.

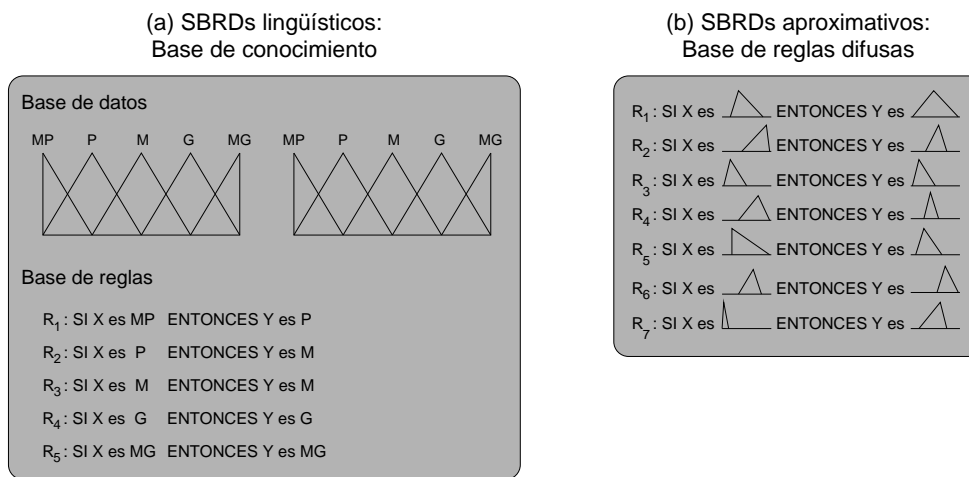


Figura A.4.: Comparación gráfica entre una BC lingüística y una BRD aproximativa

A.4. Análisis de los Componentes de los Sistemas Basados en Reglas Difusas Lingüísticos

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.4.1. 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.4.2. 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:

$$\mu_{A'}(x) \begin{cases} 1 - \frac{|x-x_0|}{\epsilon}, & \text{si } |x-x_0| \leq \epsilon \\ 0, & \text{en otro caso.} \end{cases}$$

A.4.3. 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:

$$\mu_R(x, y) = I(\mu_A(x), \mu_B(y)), \forall x \in U, y \in V,$$

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, TV85a].
- *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 [TV85a]. Además de las mencionadas funciones, otros autores sugieren el uso de t -conormas y operadores externos a las anteriores definiciones [CK89, CHP97, KKS85]. En [AT92, TV85a, TV85b] 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.4.4. 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].

B. Técnicas de Búsqueda

*Por todas partes te busco
sin encontrarte jamás,
y en todas partes te encuentro
sólo por irte a buscar.*

ANTONIO MACHADO (1875–1939),
POETA Y PROSISTA ESPAÑOL

Dedicaremos este apéndice a describir las líneas generales de las técnicas de búsqueda empleadas a lo largo de esta memoria, los algoritmos genéticos (AGs) y las tres técnicas exclusivas del capítulo 2 para la metodología COR: el enfriamiento simulado (ES), la búsqueda tabú (BT), y la optimización mediante colonias de hormigas (OCH).

B.1. Enfriamiento Simulado

El ES [vLA87] deriva de la analogía entre la mecánica estadística de partículas de un compuesto (líquido o sólido) y la búsqueda de soluciones en problemas complejos de optimización combinatoria. Dado un compuesto cualquiera, no todas sus moléculas tienen la misma energía, sino que estas se encuentran distribuidas en distintos niveles, el menor de los cuales se denomina *estado fundamental*. Los diferentes estratos de las partículas muestran diferentes niveles de energía. Si el compuesto está a cero grados Kelvin, todas las moléculas están en su estado fundamental, y conforme mayor es la temperatura, dichas moléculas ocupan niveles superiores. Si el estado del compuesto se define por el estrato en que se encuentran sus partículas y por su energía, el algoritmo de Metrópolis es un modelo matemático que describe la transición del compuesto desde el estado i con energía $E(i)$ hacia el estado j con energía $E(j)$ a una temperatura T mediante un mecanismo sencillo.

El algoritmo de Metrópolis describe el proceso de cristalización de un líquido: con temperaturas altas, las partículas son muy volátiles y se desplazan a gran velocidad debido a la gran cantidad de energía que poseen. Por contra, a temperaturas bajas, las partículas pierden movilidad como resultado de la pérdida de energía, llegando finalmente a un estado de equilibrio cuando se forma el cristal.

Tratando de simular este proceso, el ES busca la solución óptima de problemas de optimización combinatoria. Supongamos que necesitamos minimizar una función de coste definida por varias variables. Podemos utilizar un mecanismo iterativo sencillo, conocido como *búsqueda local*, para encontrar el coste mínimo. En un proceso de búsqueda local, se parte de una solución inicial y se propone aleatoriamente una nueva solución. Si el coste de la nueva solución es menor que el de la solución actual, la nueva solución reemplaza a la antigua. Si el coste de la nueva solución es mayor, esta no se acepta y se genera una nueva solución otra vez de forma aleatoria a partir de la antigua. El proceso continúa hasta que se halla la solución con coste mínimo. Desafortunadamente, la *búsqueda local* suele quedar atrapada en mínimos locales. Para evitar este problema, el ES permite, ocasionalmente, movimientos hacia peores soluciones, es decir, una transición de estado puede dar como resultado que se acepte una solución con un coste mayor que la solución actual. Esa es la esencia del ES.

De esta manera, la probabilidad de aceptación de una nueva solución candidata (S_{cand}), respecto de la solución actual (S_{act}) está definida por el *criterio de Metrópolis*:

$$P_{acep}(S_{cand}) = \begin{cases} 1, & \text{si } Coste(S_{cand}) < Coste(S_{act}) \\ \exp\left(-\frac{Coste(S_{cand}) - Coste(S_{act})}{T}\right), & \text{en otro caso} \end{cases},$$

donde T es un parámetro del algoritmo que emula el comportamiento de la temperatura en el proceso real. Es decir, va reduciendo su valor conforme aumenta el número de iteraciones.

El ES requiere dos operaciones fundamentales:

- Una operación denominada *mecanismo de enfriamiento*, que guía el descenso de la temperatura y, por tanto, la disminución de la probabilidad de aceptación de soluciones peores, además de especificar el criterio de parada del algoritmo. Existen varios mecanismos de enfriamiento propuestos en la literatura especializada [Sor91].

- Un proceso de relajación estocástico que busca soluciones de equilibrio para cada iteración, donde la temperatura permanece constante. En las primeras iteraciones, el valor de la temperatura es elevado y, por tanto, se aceptan bastantes soluciones con función de coste peor que la solución actual. Si las iteraciones fuesen muy largas, un alto número de soluciones peores aceptadas podría derivar al algoritmo hacia zonas alejadas de la solución óptima.

Se ha demostrado que el ES es capaz de encontrar las soluciones óptimas asintóticamente, es decir, que encuentra esa solución en un número infinito de pasos. En la práctica, el ES debe implementarse para que se ejecute en un tiempo finito. Si no, no produciría ninguna ventaja respecto a una simple búsqueda exhaustiva.

Por tanto, en una implementación de un algoritmo de ES para resolver un problema es necesario especificar lo siguiente:

- La temperatura inicial.
- El criterio que determina el número de transiciones de estado para cada temperatura.
- La función que decrementa la temperatura.
- La temperatura final.
- El criterio de parada del algoritmo.

Los subapartados siguientes presentan la estructura básica del ES y un ejemplo de mecanismo de enfriamiento.

B.1.1. Estructura del Enfriamiento Simulado

El planteamiento básico del algoritmo de ES se muestra en la figura B.1, donde T_0 es la temperatura inicial, S_{act} y S_{vec} son respectivamente la solución actual que maneja el proceso y la solución vecina que se propone, D es la función que decrementa la temperatura y N y M son respectivamente el número de enfriamientos (iteraciones) total y número de soluciones vecinas que se van a generar en cada enfriamiento.

Procedimiento Enfriamiento Simulado

ENTRADA (T_0, D, L, T_f)

$T \leftarrow T_0$

$S_{act} \leftarrow \text{Generar_solución_inicial}$

PARA $k \leftarrow 1$ HASTA N HACER

 PARA $cont \leftarrow 1$ HASTA M HACER

$S_{vec} \leftarrow \text{Generar_solución_vecina}(S_{act})$

$\delta \leftarrow \text{coste}(S_{vec}) - \text{coste}(S_{act})$

 SI ($(\delta < 0)$ O $(U(0, 1) < e^{(-\delta/T)})$) ENTONCES

$S_{act} \leftarrow S_{vec}$

 FIN-PARA

$T \leftarrow D(T)$

FIN-PARA

{Devuelve como salida la mejor de las S_{act} visitadas}

Figura B.1.: Algoritmo básico del enfriamiento simulado

B.1.2. Ejemplo de Mecanismo de Enfriamiento

En esta sección, analizaremos el mecanismo de enfriamiento propuesto por Kirkpatrick [Kir84], que da como resultado un algoritmo de ES con tiempo de ejecución exponencial. Se define de la siguiente forma:

- A la temperatura inicial, se le asigna un valor alto, $T_0 = 10$. Las temperaturas elevadas originan una alta probabilidad de aceptación de una nueva solución.
- La función que decrementa la temperatura es $T_{k+1} = C \cdot T_k$, donde $C < 1$ y normalmente cercano a 1. El autor propone $C = 0,9$.
- El proceso que busca soluciones de equilibrio en cada temperatura se puede caracterizar por el número de transiciones que deben realizarse para que se llegue a un estado de "casi"-equilibrio ϵ , donde ϵ es una constante positiva. Por tanto, es necesario limitar el número de transiciones aceptadas en cada temperatura. Lógicamente, cuanto mayor sea ϵ , menor será el número de soluciones nuevas aceptadas con el que se alcanzará el estado de "casi"-equilibrio ϵ .
- El proceso continúa hasta terminar por alguna de las siguientes condiciones:

- La temperatura es bastante cercana a cero.
- El coste de la mejor solución encontrada no varía en un número consecutivo de iteraciones. Los autores proponen terminar el algoritmo si no se mejora después de tres iteraciones.
- No se acepta ninguna solución en la temperatura actual.

B.2. Búsqueda Tabú

La BT [GL97] es un procedimiento adaptativo para resolver problemas de optimización combinatoria. La técnica guía una heurística de descenso de colinas para continuar la exploración sin caer en óptimos locales a partir de soluciones de las que previamente se emergió.

En cada iteración, se aplica un movimiento admisible a la solución actual, transformándola en el vecino con el costo menor. Se permiten soluciones que incrementan la función de costo, pero se prohíbe realizar un movimiento que se haya efectuado en las últimas iteraciones para evitar ciclos. Las restricciones se basan en una función de memoria a corto plazo que determina durante cuánto tiempo se mantiene una restricción tabú o, alternativamente, qué movimientos se permite en cada iteración.

La figura B.2 muestra un esquema básico de un posible algoritmo de BT, que será el que nosotros emplearemos.

B.3. Algoritmos Genéticos

Los AGs son algoritmos de búsqueda de propósito general que se basan en principios inspirados en la genética de las poblaciones naturales para llevar a cabo un proceso evolutivo sobre soluciones de problemas. Fueron inicialmente propuestos por Holland [Hol75] y han sido posteriormente estudiados en profundidad por otros autores [Gol89, Mic96]. Los AGs han demostrado ser, tanto desde un punto de vista teórico como práctico, una herramienta óptima para proporcionar una búsqueda robusta en espacios complejos, ofreciendo un enfoque válido para solucionar problemas que requieran una búsqueda eficiente y eficaz.

Los AGs se han aplicado con mucho éxito en problemas de búsqueda y optimización. La razón de gran parte de este éxito se debe a su habilidad para explotar la

Procedimiento Búsqueda Tabú

ENTRADA (N, V, L)

$S_{mejor} \leftarrow S_{act} \leftarrow \text{Generar_solución_inicial}$

$MCP \leftarrow \{S_{act}\}$

PARA $i \leftarrow 1$ HASTA N HACER

 PARA $j \leftarrow 1$ HASTA V HACER

$S_j \leftarrow \text{Generar_solución_vecina}(S_{act})$

 Evaluar el costo de la solución, $C(S_j)$

 FIN-PARA

 Ordenar las soluciones S_j de mejor a peor

 SI ($C(S_1) < C(S_{mejor})$) ENTONCES

$S_{mejor} \leftarrow S_{act} \leftarrow S_1$

 SI-NO

$j \leftarrow 1$; $Seguir \leftarrow VERDAD$

 MIENTRAS ($j \leq V$) Y ($Seguir$) HACER

 SI ($S_j \notin MCP$) ENTONCES

$Seguir \leftarrow FALSO$

$S_{act} \leftarrow S_j$

$MCP \leftarrow MCP \cup \{S_j\}$

 SI ($|MCP| > L$) ENTONCES

 Eliminar de MPC la solución
 más antigua

 FIN-SI

 SI ($C(S_{act}) < C(S_{mejor})$) ENTONCES

$S_{mejor} \leftarrow S_{act}$

 FIN-SI

 SI-NO

$j \leftarrow j + 1$

 FIN-SI

 FIN-MIENTRAS

 FIN-SI

FIN-PARA

{Devuelve como salida la solución S_{mejor} }

Figura B.2.: Algoritmo básico de la búsqueda tabú

información que van acumulando sobre el espacio de búsqueda que manejan, desconocido inicialmente, lo que les permite redirigir posteriormente la búsqueda hacia subespacios útiles. La *capacidad de adaptación* que presentan es su característica principal, especialmente en espacios de búsqueda grandes, complejos y con poca información disponible, en los que las técnicas clásicas de búsqueda (enumerativas, heurísticas. . .) no presentan buenos resultados.

La idea básica de estos algoritmos consiste en mantener una población de individuos que codifican soluciones del problema. Dichos individuos emplean una representación genética para codificar los valores de las características parciales que definen las distintas soluciones. Debido a ello, cada individuo recibe el nombre de *cromosoma* y cada una de sus componentes el de *gen*.

Los cromosomas se generan inicialmente a partir de la información disponible sobre el problema, o bien de un modo aleatorio cuando no se dispone de esta información, y la población se hace evolucionar a lo largo del tiempo mediante un proceso de competición y alteración controlada que emula los procesos genéticos que tienen lugar en la naturaleza. 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 AG por el espacio de búsqueda. Por esta razón, debe estar bien diseñada para que sea capaz, no sólo de distinguir de un modo claro los individuos bien adaptados de los que no lo están, sino también de ordenar estos en función de su capacidad para resolver el problema.

La Figura B.3, en la que $P(t)$ denota la población en la generación t , muestra la estructura general de un AG básico.

A continuación, comentaremos brevemente los aspectos básicos relacionados con los AGs, la representación de las soluciones, el mecanismo de selección y los operadores de cruce y mutación.

Procedimiento Algoritmo Genético

EMPEZAR

$t = 0$;

inicializar $P(t)$;

evaluar $P(t)$;

MIENTRAS NO (*condición de parada*) HACER

$t = t + 1$;

seleccionar $P'(t)$ a partir de $P(t - 1)$;

cruzar y mutar $P'(t)$;

$P(t) = P'(t)$;

evaluar $P(t)$;

FIN-MIENTRAS

FIN

Figura B.3.: Estructura básica de un Algoritmo Genético

B.3.1. Representación de las Soluciones

El esquema de representación o codificación es un factor clave en la aplicación de los AGs, ya que estos 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 AG afronta el problema. Existen distintos esquemas generales de codificación entre los que destacan los siguientes:

1. La *codificación binaria*: Es la más antigua de todas las existentes [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 [HLV98b], 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.3.2. El Mecanismo de Selección

El mecanismo de selección es el encargado de seleccionar la población intermedia de individuos la cual, una vez aplicados los operadores de cruce y mutación, formará la nueva población del AG en la siguiente generación. De este modo, si notamos por P la población actual formada por n cromosomas, C_1, \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.4). El número de veces que se copia cada cromosoma depende de su adecuación, por lo que generalmente aquellos que presentan un valor mayor en la función de adaptación suelen tener más oportunidades para contribuir con copias a la formación de P' .

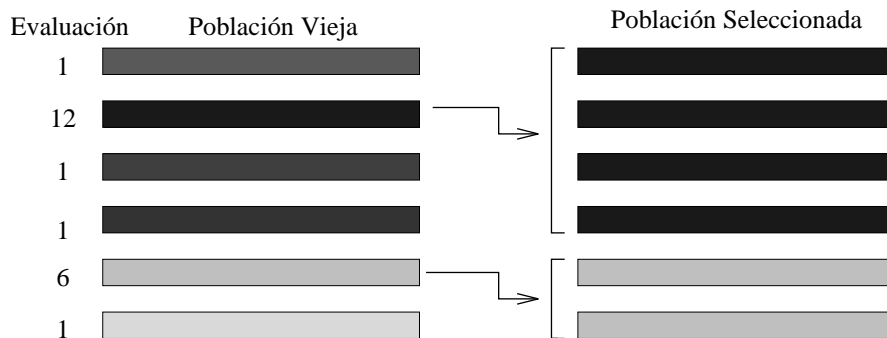


Figura B.4.: Ejemplo de aplicación del mecanismo de selección

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 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.3.3. El Operador de Cruce

Este operador constituye un mecanismo para compartir información entre cromosomas. Combina las características de dos cromosomas padre para obtener dos descendientes, con la posibilidad de que los cromosomas hijo, obtenidos mediante la recombinación de sus padres, estén mejor adaptados que estos. 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 AG. 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 presenten. Como ya hemos comentado, tanto la definición del operador de cruce como la del 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.5), o el *cruce multipunto*, que procede como el anterior pero trabajando sobre dos o más puntos de cruce.

También se pueden emplear ambos operadores cuando se trabaja con el esquema

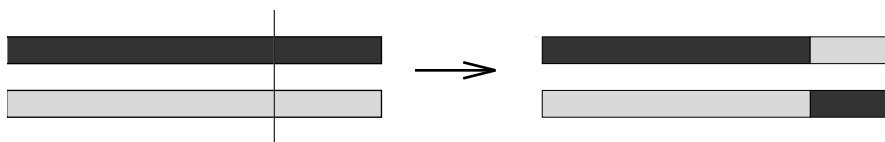


Figura B.5.: Ejemplo de aplicación del operador de cruce simple en un punto

de codificación real, aunque existe una serie de operadores diseñados para su uso específico con esta representación [HLV98b]. Entre estos, destacaremos una familia de operadores que manejan técnicas basadas en lógica difusa para mejorar el comportamiento del operador de cruce [HLV98b]. 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 AG o variable dependiendo de la edad de la población.

B.3.4. El Operador de Mutación

Este segundo operador altera arbitrariamente uno o más genes del cromosoma seleccionado con el propósito de aumentar la diversidad de la población. 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 AGs con codificación binaria se basa en cambiar el valor del bit seleccionado para mutar por su complementario en el alfabeto binario, tal como recoge la Figura B.6.

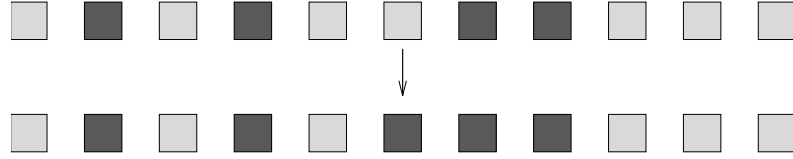


Figura B.6.: Ejemplo de aplicación del operador de mutación

Este operador puede trasladarse al campo de los AGs con codificación real, de forma que el nuevo valor del gen mutado se escoja aleatoriamente dentro del intervalo de definición asociado. Al igual que en el caso del operador de cruce, existen distintos operadores de mutación específicos para trabajar con esta codificación [HLV98b, Mic96]. 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 AG 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.

B.4. Optimización Basada en Colonias de Hormigas

B.4.1. Introducción

Recientemente, ha aparecido una nueva familia de algoritmos bioinspirados, la OCH [BDT99, DC99]. Desde la aparición del modelo inicial, el *Sistema de Hormigas*, aplicado a la resolución del problema del viajante de comercio (PVC) [DMC96], se ha desarrollado una gran cantidad de nuevos modelos cada vez más refinados que han permitido resolver una amplia gama de problemas de optimización (para un resumen de modelos y aplicaciones, véase [BDT99, DC99]).

Los algoritmos de OCH modelan el comportamiento de las colonias de hormigas reales. En concreto, se inspiran en el comportamiento social de estos insectos para proporcionar comida a la colonia. En el proceso de búsqueda de comida, consistente en la localización del alimento y la vuelta al hormiguero, depositan en el suelo una sustancia denominada *feromona*. Las hormigas son capaces de oler la feromona y la colonia dirige la búsqueda en función de esta. Cuando una hormiga llega a una intersección, decide el camino a seguir de un modo probabilístico en función de la cantidad de feromona existente en cada ramificación.

De este modo, las deposiciones de feromona acaban formando un rastro entre el hormiguero y la comida, que las nuevas hormigas pueden seguir. La acción continuada de los miembros de la colonia provoca que la longitud del camino se reduzca progresivamente (la similitud entre este comportamiento y las características del PVC fue la que llevó a escoger dicho problema como la primera aplicación de la OCH). Los caminos más cortos terminan por ser los más frecuentados, con lo que reciben un aporte muy alto de feromona, mientras que los largos dejan de ser visitados y la feromona asociada termina por evaporarse.

Los algoritmos de OCH se basan, por tanto, en la colaboración entre múltiples agentes, hormigas, cada uno de los cuales genera una solución posible al problema en cada iteración del algoritmo. Para ello, cada hormiga recorre un grafo que representa el caso concreto del problema y hace uso de dos tipos de información, comunes a toda la colonia, que especifican la preferencia de los arcos del grafo en cada instante:

- *Información heurística*, la cual depende del caso concreto del problema, se calcula antes de la ejecución del algoritmo y se mantiene constante a lo largo de esta. El valor de dicha información para el arco (i, j) se nota por η_{ij} .
- *Información memorística*, que se va modificando a lo largo de la ejecución del algoritmo en función de las hormigas que recorrieron cada arco en el pasado y de la bondad de las soluciones que generaron. Se suele representar en forma de una matriz τ que modela la feromona que depositan las hormigas reales en sus recorridos. La feromona asociada al arco (i, j) se nota por τ_{ij} .

El funcionamiento básico es el siguiente: en cada iteración, una población de m hormigas construye progresivamente —según una *regla probabilística de transición de estados* que depende de la información existente— distintos recorridos por el grafo (soluciones al problema). Una vez transcurrido este proceso, todos los arcos sufren una evaporación de feromona —tal como ocurre en el mundo real—, pero aquellos arcos presentes en las soluciones más prometedoras son reforzados con un aporte adicional de feromona directamente proporcional a la bondad de la solución. Esta última tarea se lleva a cabo mediante la *regla de actualización global de feromona*.

Las distintas variantes de algoritmos OCH existentes dependen de la forma en que se implementen las dos reglas anteriores, así como de la introducción de nuevas componentes. Por otro lado, con objeto de mejorar la precisión de las soluciones obtenidas por la OCH, y sin ningún paralelismo con la naturaleza, es común que se aplique un algoritmo de búsqueda local para mejorar la solución obtenida por cada hormiga antes de realizar la actualización global de feromona [BHS99, MC99, SH00].

Para resolver un problema mediante un algoritmo de OCH, es necesario llevar a cabo los siguientes pasos:

1. Obtener una representación del problema en forma de grafo o con una estructura tal que pueda ser recorrida por las hormigas para obtener soluciones.
2. Determinar la forma de asignar una preferencia heurística a cada decisión que ha de tomar la hormiga en cada paso para generar su solución, es decir, definir la información heurística η_{ij} asociada a cada arco.
3. Buscar una forma adecuada para especificar los valores iniciales de feromona τ_0 .
4. Definir una función objetivo a optimizar.

5. Escoger un algoritmo de OCH concreto y aplicarlo al problema en cuestión considerando los parámetros anteriores.

Dedicaremos las dos secciones siguientes a los dos primeros algoritmos de OCH que se propusieron en la literatura especializada. En [BDT99, DC99] pueden encontrarse descripciones de otros modelos.

B.4.2. El Sistema de Hormigas

El *sistema de hormigas* (SH) [DMC96] implementa el comportamiento general descrito en la sección anterior del siguiente modo. Dada una hormiga k situada en el nodo i , la *regla de transición de estados* escoge aleatoriamente el nodo destino j en función de la distribución de probabilidad

$$p_k(i, j) = \begin{cases} \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{u \in J_k(i)} [\tau_{iu}]^\alpha \cdot [\eta_{iu}]^\beta}, & \text{si } j \in J_k(i) \\ 0, & \text{en otro caso} \end{cases},$$

donde τ_{ij} es el nivel de feromona del arco (i, j) , η_{ij} es la información heurística — por ejemplo, en el caso del PVC, $\eta_{ij} = \frac{1}{l_{ij}}$, siendo l_{ij} la longitud del arco (i, j) —, $J_k(i)$ es el conjunto de nodos permitidos no visitados aún por la hormiga k , y α y β son parámetros que especifican la importancia relativa de la información memorística (feromona) con respecto a la información heurística.

Por otro lado, la actualización global de feromona se aplica según la expresión

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k,$$

en la que

$$\Delta\tau_{ij}^k = \begin{cases} f(C(S_k)), & \text{si } (i, j) \in S_k \\ 0, & \text{en otro caso} \end{cases},$$

donde $\rho \in [0, 1]$ es el parámetro de evaporación de feromona, m es el número de hormigas considerado, S_k es la solución construida por la hormiga k y $f(C(S_k))$ es el aporte de feromona que depende del costo de dicha solución, $C(S_k)$ (por ejemplo, en el PVC, $f(C(S_k)) = \frac{1}{C(S_k)}$). Como puede observarse, todas las hormigas aportan feromona a los arcos de sus soluciones en función de la bondad de estas. Aquellos arcos que no forman parte de ninguna solución, sufren la evaporación de un ρ por uno de la feromona de la que disponían.

B.4.3. El Sistema de Colonias de Hormigas

El *sistema de colonias de hormigas* (SCH) [DG97] extiende a su predecesor, el SH, en tres aspectos principales:

- Presenta una nueva regla de transición de estados que establece un balance entre la exploración y la explotación de la información disponible. El nodo destino j se escoge de la forma

$$j = \begin{cases} \arg \max_{u \in J_k(i)} \{[\tau_{iu}]^\alpha \cdot [\eta_{iu}]^\beta\}, & \text{si } q < q_0 \\ S, & \text{en otro caso} \end{cases},$$

donde q es uniforme generado aleatoriamente en $[0, 1]$, $q_0 \in [0, 1]$ es la probabilidad con que se escoge determinísticamente el arco más prometedor (explotación) y S es un nodo aleatorio seleccionado según la regla de transición del SH (exploración dirigida).

- La regla de actualización global también varía. El aporte de feromona pasa a ser realizado por una sola hormiga, aquella que generó la mejor solución encontrada hasta el momento, S_{global} :

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \rho \cdot \Delta\tau_{ij},$$

donde

$$\Delta\tau_{ij} = \begin{cases} f(C(S_{global})), & \text{si } (i, j) \in S_{global} \\ 0, & \text{en otro caso} \end{cases}.$$

- Cada vez que una hormiga recorre un arco, se produce también una actualización de la feromona, empleando la *regla de actualización local*:

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \rho \cdot \Delta\tau_{ij},$$

habitualmente, $\Delta\tau_{ij} = \tau_0$.

Este hecho provoca diversidad al reducir la preferencia del arco para las siguientes hormigas que lo visiten en la iteración actual.

C. Consideraciones en la Experimentación Realizada

Siempre que enseñes, enseña a la vez a dudar de lo que enseñas.

JOSÉ ORTEGA Y GASSET (1883–1955),
FILÓSOFO Y ENSAYISTA ESPAÑOL

Este apéndice contiene una descripción de los aspectos de experimentación considerados y aplicaciones de modelado empleadas en los diferentes estudios experimentales desarrollados a lo largo de esta memoria.

C.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 (v.gr., un esquema de nichos en AGs). 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.

- *BD inicial* — En todos los métodos de aprendizaje que precisen la definición de una BD inicial, bien porque sólo deriven la BR o bien porque necesiten esta información para derivar otra BD (v.gr., para determinar los intervalos de variación de cada parámetro que define las funciones de pertenencia), se empleará la misma BD. Estará constituida por particiones difusas uniformemente distribuidas en cada variable como se muestra en la figura 2.1 (página 62).

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 y el universo de discurso de cada variable dependerá del problema empleado, como se indica en la siguiente sección.

- *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.4.4) [CHP97].

La única excepción la conforma el método de aprendizaje de Nozaki *et al.* [NIT97], que precisa de un método de razonamiento particular para considerar adecuadamente el empleo de dos BCs distintas.

C.2. Descripción de las Aplicaciones Consideradas

La tabla C.1 recoge un resumen de las principales características de las seis aplicaciones empleadas a lo largo de esta memoria para los estudios experimentales. A continuación describiremos los problemas considerados.

C.2.1. Modelado de la Función Tridimensional F_1

El objetivo de este primer problema será modelar la superficie de la función tridimensional sencilla F_1 presentada en la figura C.1 [HLV98a]. Para realizar esto, se ha obtenido experimentalmente un conjunto de entrenamiento uniformemente distribuido en el espacio de definición tridimensional. Así, se ha generado un conjunto con 1.681 valores tomando 41 valores para cada uno de las dos variables de entrada considerados tomados uniformemente en los correspondientes intervalos.

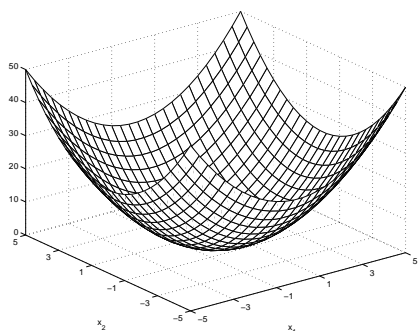
Tabla C.1.: Principales características de las aplicaciones consideradas en esta memoria

Aplicación	Complejidad	#V	#Entr	#Prue	#TL	#P
Función tridimensional F_1	baja	2	1.681	168	7	1
Función tridimensional F_2	media-baja	2	674	67	7	1
Función tridimensional F_3	muy alta	2	1.681	168	7	1
Evaluación del arroz	media	5	75	30	2	10
Longitud de líneas de baja tensión	alta	2	396	99	7	1
Costo de líneas de media tensión	media-alta	4	847	212	5	1

#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, #P = número de particiones de ejemplos consideradas

Se ha generado otro conjunto de datos para su uso como conjunto de prueba para validar el comportamiento de los modelos obtenidos, evitando cualquier posible sesgo relacionado con los datos de entrenamiento. El tamaño de este conjunto de datos (168) es el diez por ciento del de entrenamiento. Los datos se han obtenido generando valores aleatorios dentro de cada universo de discurso de las variables de entrada, y calculando el valor de la variable de salida.

En la experimentación realizada consideraremos siete términos lingüísticos en cada una de las particiones difusas de las distintas variables de entrada y salida.



$$F_1(x_1, x_2) = x_1^2 + x_2^2$$

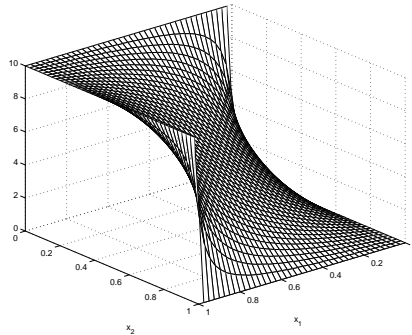
$$x_1, x_2 \in [-5, 5],$$

$$F_1(x_1, x_2) \in [0, 50]$$

Figura C.1.: Representación gráfica, expresión matemática, y universos de discurso de las variables de la función tridimensional F_1

C.2.2. Modelado de la Función Tridimensional F_2

Consideraremos también una función tridimensional algo más compleja, F_2 [CH97]. En la figura C.2 se muestra su representación gráfica, expresión matemática y universos de discurso de las variables. Esta función presenta discontinuidades en $(0,0)$ y $(1,1)$, como puede observarse en la representación gráfica. Los datos de la función se han obtenido de forma similar al proceso anteriormente descrito. Se han generado los conjuntos de entrenamiento y prueba con 674 (tomando 26 valores para cada una de las dos variables de entrada exceptuando los dos puntos del espacio donde F_2 no está definida) y 67 (diez por ciento) valores, respectivamente. Se usarán siete términos lingüísticos en cada variable.



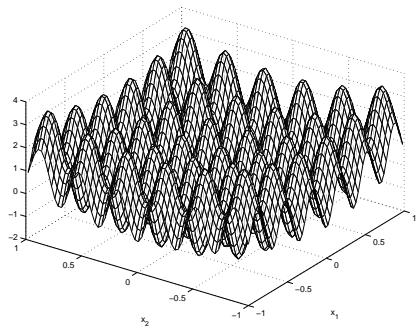
$$F_2(x_1, x_2) = \frac{10 \cdot (x_1 - x_1 x_2)}{x_1 - 2x_1 x_2 + x_2}$$

$$x_1, x_2 \in [0, 1],$$
$$F_2(x_1, x_2) \in [0, 10]$$

Figura C.2.: Representación gráfica, expresión matemática y universos de discurso de las variables de la función tridimensional F_2

C.2.3. Modelado de la Función Tridimensional F_3

Como función compleja para estudiar el comportamiento comparativo entre el ML y el MD, consideraremos la *función generalizada de Rastrigin*, F_3 [HLV98a], que es fuertemente multimodal. La figura C.3 muestra su representación gráfica, expresión matemática y universos de discurso de las variables. Los datos de la función se han obtenido, de nuevo, con un proceso similar a los anteriores. El número de ejemplos de entrenamiento y prueba será similar a la función F_1 , con 1.681 y 168 valores respectivamente. Emplearemos siete términos lingüísticos en cada variable.



$$F_3(x_1, x_2) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2)$$

$$x_1, x_2 \in [-1, 1],$$

$$F_3(x_1, x_2) \in [-2, 3,382745]$$

Figura C.3.: Representación gráfica, expresión matemática y universos de discurso de las variables de la función tridimensional F_3

C.2.4. Estimación de la Calidad del Arroz

La valoración subjetiva de la calidad de las comidas es un problema muy importante pero, a la vez, muy difícil. En el caso de la valoración de la calidad del arroz, el análisis de su sabor se suele llevar a cabo mediante un tipo de valoración subjetiva denominada *prueba sensorial*. En esta prueba, un grupo de expertos, habitualmente formado por 24 personas, proceden a dar su evaluación sobre una serie de características asociadas a cada tipo de arroz.

Dichas características son las siguientes: *sabor*, *aspecto*, *gusto*, *pegajosidad*, *dureza* y *valoración global del tipo de arroz*. Debido a la gran cantidad de variables relevantes, el problema de la valoración de la calidad del arroz se hace muy complejo, lo que justifica la necesidad de afrontar un modelado que permita determinar la relación no lineal existente en el mismo.

Para ello, emplearemos el conjunto de datos extraído de [NIT97]. Dicho conjunto está compuesto por 105 vectores de datos que recogen valoraciones subjetivas de las seis variables comentadas efectuadas por expertos sobre otros tantos tipos de arroz cultivados en Japón (como por ejemplo el Sananishiki, el Akita-Komachi, etc.). El sistema a modelar está basado, por tanto, en cinco variables de entrada, las cinco características comentadas, y una de salida, la valoración global del tipo de arroz estudiado. Las seis variables están normalizadas, con lo que toman valores en el intervalo real $[0, 1]$.

Vamos a proceder del mismo modo en que los autores hacen en [INT⁺94, NIT97]. En lugar de realizar una única partición del conjunto de datos original en un subconjunto de entrenamiento y otro de prueba, obtendremos varias particiones distintas de

un modo aleatorio y consideraremos dichas particiones para generar distintos modelos lingüísticos en cada experimento. Al actuar de este modo evitamos que el comportamiento del proceso de diseño esté sesgado por la partición escogida, caso que podría darse debido a la dimensión tan pequeña que presenta el conjunto de datos del que se dispone. Vamos a generar diez particiones distintas compuestas por 75 datos para entrenamiento y 30 para prueba. El tamaño empleado coincide con el que usaron originalmente los autores. Consideraremos dos términos lingüísticos en cada variable.

El objetivo de este problema real consistía no sólo en obtener la estimación más exacta posible, sino también en que el modelo obtenido finalmente fuera interpretable. Según esto sólo sería posible utilizar modelos lingüísticos, sin embargo, será utilizado también en el MD (capítulo 1) para permitir comprobar cómo reaccionan los sistemas aproximativos en casos como este.

C.2.5. Problemas de Distribución de Energía Eléctrica

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, que 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 cierto grado.

C.2.5.1. 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, Sán97]. 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.

Tabla C.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]

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 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 [Sán97, Sán00]. En la tabla C.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. Consideraremos siete términos lingüísticos en cada variable.

C.2.5.2. 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 C.3). Para ello disponemos de estimaciones de costos de mantenimiento basa-

C. Consideraciones en la Experimentación Realizada

das 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. Emplearemos cinco términos lingüísticos en cada variable.

Tabla C.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]

Bibliografía

- [ACC⁺00] Alcalá R., Casillas J., Cordón O., Herrera F. y Zwir S. J. 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, capítulo 29, páginas 889–941. Academic Press, San Diego, CA, EE. UU.
- [ACCH99] Alcalá R., Casillas J., Cordón O. y Herrera F. (1999) Building fuzzy graphs: features and taxonomy of learning for non-grid-oriented fuzzy rule-based systems. Informe Técnico #DECSAI-99-01-17, Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada, Granada, España. Disponible en <http://decsai.ugr.es/~casillas/>.
- [AT92] Alsina C. y Trillas E. (1992) Synthesizing implications. *International Journal of Intelligent Systems* 7: 705–713.
- [Bab98] Babuška R. (1998) *Fuzzy modeling for control*. Kluwer Academic, Norwell, MA, EE. UU.
- [Bak87] Baker J. E. (1987) Reducing bias and inefficiency in the selection algorithm. En *Proceedings of the 2nd International Conference on Genetic Algorithms*, páginas 14–21. Lawrence Erlbaum, 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* 2(4): 463–484.
- [BD95] Bárdossy A. y Duckstein L. (1995) *Fuzzy rule-based modeling with application to geophysical, biological and engineering systems*. CRC Press, Boca Raton, FL, EE. UU.
- [BDT99] Bonabeau E., Dorigo M. y Theraulaz G. (1999) *Swarm intelligence. From natural to artificial systems*. Oxford University Press, Oxford, Reino Unido.
- [BHS99] Bullnheimer B., Hartl R. F. y Strauss C. (1999) A new rank based version of the ant system: a computational study. *Central European Journal for Operations Research and Economics* 7(1): 25–38.
- [BKC96] Bonissone P. P., Khedkar P. S. y Chen Y. (1996) Genetic algorithms for automated tuning of fuzzy controllers: a transportation application. En *Proceedings of the 5th IEEE International Conference on Fuzzy Systems*, páginas 674–680. Nueva Orleans, LA, EE. UU.

- [BS91] Bäck T. y Schwefel H. P. (1991) Extended selection mechanism in genetic algorithms. En *Proceedings of the 4th International Conference on Genetic Algorithms*, páginas 2–9. San Diego, CA, EE. UU.
- [CCdJH01a] Casillas J., Cordón O., del Jesus M. J. y Herrera F. (2001) Genetic feature selection in a fuzzy rule-based classification system learning process for high dimensional problems. Por aparecer en *Information Sciences*. Versión preliminar disponible en <http://decsai.ugr.es/~casillas/>.
- [CCdJH01b] Casillas J., Cordón O., del Jesus M. J. y Herrera F. (2001) Genetic tuning of fuzzy rule-based systems integrating linguistic hedges. En *Proceedings of the 9th IFSA World Congress and 20th NAFIPS International Conference*. Vancouver, Canadá. Por aparecer.
- [CCdJH01c] Casillas J., Cordón O., del Jesus M. J. y Herrera F. (2001) Genetic tuning of fuzzy rule deep structures for linguistic modeling. Informe Técnico #DECSAI-01-01-02, Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada, Granada, España. Disponible en <http://decsai.ugr.es/~casillas/>.
- [CCH00a] Casillas J., Cordón O. y Herrera F. (2000) Can linguistic modeling be as accurate as fuzzy modeling without losing its description to a high degree? Informe Técnico #DECSAI-00-01-20, Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada, Granada, España. Disponible en <http://decsai.ugr.es/~casillas/>.
- [CCH00b] Casillas J., Cordón O. y Herrera F. (2000) Generación de reglas difusas con buen nivel de cooperación mediante algoritmos de hormigas. En *Actas del X Congreso Español Sobre Tecnología y Lógica Fuzzy*, páginas 291–296. Sevilla, España.
- [CCH00c] Casillas J., Cordón O. y Herrera F. (2000) Improving the Wang and Mendel's fuzzy rule learning method by inducing cooperation among rules. En *Proceedings of the 8th Information Processing and Management of Uncertainty in Knowledge-Based Systems Conference*, páginas 1682–1688. Madrid, España.
- [CCH00d] Casillas J., Cordón O. y Herrera F. (2000) Learning cooperative fuzzy linguistic rules using ants colony algorithms. Informe Técnico #DECSAI-00-01-19, Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada, Granada, España. Disponible en <http://decsai.ugr.es/~casillas/>.
- [CCH00e] Casillas J., Cordón O. y Herrera F. (2000) Learning fuzzy rules using ants colony algorithms. En *Proceedings of the 2nd International Workshop on Ant Algorithms*, páginas 13–21. Bruselas, Bélgica.
- [CCH00f] Casillas J., Cordón O. y Herrera F. (2000) A methodology to improve ad hoc data-driven linguistic rule learning methods by inducing cooperation among rules. Informe Técnico #DECSAI-00-01-01, Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada, Granada, España. Disponible en <http://decsai.ugr.es/~casillas/>.

-
- [CCH01] Casillas J., Cordon O. y Herrera F. (2001) Different approaches to induce cooperation in fuzzy linguistic models under the COR methodology. En Bouchon-Meunier B., Gutiérrez-Ríos J., Magdalena L. y Yager R. R. (Eds.) *Techniques for constructing intelligent systems*. Springer-Verlag, Heidelberg, Alemania. En impresión.
- [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: 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. *International Journal of Intelligent Systems* 14(11): 1123–1153.
- [CDM99] Castro J. L., Delgado M. y Mantas C. J. (1999) A hybrid system for approximate reasoning. Informe técnico, Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada. Sometido en *IEEE Transactions on Systems, Man, and Cybernetics. Part A: Systems and Humans*.
- [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.
- [CH97] Cordon 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] Cordon O. y Herrera F. (2000) A proposal for improving the accuracy of linguistic modeling. *IEEE Transactions on Fuzzy Systems* 8(3): 335–344.
- [CH01] 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.
- [CHHM01] Cordon O., Herrera F., Hoffmann F. y Magdalena L. (2001) *Genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge bases*. World Scientific, Singapur, Singapur. En impresión.
- [CHP97] Cordon O., Herrera F. y Peregrín A. (1997) Applicability of the fuzzy operators in the design of fuzzy logic controllers. *Fuzzy Sets and Systems* 86(1): 15–41.
- [CHS99] Cordon O., Herrera F. y Sánchez L. (1999) Solving electrical distribution problems using hybrid evolutionary data analysis techniques. *Applied Intelligence* 10(1): 5–24.
- [CHV01] 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. Por aparecer en *IEEE Transactions on Fuzzy Systems*. Versión preliminar disponible en <http://decsai.ugr.es/~herrera/>.
- [CK89] Cao Z. y Kandel A. (1989) Applicability of some fuzzy implication operators. *Fuzzy Sets and Systems* 31: 151–186.

- [CP00] Cho J.-S. y Park D.-J. (2000) Novel fuzzy logic control based on weighting of partially inconsistent rules using neural network. *Journal of Intelligent and Fuzzy Systems* 8: 99–110.
- [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.
- [CYP96] Chi Z., Yan H. y Pham T. (1996) *Fuzzy algorithms with application to image processing and pattern recognition*. World Scientific, Singapur, Singapur.
- [DC99] Dorigo M. y Caro G. D. (1999) The ant colony optimization meta-heuristic. En Corne D., Dorigo M. y Glover F. (Eds.) *New ideas in optimization*, páginas 11–32. McGraw-Hill, Nueva York, NY, EE. UU.
- [DG97] Dorigo M. y Gambardella L. M. (1997) Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation* 1(1): 53–66.
- [DGM97] Delgado M., Gómez-Skarmeta A. F. y Marín F. (1997) A fuzzy clustering based rapid-prototyping for fuzzy rule-based modeling. *IEEE Transactions on Fuzzy Systems* 5(2): 223–233.
- [DHR93] Driankov D., Hellendoorn H. y Reinfrank M. (1993) *An introduction to fuzzy control*. Springer-Verlag, Heidelberg, Alemania.
- [DMC96] Dorigo M., Maniezzo V. y Colorni A. (1996) The ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics. Part B: Cybernetics* 26(1): 29–41.
- [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.
- [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.
- [Ful00] Fullér R. (2000) *Introduction to neuro-fuzzy systems*. Springer-Verlag, Heidelberg, Alemania.
- [GJ99] Gómez-Skarmeta A. F. y Jiménez F. (1999) Fuzzy modeling with hybrid systems. *Fuzzy Sets and Systems* 104(2): 199–208.
- [GL97] Glover F. y Laguna M. (1997) *Tabu search*. Kluwer Academic, Norwell, MA, EE. UU.
- [Glo96] Glorennec P. (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, Heidelberg, Alemania.

-
- [Gol89] Goldberg D. E. (1989) *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, Reading, MA, EE. UU.
- [Gol98] Goldberg D. E. (1998) The race, the hurdle, and the sweet spot: lessons from genetic algorithms for the automation of design innovation and creativity. Informe Técnico Illigal 98007, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, Illinois, IL, EE. UU. Disponible en <http://www-illigal.ge.uiuc.edu/>.
- [Gol99] Goldberg D. E. (1999) Genetic and evolutionary algorithms in the real world. Informe Técnico Illigal 99013, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, Illinois, IL, EE. UU. Disponible en <http://www-illigal.ge.uiuc.edu/>.
- [Gol00] Goldberg D. E. (2000) A meditation on the computational intelligence and its future. Informe Técnico Illigal 2000019, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, Illinois, IL, EE. UU. Disponible en <http://www-illigal.ge.uiuc.edu/>.
- [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.
- [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.
- [GT95] Goonatilake S. y Treleaven P. (Eds.) (1995) *Intelligent systems for finance and bussiness*. John Willey & Sons, Nueva York, NY, EE. UU.
- [Hir93] Hirota K. (Ed.) (1993) *Industrial applications of fuzzy technology*. Springer-Verlag, Heidelberg, Alemania.
- [HLV95] Herrera F., Lozano M. y Verdegay J. L. (1995) Tuning fuzzy controllers by genetic algorithms. *International Journal of Approximate Reasoning* 12: 299–315.
- [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 behavioural analysis. *Artificial Intelligence Review* 12: 265–319.
- [Hol75] Holland J. H. (1975) *Adaptation in natural and artificial systems*. Ann arbor: The University of Michigan Press, Michigan, MI, EE. UU.
- [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, Heidelberg, Alemania.

- [IM96] Ishibuchi H. y Murata T. (1996) A genetic-algorithm-based fuzzy partition method for pattern classification problems. En Herrera F. y Verdegay J. L. (Eds.) *Genetic algorithms and soft computing*, páginas 555–578. Physica-Verlag, Heidelberg, Alemania.
- [INT⁺94] Ishibuchi H., Nozaki K., Tanaka H., Hosaka Y. y Matsuda M. (1994) Empirical study on learning in fuzzy systems by rice taste analysis. *Fuzzy Sets and Systems* 64: 129–144.
- [INYT95] Ishibuchi H., Nozaki K., Yamamoto N. y Tanaka H. (1995) Selecting fuzzy if-then rules for classification problems using genetic algorithms. *IEEE Transactions on Fuzzy Systems* 3(3): 260–270.
- [Jan93] Jang J.-S. R. (1993) Anfis: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics* 23(3): 665–685.
- [Kar91] Karr C. L. (1991) Genetic algorithms for fuzzy controllers. *AI Expert* 6(2): 26–33.
- [Kir84] Kirkpatrick S. (1984) Optimization by simulated annealing — quantitative studies. *Journal of Statistical Physics* 34: 975–986.
- [KKK94] Kinzel J., Klawonn F. y Kruse R. (1994) Modifications of genetic algorithms for designing and optimizing fuzzy controllers. En *Proceedings of the 1st IEEE International Conference on Evolutionary Computation*, páginas 28–33. Orlando, FL, EE. UU.
- [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 P. 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.
- [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.
- [Lak73] Lakoff G. (1973) Hedges: a study in meaning criteria and the logic of fuzzy concepts. *Journal of Philosophical Logic* 2: 458–508.
- [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 fuzzy logic controller. *International Journal of Man-Machine Studies* 7: 1–13.
- [Mam74] Mamdani E. H. (1974) Applications of fuzzy algorithms for control a simple dynamic plant. *Proceedings of the IEE* 121 12: 1585–1588.
- [MC99] Maniezzo V. y Colomi A. (1999) The ant system applied to the quadratic assignment problem. *IEEE Transactions on Knowledge and Data Engineering* 11(5): 769–778.
- [MF00] Michalewicz Z. y Fogel D. B. (2000) *How to solve it: modern heuristics*. Springer, Heidelberg, Alemania.
- [Mic96] Michalewicz Z. (1996) *Genetic algorithms + data structures = evolution programs*. Springer-Verlag, Heidelberg, Alemania, tercera edición.
- [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.
- [MSG00] Marín-Blázquez J. G., Shen Q. y Gómez-Skarmeta A. F. (2000) From approximative to descriptive models. En *Proceedings of the 9th IEEE International Conference on Fuzzy Systems*, páginas 829–834. San Antonio, TX, EE. UU.
- [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(3): 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.
- [NKK97] Nauck D., Klawonn F. y Kruse R. (1997) *Foundations of neuro-fuzzy systems*. John Wiley & Sons, Nueva York, NY, EE. UU.
- [Nov96] Novák V. (1996) A horizon shifting model of linguistic hedges for approximate reasoning. En *Proceedings of the 5th IEEE International Conference on Fuzzy Systems*, páginas 423–427. Nueva Orleans, FL, EE. UU.
- [Par95] Paredis J. (1995) Coevolutionary computation. *Artificial Life* 2: 355–375.
- [Ped96] Pedrycz W. (Ed.) (1996) *Fuzzy modelling: paradigms and practice*. Kluwer Academic, Norwell, MA, EE. UU.
- [Ped97] Pedrycz W. (1997) *Fuzzy evolutionary computation*. Kluwer Academic, Norwell, MA, EE. UU.

- [PGG97] Pedrycz W., Gudwin R. R. y Gomide F. A. C. (1997) Nonlinear context adaptation in the calibration of fuzzy sets. *Fuzzy Sets and Systems* 88: 91–97.
- [PJ00] Potter M. A. y Jong K. A. D. (2000) Cooperative coevolution: an architecture for evolving coadapted subcomponents. *Evolutionary Computation* 8(1): 1–29.
- [PP99] Pal N. R. y Pal K. (1999) Handling of inconsistent rules with an extended model of fuzzy reasoning. *Journal of Intelligent and Fuzzy Systems* 7: 55–73.
- [PS01] Peña-Reyes C. A. y Sipper M. (2001) Fuzzy CoCo: a cooperative coevolutionary approach to fuzzy modeling. Por aparecer en *IEEE Transactions on Fuzzy Systems*. Versión preliminar disponible en <http://lslwww.epfl.ch/~penha/>.
- [RB97] Rosin C. D. y Belew R. K. (1997) New methods for competitive coevolution. *Evolutionary Computation* 5(1): 1–29.
- [Sán97] Sánchez L. (1997) Study of the asturias rural and urban low voltage network. Informe técnico, Hidroeléctrica del Cantábrico, Departamento de Investigación y Desarrollo, Asturias, España.
- [Sán00] Sánchez L. (2000) Interval-valued GA-P algorithms. *IEEE Transactions on Evolutionary Computation* 4(1): 64–72.
- [SH00] Stützle T. y Hoos H. (2000) Max-min ant system. *Future Generation Computer Systems* 8: 889–914.
- [SK88] Sugeno M. y Kang G. T. (1988) Structure identification of fuzzy model. *Fuzzy Sets and Systems* 28: 15–33.
- [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.
- [Sor91] Sorkin G. (1991) Efficient simulated annealing on fractal energy landscapes. *Algorithmica* 6: 367–418.
- [SR00] Setnes M. y Roubos H. (2000) GA-fuzzy modeling and classification: complexity and performance. *IEEE Transactions on Fuzzy Systems* 8(5): 509–522.
- [SY93] Sugeno M. y Yasukawa T. (1993) A fuzzy-logic-based approach to qualitative modeling. *IEEE Transactions on Fuzzy Systems* 1(1): 7–31.
- [Tho15] Thorburn W. M. (1915) Occam's razor. *Mind* 24: 287–288.
- [Tho18] Thorburn W. M. (1918) The myth of occam's razor. *Mind* 27: 345–353.
- [Thr91] Thrift P. (1991) Fuzzy logic synthesis with genetic algorithms. En Belew R. K. y Booker L. B. (Eds.) *Proceedings of the 4th International Conference on Genetic Algorithms*, páginas 509–513. Morgan Kaufmann Publishers, 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: 116–132.
- [TV85a] Trillas E. y Valverde L. (1985) On implications and indistinguishability in the setting of fuzzy logic. En Kracprzyk J. y Yager R. (Eds.) *Managements decision support systems using fuzzy logic and possibility theory*, páginas 198–212. Verlag TUV Rheinland, Cologne, Alemania.
- [TV85b] Trillas E. y Valverde L. (1985) One mode and implication in approximate reasoning. En Gupta M. M., Kandel A., Bandler W. y Kiszka J. B. (Eds.) *Approximate reasoning in expert systems*. North-Holland, Amsterdam, Holanda.
- [Val99] 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.
- [vLA87] van Laarhoven P. J. M. y Aarts E. H. L. (1987) *Simulated annealing: theory and applications*. Kluwer Academic, Norwell, MA, EE. UU.
- [Wan94] Wang L. X. (1994) *Adaptive fuzzy systems and control, design and stability analysis*. Prentice Hall, Englewood Cliffs, NJ, EE. UU.
- [WHT98] Wang C., Hong T. y Tseng S. (1998) Integrating fuzzy knowledge by genetic algorithms. *IEEE Transactions on Evolutionary Computation* 2(4): 138–149.
- [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.
- [YB94] Yu W. y Bien Z. (1994) Design of fuzzy logic controller with inconsistent rule base. *Journal of Intelligent and Fuzzy Systems* 2: 147–159.
- [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 desision processes. *IEEE Transactions on Systems, Man, and Cybernetics* 1: 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 application to approximate reasoning. Partes I, II y III *Information Science* 8, 8, 9: 199–249, 301–357, 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.
- [Zad94] Zadeh L. A. (1994) Soft computing and fuzzy logic. *IEEE Software* 11(6): 48–56.
- [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.
- [Zim96] Zimmermann H.-J. (1996) *Fuzzy set theory and its application*. Kluwer Academic, Norwell, MA, EE. UU., tercera edición.

Índice de Autores

- Aarts, E. H. L., 185
Alcalá, R., 19, 21, 30, 32
Alsina, C., 181
Assilian, S., 15, 174
Babuška, R., 11, 13
Bäck, T., 194
Baker, J. E., 79, 120, 194
Bárdossy, A., 14, 20, 25, 31, 37, 182, 183
Bastian, A., 17, 21, 176
Belew, R. K., 142
Bien, Z., 26
Bonabeau, E., 80, 197, 199
Bonissone, P. P., 22, 23, 28, 115, 130, 141
Bullheimer, B., 99, 198
Cao, Z., 181
Carse, B., 19, 21, 31, 38, 173, 176
Casillas, J., 19, 21, 30, 32, 164, 165, 168
Castro, J. L., 60
Chen, Y., 22, 23, 28, 115, 130, 141
Chi, Z., 25
Cho, J.-S., 26
Colorni, A., 80, 82, 99, 197–199
Cooper, M. G., 19, 173
Cordón, O., 19–24, 26, 28–35, 37, 59–61, 65, 101, 115, 116, 122, 141, 164, 165, 168, 181–183, 202, 204, 206
Delgado, M., 19, 60
del Jesus, M. J., 22–24, 34, 37, 115, 116, 165, 168
De Jong, K. A., 140, 142
Di Caro, G., 197, 199
Dorigo, M., 80, 82, 97, 99, 197, 199, 200
Driankov, D., 179, 180, 182, 183
Duckstein, L., 14, 20, 25, 31, 37, 182, 183
Espinosa, J., 23, 119, 141
Fogarty, T. C., 19, 21, 31, 38, 173, 176
Fogel, D. B., 9
Fritzke, B., 174
Fullér, R., 59
Gambardella, L. M., 97, 99, 200
Glorennec, P., 23, 141
Glover, F., 189
Goldberg, D. E., 140, 189, 192–194
Gómez-Skarmeta, A. F., 19, 117, 141
Gomide, F. A. C., 24
González, A., 22–24, 115, 116
Goonatilake, S., 14
Goser, K., 22, 23, 115

- Gudwin, R. R., 24
Gupta, M. M., 181
- Hartl, R. F., 99, 198
Hellendoorn, H., 179, 180, 182, 183
Herrera, F., 19–24, 26, 28–35, 37, 38,
59–61, 65, 101, 115, 116, 122,
124, 141, 164, 165, 168, 181–
183, 192, 195, 196, 202, 204,
206
- Hirota, K., 14
Hoffmann, F., 59
Holland, J. H., 189, 192
Hong, T., 23, 141
Hoos, H., 99, 198
Hosaka, Y., 59, 61, 65, 205
- Ishibuchi, H., 22–24, 26, 31, 35, 59,
61, 65, 83, 101, 141, 202, 205
- Jang, J.-S. R., 115
Jiménez, F., 141
- Kandel, A., 181
Kang, G. T., 16
Kanstein, A., 22, 23, 115
Karr, C. L., 22, 23, 28, 115, 141
Khedkar, P. S., 22, 23, 28, 115, 130,
141
Kinzel, J., 141
Kirkpatrick, S., 77, 188
Kiszka, J., 181
Klawonn, F., 59, 141
Kochanska, M., 181
Krause, P., 101
KrishnaKumar, K., 23, 115, 141
Krone, A., 101
- Kruse, R., 29, 59, 141
- Laguna, M., 189
Lakoff, G., 117
Liska, J., 22, 23, 28, 31, 33, 141
López, S., 23
Lozano, M., 19, 34, 35, 37, 38, 101,
115, 124, 141, 192, 195, 196,
202, 204
- Magdalena, L., 23, 59, 141
Mamdani, E. H., 15, 174, 181, 182
Maniezzo, V., 80, 82, 99, 197–199
Mantas, C. J., 60
Marín, F., 19
Marín-Blázquez, J. G., 117
Matsuda, M., 59, 61, 65, 205
Melsheimer, S. S., 22, 23, 28, 31, 33,
141
Mendel, J. M., 22, 31, 32, 59, 61, 63,
115
Michalewicz, Z., 9, 124, 141, 189, 193,
194, 196
Mitra, S., 25
Monasterio-Huelin, F., 23, 141
Müller, B., 13
Munro, A., 19, 21, 31, 38, 173, 176
Murata, T., 23, 141
- Nauck, D., 29, 59
Novák, V., 117
Nozaki, K., 22, 24, 26, 31, 35, 59, 61,
65, 83, 101, 202, 205
- Pal, K., 22, 26
Pal, N. R., 22, 26
Pal, S. K., 25

- Paredis, J., 141
Park, D.-J., 26
Pedrycz, W., 14, 24, 59
Peña-Reyes, C. A., 141–143
Peregrín, A., 26, 181–183, 202
Pérez, R., 22–24, 115, 116
Pham, T., 25
Potter, M. A., 140, 142

Qi, J., 181

Reinfrank, M., 179, 180, 182, 183
Reinhardt, J., 13
Rosin, C. D., 142
Roubos, H., 115, 141

Sánchez, L., 206, 207
Satyadas, A., 23, 115, 141
Schwefel, H. P., 194
Setnes, M., 115, 141
Shen, Q., 117
Sipper, M., 141–143
Slawinski, T., 101
Sliwinska, D., 181
Sorkin, G., 186
Strauss, C., 99, 198
Stützel, T., 99, 198
Sugeno, M., 13, 14, 16, 182, 183
Surmann, H., 22, 23, 115

Takagi, T., 13, 16
Tanaka, H., 22, 24, 26, 31, 35, 59, 61,
65, 83, 101, 202, 205
Theraulaz, G., 80, 197, 199
Thomas, C., 183
Thorburn, W. M., 19
Thrift, P., 22, 31, 32, 74, 83, 115

Treleaven, P., 14
Trillas, E., 181
Tseng, S., 23, 141

Valente de Oliveira, J., 28, 119
Valverde, L., 181
Vandewalle, J., 23, 119, 141
van Laarhoven, P. J. M., 185
Velasco, J. R., 23
Verdegay, J. L., 19, 34, 35, 38, 59,
101, 115, 124, 141, 192, 195,
196, 202, 204
Vidal, J. J., 19, 173
Villar, P., 23, 60, 141, 168

Wang, C., 23, 141
Wang, L. X., 13, 14, 22, 31, 32, 59, 61,
63, 115, 182

Yamamoto, N., 101
Yan, H., 25
Yasukawa, T., 14, 182, 183
Yu, W., 26

Zadeh, L. A., 13–15, 24, 114, 115, 117,
171, 172, 174, 180
Zimmermann, H.-J., 117
Zwir, S. J. I., 32

