

UNIVERSIDAD DE GRANADA
E.T.S. DE INGENIERIA INFORMATICA



Departamento de Ciencias de la Computación
e Inteligencia Artificial

UNA METODOLOGIA PARA EL DISEÑO
AUTOMATICO DE SISTEMAS BASADOS
EN REGLAS DIFUSAS MEDIANTE
ALGORITMOS EVOLUTIVOS

TESIS DOCTORAL

Oscar Cordón García

Granada, Julio de 1997

UNA METODOLOGIA PARA EL DISEÑO
AUTOMATICO DE SISTEMAS BASADOS
EN REGLAS DIFUSAS MEDIANTE
ALGORITMOS EVOLUTIVOS

OSCAR CORDON GARCIA



**UNA METODOLOGIA PARA EL DISEÑO
AUTOMATICO DE SISTEMAS BASADOS
EN REGLAS DIFUSAS MEDIANTE
ALGORITMOS EVOLUTIVOS**

MEMORIA QUE PRESENTA

OSCAR CORDON GARCIA

PARA OPTAR AL GRADO DE DOCTOR EN INFORMATICA

JULIO 1997

DIRECTOR

FRANCISCO HERRERA TRIGUERO

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACION
E INTELIGENCIA ARTIFICIAL

E.T.S. de INGENIERIA INFORMATICA

UNIVERSIDAD DE GRANADA

La memoria titulada **Una metodología para el diseño automático de Sistemas Basados en Reglas Difusas mediante Algoritmos Evolutivos**, que presenta D. Oscar Cordon García 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 del Doctor D. Francisco Herrera Triguero.

Granada, Julio de 1997.

El Doctorando

El Director

Fdo. O. Cordon

F. Herrera

AGRADECIMIENTOS

En primer lugar, quiero dar las gracias a mis padres, porque sin ellos no habría podido escribir esta memoria, y a mi novia, Mari Carmen, por todo su apoyo, paciencia y comprensión a lo largo del tiempo que me ha llevado hacerla.

Quiero agradecerle a mi director, Francisco Herrera, el haber confiado en mí desde el primer momento y el haberme dedicado tanto tiempo y trabajo para conseguir que esta memoria llegara a ser una realidad.

También quiero destacar especialmente a Luciano Sánchez y a Luis Magdalena por varias razones. La primera y más importante, su amistad y apoyo, y después sus valiosos comentarios que han ayudado a que esta memoria tome la forma que finalmente presenta. Además, un agradecimiento muy especial a Luciano por los datos de las aplicaciones, que me han permitido cerrarla en la forma que tenía en mente.

Agradecer también a todos mis amigos y compañeros del Departamento de Ciencias de la Computación e Inteligencia Artificial todo el apoyo que me han demostrado y, en particular, a Manolo Lozano, por sentar las bases de un trabajo que posteriormente yo he desarrollado y por sus ideas posteriores para mejorarlo, a Raúl Pérez, por su ayuda con el *gnuplot* y con los datos de las funciones, a María José Martín, por su apoyo y por sus comentarios y correcciones, y a José Manuel Benítez, por su paciencia y por su ayuda en todos los temas relacionados con el sistema operativo, el *latex*, el compilador de *C* y todas las aplicaciones que he utilizado.

Finalmente, dar las gracias a a María José del Jesus, por su comprensión y por dejarme la máquina para terminar los últimos experimentos, y a mis amigos y compañeros de despacho, por hacer más agradable el tiempo que ha llevado el desarrollo de este trabajo y porque, a fin de cuentas, con ellos es con quien más tiempo he pasado mientras lo llevaba a cabo.

Y por si me olvido de alguien:

Gracias a Todos

Indice

| | |
|---|-----------|
| Introducción | 9 |
| 1 Sistemas Basados en Reglas Difusas | 15 |
| 1.1 Tipos de Sistemas Basados en Reglas Difusas | 17 |
| 1.1.1 Sistemas Basados en Reglas Difusas Puros | 17 |
| 1.1.2 Sistemas Basados en Reglas Difusas de tipo Takagi-Sugeno | 18 |
| 1.1.3 Sistemas Basados en Reglas Difusas de tipo Mamdani | 19 |
| 1.2 Análisis de las Componentes de los Sistemas Basados en Reglas Difusas de tipo Mamdani | 23 |
| 1.2.1 La Base de Conocimiento o Base de Reglas Difusas | 23 |
| 1.2.2 El Interfaz de Fuzzificación | 24 |
| 1.2.3 El Sistema de Inferencia | 25 |
| 1.2.4 El Interfaz de Defuzzificación | 26 |
| 1.3 Diseño de Sistemas Basados en Reglas Difusas | 27 |
| 1.3.1 Diseño del Mecanismo de Inferencia | 28 |
| 1.3.2 Obtención de la Base de Reglas Difusas | 30 |
| 1.3.2.1 Tareas de diseño para la obtención de la base de reglas difusas | 30 |
| 1.3.2.2 Distintas posibilidades para la definición de la base de reglas difusas | 31 |
| 1.3.2.3 Análisis del proceso de obtención de la base de reglas difusas según el tipo de Sistema Basado en Reglas Difusas empleado | 34 |
| 1.4 Propiedades Deseables en una Base de Reglas Difusas | 35 |

| | | |
|----------|--|-----------|
| 1.4.1 | Complejidad de una Base de Reglas Difusas | 35 |
| 1.4.2 | Consistencia de una Base de Reglas Difusas | 36 |
| 1.4.3 | Continuidad de una Base de Reglas Difusas | 37 |
| 1.4.4 | Interacción de una Base de Reglas Difusas | 37 |
| 1.4.5 | Otras Propiedades asociadas a las Bases de Reglas Difusas | 37 |
| 1.5 | Aplicaciones de los Sistemas Basados en Reglas Difusas | 38 |
| 1.5.1 | Diferencias y Semejanzas entre el Modelado y el Control Difuso | 39 |
| 1.5.2 | Aplicación de los Sistemas Basados en Reglas Difusas al Modelado de Sistemas | 41 |
| 1.5.2.1 | Justificación del empleo de los Sistemas Basados en Reglas Difusas para modelar sistemas | 41 |
| 1.5.2.2 | Relación del modelado difuso con la identificación de sistemas | 43 |
| 1.5.2.3 | Algunas aplicaciones concretas del modelado difuso de sistemas | 44 |
| 1.5.3 | Aplicación de los Sistemas Basados en Reglas Difusas al Control de Sistemas | 45 |
| 2 | Sistemas Basados en Reglas Difusas Evolutivos | 47 |
| 2.1 | Computación Evolutiva | 48 |
| 2.1.1 | Algoritmos Genéticos | 49 |
| 2.1.1.1 | Representación y evaluación de las soluciones | 50 |
| 2.1.1.2 | El mecanismo de selección | 51 |
| 2.1.1.3 | Los operadores de cruce y mutación | 52 |
| 2.1.1.4 | Algoritmos Genéticos para funciones multimodales. Algoritmos Genéticos con nichos | 55 |
| 2.1.1.5 | Características principales de los Algoritmos Genéticos | 56 |
| 2.1.2 | Estrategias de Evolución | 57 |
| 2.1.2.1 | La Estrategia de Evolución (1 + 1) | 58 |
| 2.1.2.2 | La Estrategia de Evolución (μ, λ) | 58 |
| 2.2 | Aprendizaje de Sistemas Basados en Reglas mediante Algoritmos Evolutivos | 60 |
| 2.2.1 | El Enfoque Michigan | 61 |

| | | |
|---------|--|----|
| 2.2.2 | El Enfoque Pittsburgh | 62 |
| 2.2.3 | El Enfoque de Aprendizaje Iterativo de Reglas | 62 |
| 2.2.4 | Análisis de los Tres Enfoques | 63 |
| 2.3 | Sistemas Basados en Reglas Difusas Evolutivos | 64 |
| 2.3.1 | ¿ Qué es un Sistema Basado en Reglas Difusas Evolutivo ? | 64 |
| 2.3.2 | Tipos de Sistemas Basados en Reglas Difusas Evolutivos | 65 |
| 2.3.2.1 | Definición evolutiva de la Base de Datos | 66 |
| 2.3.2.2 | Aprendizaje evolutivo de la Base de Reglas Lingüísticas | 67 |
| 2.3.2.3 | Aprendizaje evolutivo de la Base de Conocimiento | 68 |
| 2.3.3 | El Problema de Cooperación-Competición | 69 |
| 2.3.4 | Aspectos Clave en el Diseño de Sistemas Basados en Reglas Difusas Evolutivos | 71 |
| 2.4 | Algunos Ejemplos Concretos de Sistemas Basados en Reglas Difusas Evolutivos | 74 |
| 2.4.1 | Sistemas Basados en Reglas Difusas Evolutivos para la Definición Automática de la Base de Datos | 74 |
| 2.4.1.1 | El método evolutivo de definición de la Base de Datos de Karr | 75 |
| 2.4.1.2 | El método evolutivo de ajuste de funciones de pertenencia de Herrera, Lozano y Verdegay | 76 |
| 2.4.2 | Sistemas Basados en Reglas Difusas Evolutivos para el Aprendizaje de la Base de Reglas Lingüísticas | 78 |
| 2.4.2.1 | El método evolutivo de generación de la Base de Reglas Lingüísticas de Thrift | 79 |
| 2.4.2.2 | El método evolutivo de generación de la Base de Reglas Lingüísticas de Bonarini | 79 |
| 2.4.3 | Sistemas Basados en Reglas Difusas Evolutivos para el Aprendizaje de la Base de Conocimiento | 81 |
| 2.4.3.1 | El método evolutivo de Lee y Takagi para el aprendizaje de la Base de Conocimiento de un Sistema Basado en Reglas Difusas de tipo TSK | 81 |
| 2.4.3.2 | El método evolutivo de Magdalena y Velasco para el aprendizaje de la Base de Conocimiento de un Sistema Basado en Reglas Difusas descriptivo de tipo Mamdani | 83 |

| | | |
|---------|---|----|
| 2.4.3.3 | El método evolutivo de Magdalena y Velasco para el aprendizaje de Bases de Reglas Difusas aproximativas de tipo Mamdani | 85 |
|---------|---|----|

3 Una Metodología para el Diseño Automático de Sistemas Basados en Reglas Difusas Mediante Algoritmos Evolutivos 89

| | | |
|---------|---|-----|
| 3.1 | Aspectos Básicos de la Metodología Evolutiva de Aprendizaje | 90 |
| 3.1.1 | Propiedades Exigidas a la Base de Reglas Difusas Aprendida | 91 |
| 3.1.2 | Mecanismos para Resolver Adecuadamente el Problema de Cooperación-Competición | 93 |
| 3.1.2.1 | Aspectos de diseño asociados al proceso de generación de reglas difusas | 94 |
| 3.1.2.2 | Aspectos de diseño asociados al proceso de postprocesamiento | 95 |
| 3.1.2.3 | Aspectos de diseño asociados a la composición de los Algoritmos Evolutivos considerados | 96 |
| 3.1.3 | Estructura de los Sistemas Basados en Reglas Difusas Evolutivos en Varias Etapas | 99 |
| 3.1.4 | Manejo de la Información Disponible para el Diseño de la Base de Reglas Difusas | 101 |
| 3.2 | El Proceso de Generación de Reglas Difusas | 102 |
| 3.2.1 | El Método de Generación de Reglas Difusas | 103 |
| 3.2.1.1 | Los criterios frecuentistas considerados para la generación de reglas lingüísticas | 104 |
| 3.2.1.2 | El criterio de baja interacción entre nichos considerado para la generación de reglas aproximativas de tipo Mamdani | 105 |
| 3.2.1.3 | La medida local de error considerada para la generación de reglas difusas TSK | 108 |
| 3.2.1.4 | Un nuevo esquema para la representación de los consecuentes de las reglas difusas TSK: La codificación angular | 108 |
| 3.2.2 | El Método Iterativo de Cubrimiento | 111 |
| 3.3 | El Proceso Genético de Multisimplificación | 113 |
| 3.3.1 | El Proceso Genético Básico de Simplificación | 114 |
| 3.3.2 | Composición del Proceso Genético de Multisimplificación | 115 |

| | | |
|----------|--|------------|
| 3.4 | El Proceso Evolutivo de Ajuste | 117 |
| 3.4.1 | Ajuste de Funciones de Pertenencia y de Consecuentes TSK | 117 |
| 3.4.2 | Definición de la Función de Adaptación | 120 |
| 3.5 | Análisis de los Parámetros Empleados | 122 |
| 4 | Un Sistema Basado en Reglas Difusas Evolutivo para el Aprendizaje de Bases de Conocimiento Descriptivas de tipo Mamdani | 125 |
| 4.1 | Estructura del Sistema Basado en Reglas Difusas Considerado | 127 |
| 4.2 | Estructura del Sistema Basado en Reglas Difusas Evolutivo en Varias Etapas | 128 |
| 4.3 | El Proceso Iterativo de Generación de Reglas Lingüísticas | 129 |
| 4.4 | El Proceso Genético de Ajuste de Bases de Conocimiento Descriptivas | 130 |
| 4.5 | Aplicación del Sistema Basado en Reglas Difusas Evolutivo Propuesto en el Modelado Cualitativo de Algunas Funciones Tridimensionales | 132 |
| 4.5.1 | Experimentación Preliminar | 132 |
| 4.5.2 | Experimentos Realizados | 135 |
| 4.5.3 | Estudio Comparativo y Análisis de los Resultados Obtenidos | 137 |
| 5 | Dos Sistemas Basados en Reglas Difusas Evolutivos para el Aprendizaje de Bases de Reglas Difusas Aproximativas de tipo Mamdani | 149 |
| 5.1 | Un Sistema Basado en Reglas Difusas Evolutivo Según el Enfoque de Semántica Libre No Restringida | 150 |
| 5.1.1 | Estructura del Sistema Basado en Reglas Difusas Evolutivo en Varias Etapas | 150 |
| 5.1.2 | El Proceso Evolutivo de Generación de Reglas Difusas Aproximativas de tipo Mamdani con Semántica Libre No Restringida | 151 |
| 5.1.2.1 | El esquema de representación | 152 |
| 5.1.2.2 | El proceso de mutación | 152 |
| 5.1.2.3 | La función de adaptación | 154 |
| 5.1.3 | El Proceso Genético de Ajuste de Bases de Reglas Difusas Aproximativas | 155 |
| 5.1.4 | Aplicación del Sistema Basado en Reglas Difusas Evolutivo Propuesto en el Modelado Difuso de Algunas Funciones Tridimensionales | 156 |

| | | |
|---------|---|-----|
| 5.2 | Un Sistema Basado en Reglas Difusas Evolutivo Según el Enfoque de Semántica Libre Restringida | 158 |
| 5.2.1 | Estructura del Sistema Basado en Reglas Difusas Evolutivo en Varias Etapas | 158 |
| 5.2.2 | El Proceso Evolutivo de Generación de Reglas Difusas Aproximativas de tipo Mamdani con Semántica Libre Restringida | 159 |
| 5.2.2.1 | El esquema de representación | 159 |
| 5.2.2.2 | La generación de la población inicial | 160 |
| 5.2.2.3 | La función de adaptación | 161 |
| 5.2.2.4 | Los operadores genéticos | 162 |
| 5.2.3 | Aplicación del Sistema Basado en Reglas Difusas Evolutivo Propuesto en el Modelado Difuso de Algunas Funciones Tridimensionales | 166 |
| 5.3 | Estudio Comparativo del Comportamiento de los dos Sistemas Basados en Reglas Difusas Evolutivos Propuestos y Análisis de los Resultados Obtenidos | 169 |

6 Un Sistema Basado en Reglas Difusas Evolutivo para el Aprendizaje de Bases de Conocimiento de tipo TSK 175

| | | |
|---------|---|-----|
| 6.1 | Estructura del Sistema Basado en Reglas Difusas Evolutivo en Varias Etapas | 176 |
| 6.2 | El Proceso Evolutivo de Generación de Reglas de Tipo TSK | 177 |
| 6.2.1 | El Método Evolutivo de Aprendizaje de Consecuentes TSK | 178 |
| 6.2.1.1 | El esquema de representación | 178 |
| 6.2.1.2 | La generación de la población inicial | 178 |
| 6.2.1.3 | La función de adaptación | 180 |
| 6.2.1.4 | Los operadores evolutivos | 181 |
| 6.2.2 | El Método Iterativo de Generación de Antecedentes | 181 |
| 6.3 | El Proceso Evolutivo de Refinamiento de Bases de Conocimiento de Tipo TSK | 182 |
| 6.3.1 | El esquema de representación | 182 |
| 6.3.2 | La generación de la población inicial | 184 |
| 6.3.3 | La función de adaptación | 184 |
| 6.3.4 | Los operadores genéticos | 184 |

| | | |
|----------|--|------------|
| 6.4 | Aplicación del Sistema Basado en Reglas Difusas Evolutivo Propuesto en el Modelado Difuso de Algunas Funciones Tridimensionales | 185 |
| 6.4.1 | Experimentos Realizados | 186 |
| 6.4.2 | Estudio Comparativo y Análisis de los Resultados Obtenidos | 191 |
| 7 | Aplicaciones | 195 |
| 7.1 | Aplicación a la Estimación de Costos de Cable de Baja y Media Tensión en Zonas Rurales y Urbanas | 196 |
| 7.1.1 | Planteamiento del Problema | 196 |
| 7.1.2 | Obtención de un Modelo Difuso de Zona Rural a partir de los Sistemas Basados en Reglas Difusas Evolutivos Propuestos | 198 |
| 7.1.3 | Obtención de un Modelo Difuso de Zona Urbana a partir de los Sistemas Basados en Reglas Difusas Evolutivos Propuestos | 203 |
| 7.1.4 | Conclusiones | 205 |
| | 7.1.4.1 Modelo Difuso de Zona Rural | 205 |
| | 7.1.4.2 Modelo Difuso de Zona Urbana | 207 |
| 7.2 | Valoración de la Calidad del Arroz | 208 |
| 7.2.1 | Planteamiento del Problema | 208 |
| 7.2.2 | Obtención de un Modelo Cualitativo para la Valoración de la Calidad del Arroz a partir de los Sistemas Basados en Reglas Difusas Evolutivos Propuestos | 209 |
| 7.2.3 | Conclusiones | 211 |
| | Comentarios Finales | 215 |
| | A Funciones Tridimensionales Consideradas para su Modelado en la Experimentación Realizada | 223 |
| | B El Proceso Inductivo de Generación de Reglas Difusas de Wang y Mendel | 227 |
| | C El Algoritmo de Conteo Ponderado para la Generación de Reglas Difusas Aproximativas de tipo Mamdani | 229 |
| | Bibliografía | 231 |

Introducción

Planteamiento

En la actualidad, una de las áreas de aplicación más importantes de la *Teoría de Conjuntos Difusos* y de la *Lógica Difusa* [KY95, Zad65, Zim96] la componen los Sistemas Basados en Reglas Difusas (SBRDs). Este tipo de sistemas constituyen una extensión de los Sistemas Basados en Reglas que hacen uso de la Lógica Clásica puesto que emplean reglas de tipo “*SI-ENTONCES*” en las que los antecedentes y consecuentes están compuestos por proposiciones difusas en lugar de proposiciones de la Lógica Clásica. Existen distintos sistemas de este tipo dependiendo de la naturaleza de las entradas y salidas que manejan, así como de la estructura de regla difusa con la que trabajan, pudiendo distinguirse entre SBRDs puros, descriptivos y aproximativos de tipo Mamdani, y de tipo TSK.

Una de las principales aplicaciones de los SBRDs es el *modelado de sistemas*, que puede ser considerado en este campo como una aproximación para modelar un sistema haciendo uso de un lenguaje de descripción basado en Lógica Difusa con predicados difusos. En este modelado suelen existir dos requisitos enfrentados, la precisión y el nivel de interpretabilidad del modelo obtenido.

Es posible distinguir dos tipos de modelado cuando se hace uso de los SBRDs: *modelado cualitativo* y *modelado difuso*, según el requerimiento prioritario sea la interpretabilidad o la precisión del modelo, respectivamente. El primero de ellos se realiza mediante los SBRDs descriptivos, los cuales emplean reglas difusas con variables lingüísticas que toman valores en un conjunto de términos con un significado real, de modo que el modelo cualitativo obtenido finalmente consiste en una serie de descripciones lingüísticas sobre el comportamiento del sistema que está siendo modelado. El modelado cualitativo, en cambio, se lleva a cabo mediante los SBRDs aproximativos, sistemas en los que las reglas difusas están compuestas por predicados difusos sin interpretación lingüística, es decir, las variables que componen las reglas no toman como valores una serie de términos lingüísticos con una semántica asociada en forma de conjunto difuso, sino directamente un conjunto difuso.

Para obtener un SBRD de cualquiera de los tipos anteriores es necesario llevar a cabo una serie de tareas de diseño, englobadas principalmente en dos grupos:

- El diseño del Sistema de Inferencia, elemento del SBRD que lleva a cabo el proceso de inferencia difuso.
- La generación de la base de reglas difusas que contiene la información empleada por el sistema para proporcionar una salida ante una entrada determinada.

Esta segunda tarea es complicada de llevar a cabo cuando no existe un experto capaz de determinar la estructura de la base de reglas difusas y su realización requiere, en muchos casos, el empleo de técnicas automáticas de aprendizaje que generen la base a partir de la información numérica disponible sobre el problema a resolver.

En los últimos años, se han desarrollado una gran cantidad de métodos de aprendizaje automático que tratan de afrontar la generación de la base de reglas difusas a partir de la información numérica disponible y haciendo uso de las técnicas más variadas. Una gran parte de estos procesos presentan una serie de inconvenientes tales como el hecho de que desperdician la información experta disponible o que solamente permiten el diseño de un tipo concreto de SBRD.

Debido a estas razones, parece adecuado el desarrollo de una metodología general que permita la generación automática de la base de reglas difusas en el caso en el que éstas no puedan ser obtenidas directamente a partir de la información experta existente. Esta metodología debería posibilitar la creación de procesos de aprendizaje automático que permitan diseñar SBRDs de todos los tipos existentes y que aprovechen la información experta disponible para mejorar la definición de la base de reglas difusas. Además, se debería establecer una estructura genérica y una serie de recomendaciones de diseño que sean, a la vez, lo suficientemente flexibles como para que el construir procesos concretos de aprendizaje a partir de las mismas sea sencillo y lo suficientemente robustas como para que dichos procesos permitan el diseño de SBRDs que presenten buen comportamiento.

Por otro lado, la combinación de la Lógica Difusa y los Algoritmos Evolutivos ha obtenido unos resultados muy prometedores en los últimos años [CHL96b, CHL97a, CHL97b, HV96, Ped97, SSZ97, Tet95]. La definición automática de un Sistema Difuso puede ser afrontada como un proceso de optimización o de búsqueda y los Algoritmos Evolutivos (AEs) [Bäc96], en particular los Algoritmos Genéticos [Gol89, Hol75, Mic96], están considerados actualmente como la técnica de búsqueda global más conocida y empleada. Este tipo de algoritmos presentan la capacidad de explorar y explotar espacios de búsqueda complejos, lo que les permite obtener soluciones muy próximas a la óptima en dichos espacios. Además, la codificación genética que emplean les permite incorporar conocimiento a priori de una forma muy sencilla y aprovecharlo para guiar la búsqueda. Todas estas razones han incrementado el empleo de este tipo de algoritmos para el diseño de Sistemas Difusos a lo largo de los últimos años, lo que ha dado lugar a la creación de los *Sistemas Difusos Evolutivos* [CH95, HM97].

Un Sistema Difuso Evolutivo es un método de diseño automático de Sistemas Difusos basado en AEs. Dentro de este gran grupo, los más conocidos y estudiados son los *Sistemas Basados en Reglas Difusas Evolutivos* (SBRDEs) [CH95, GH97, HM97], en los

que se hace uso de los AEs para obtener de un modo automático la totalidad o una parte de la base de reglas difusas del SBRD. Existen distintos tipos según que la tarea de diseño que lleven a cabo sea la definición de las funciones de pertenencia de los conjuntos difusos de las reglas, la obtención de las propias reglas o ambas.

Objetivos

El objetivo de esta memoria es presentar una metodología para la obtención de procesos evolutivos de aprendizaje automático de bases de reglas difusas que sea lo suficientemente general como para permitir el diseño de SBRDs de cualquier tipo de un modo eficiente y sencillo, en la línea indicada anteriormente. Además, nos planteamos emplear el modelo evolutivo de aprendizaje propuesto para construir una serie de SBRDEs concretos que permitan diseñar SBRDs de todas las clases existentes.

Este objetivo general se descompone en varios objetivos particulares que se comentan a continuación:

1. *Hacer un análisis en profundidad de los distintos aspectos básicos relacionados con los SBRDEs.* Puesto que el objetivo final es el diseño de una metodología general que permita la obtención de SBRDEs, debemos poner especial atención en los siguientes aspectos:
 - Analizar la resolución del problema fundamental que aparece en los SBRDEs, el *problema de cooperación-competición*, consistente en hallar la forma más adecuada para conseguir que la competición inducida por el AE permita generar bases de reglas difusas cuyas reglas cooperen de un modo adecuado para proporcionar una respuesta correcta ante cualquier entrada que reciba el sistema.
 - Estudiar los tres enfoques evolutivos de aprendizaje de Sistemas Basados en Reglas existentes: el enfoque Michigan, el enfoque Pittsburgh y el enfoque de Aprendizaje Iterativo de Reglas, con objeto de analizar las ventajas e inconvenientes que presentan así como de estimar el grado de dificultad asociado al problema de cooperación-competición cuando se considera alguno de estos enfoques en el diseño de un SBRDE.
 - Determinar cuáles son las tareas de diseño que es necesario desarrollar para construir un SBRDE y cuál es el mejor modo de llevarlas a cabo.
2. *Diseñar un paradigma evolutivo de aprendizaje automático de bases de reglas difusas.* Este modelo debe ser lo suficientemente flexible, robusto y general como para posibilitar la creación de procesos de aprendizaje automático con un buen funcionamiento que permitan diseñar SBRDs de todos los tipos existentes y que hagan un uso adecuado de la información experta disponible para mejorar la definición de la base de reglas difusas.

3. *Diseñar un SBRDE para el aprendizaje automático de Bases de Conocimiento descriptivas de tipo Mamdani.* Nos proponemos hacer uso de la metodología propuesta para obtener un SBRDE que pueda ser empleado en tareas de modelado cualitativo. Dicho proceso evolutivo de aprendizaje deberá permitir el diseño de SBRDs descriptivos de tipo Mamdani que mantengan el nivel de descripción asociado a este tipo de sistemas al mismo tiempo que aumenten la precisión.
4. *Construir SBRDEs para el aprendizaje de Bases de Reglas Difusas aproximativas de tipo Mamdani.* Dichos SBRDEs deberán afrontar la generación de una Base de Reglas Difusas aproximativa al completo, es decir, el aprendizaje automático de las reglas difusas y de las funciones de pertenencia manejadas en las mismas. Estos procesos evolutivos de aprendizaje serán adecuados para su aplicación en problemas complejos de modelado difuso en los que un modelo cualitativo no permite obtener el nivel de precisión deseado.
5. *Diseñar un SBRDE para el aprendizaje de Bases de Conocimiento de tipo TSK:* Con objeto de completar todos los tipos de SBRDs existentes, pretendemos construir un proceso evolutivo de aprendizaje para el diseño de SBRDs de tipo TSK. Debido a la estructura de regla empleada por el SBRD considerado, en la que el consecuente no es interpretable, el área de aplicación de este último proceso serán aquellos problemas de modelado en los que el único requisito sea la precisión del modelo generado.
6. *Estudiar el comportamiento de los SBRDEs obtenidos en la resolución de una serie de problemas de modelado con distintos requerimientos.*

Resumen

La presente memoria está organizada en siete capítulos, una sección de "Comentarios Finales" y tres apéndices. A continuación resumimos brevemente los contenidos de cada uno de ellos:

En el capítulo 1 introduciremos las nociones básicas de los SBRDs, describiendo los distintos tipos existentes, su composición y funcionamiento, analizando detenidamente las tareas de diseño que es necesario llevar a cabo para obtenerlos.

En el capítulo 2 nos centraremos en el estudio de los *Sistemas Basados en Reglas Difusas Evolutivos*. Introduciremos brevemente los tipos de AEs más empleados, analizaremos su aplicación clásica al aprendizaje de Sistemas Basados en Reglas y estudiaremos detalladamente los SBRDEs, presentando los tipos existentes y la problemática de diseño asociada, así como unos cuantos ejemplos.

En el capítulo 3 presentaremos la metodología evolutiva de aprendizaje de bases de reglas difusas propuesta. En dicho capítulo sentaremos las bases de este modelo, que tendrá en cuenta las características principales de los dos elementos considerados en el proceso de diseño (el sistema a diseñar, el SBRD, y la técnica utilizada para llevar a cabo

dicho diseño, el AE) para resolver del modo más adecuado el problema de cooperación-competición y para refinar al máximo la definición de las bases de reglas difusas obtenidas.

En el capítulo 4 introduciremos un SBRDE para el aprendizaje de Bases de Conocimiento de SBRDs descriptivos de tipo Mamdani, el cual trabajará sobre una estructura especial de base de reglas difusas que permitirá mejorar el grado de precisión del sistema al mismo tiempo que se mantiene un nivel de descripción aceptable. Al final del capítulo mostraremos los resultados obtenidos por el proceso propuesto en una serie de aplicaciones de modelado cualitativo, presentadas en el Apéndice A, y lo compararemos con otros SBRDEs pertenecientes a la misma familia.

En el capítulo 5 presentaremos dos SBRDEs para el aprendizaje de Bases de Reglas Difusas aproximativas de tipo Mamdani con una filosofía de generación de reglas distinta, basadas respectivamente en una semántica libre restringida o no restringida. Este hecho dará lugar a que cada uno de ellos sea más adecuado para un problema de modelado difuso concreto en función de la complejidad del sistema que se desee modelar. Al término de este capítulo estudiaremos también el comportamiento de ambos SBRDEs en las aplicaciones comentadas y los compararemos con otros dos SBRDEs de características similares.

En el capítulo 6 introduciremos el último de los SBRDEs construido a partir de la metodología evolutiva de aprendizaje propuesta en el capítulo 3. En este caso, el proceso evolutivo de aprendizaje presentado permitirá el diseño de SBRDs de tipo TSK. Al igual que en los dos capítulos anteriores, una vez presentado el propio SBRDE, se estudiará su comportamiento en una serie de aplicaciones y se comparará con otro SBRDE incluido en la misma familia.

En el capítulo 7 recogeremos una serie de aplicaciones reales, dos pertenecientes al campo de la ingeniería eléctrica y una al de la valoración de la calidad de las comidas, que nos permitirán estudiar el comportamiento de los cuatro SBRDEs propuestos en los tres capítulos anteriores. Puesto que los requisitos a satisfacer y la complejidad de los sistemas a modelar son distintos en cada caso, estos problemas de modelado constituyen una buena herramienta para comparar el comportamiento de distintos tipos de modelos difusos en su resolución.

Posteriormente, incluiremos un apartado de "Comentarios finales", que resumirá los resultados obtenidos en esta memoria y presentará algunas conclusiones sobre éstos, y tres apéndices, dedicados respectivamente a describir las aplicaciones consideradas para los problemas de modelado cualitativo y difuso afrontados en los capítulos 4, 5 y 6, y a introducir dos métodos inductivos de generación de reglas difusas utilizados para construir distintos SBRDEs empleados en dichos capítulos. Finalizaremos la memoria con una recopilación bibliográfica que recogerá las contribuciones más destacadas en la materia estudiada.

Capítulo 1

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 Sistemas Basados en Reglas Difusas (SBRDs). Este tipo de sistemas constituyen una extensión de los Sistemas Basados en Reglas que hacen uso de la Lógica Clásica puesto que emplean reglas de tipo “*SI-ENTONCES*” en las que los antecedentes y consecuentes están compuestos por proposiciones difusas en lugar de proposiciones de la Lógica Clásica.

En un sentido muy general, un SBRD es un Sistema Basado en Reglas en el que la Lógica Difusa puede ser empleada 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 difuso de sistemas* [BD95, Ped96, SY93], puesto que existe una amplia gama de sistemas reales en los que, debido a su nivel de complejidad o imprecisión, las herramientas clásicas no permiten obtener buenos resultados. Los SBRDs se han aplicado con éxito a una gran cantidad de problemas reales a lo largo de los últimos años [BD95, DHR93, GT95, Hir93, Ped96, Wan94].

El modelado difuso de sistemas se puede considerar como una vía para modelar un sistema haciendo uso de un lenguaje de descripción basado en Lógica Difusa con predicados difusos [SY93]. Tal y como veremos a lo largo de este capítulo, empleando distintos tipos de SBRDs es posible llevar a cabo distintos tipos de modelado, dependiendo del grado de descripción y precisión que deseemos presente el modelo obtenido. Estos dos requisitos, el grado en el cual es interpretable el modelo por el ser humano y el grado en que ese modelo aproxima al sistema real, suelen estar enfrentados. Como ya indicaba Zadeh en su *principio de incompatibilidad* [Zad73], *cuando la complejidad de un problema aumenta, nuestra capacidad para describirlo en términos precisos disminuye*. De este modo, aunque el uso de técnicas basadas en Lógica Difusa, concretamente de los SBRDs, nos permite afrontar el modelado de sistemas que están envueltos en una cierta imprecisión, obteniendo un modelo cualitativo claramente interpretable por el ser humano, muchas veces el rendimiento del modelo obtenido no es el deseado y preferimos perder en la claridad de descripción para

ganar en el rendimiento global del modelo. La decisión entre cuán interpretable y cuán preciso debe ser el modelo a obtener suele depender de las necesidades que se tengan para el problema concreto.

La Lógica Difusa posee herramientas que permiten obtener distintos tipos de modelos difusos según deseemos que predomine la interpretabilidad o la precisión en los mismos. De esta forma, es posible distinguir dos tipos de modelado cuando se trabaja en este campo: *modelado cualitativo* y *modelado difuso* [SY93]:

- El primero de ellos se realiza mediante los *SBRDs descriptivos*, los cuales emplean reglas difusas con variables lingüísticas que toman valores en un conjunto de términos con un significado real, de tal modo que el modelo cualitativo finalmente obtenido forma un modelo difuso generalizado que consiste en una serie de descripciones lingüísticas sobre el comportamiento del sistema que está siendo modelado. En este primer tipo de modelado, la interpretabilidad del modelo es la nota predominante. Como se indica en [SY93], las ideas básicas de este tipo de modelado ya aparecían en los primeros trabajos de Zadeh [Zad73].
- El segundo tipo de modelado, en cambio, se lleva a cabo mediante los *SBRDs aproximativos*. En este tipo de sistemas, las reglas difusas están compuestas por predicados difusos sin ninguna interpretación lingüística, es decir, las variables que componen las reglas no toman como valores una serie de términos lingüísticos con una semántica asociada en forma de conjunto difuso, sino directamente un conjunto difuso. En este tipo de modelado basado en reglas difusas se asume que la meta principal es la precisión en el modelado del comportamiento del sistema real y se hace uso de la potencialidad de los SBRDs como aproximadores universales [Buc93, Cas95, Cas96, Kos92, Wan92] y no tanto de su capacidad para representar el conocimiento en forma de reglas lingüísticas. Aún así, el modelo obtenido seguirá presentando un cierto nivel de interpretabilidad que provendrá del hecho de estar compuesto por un SBRD; siempre será posible determinar qué reglas han actuado ante una determinada entrada y, gracias a ello, se podrá estudiar localmente el comportamiento del modelo.

En este capítulo, vamos a introducir las nociones básicas de los SBRDs, describiendo los distintos tipos existentes, su composición y funcionamiento, y analizando las tareas de diseño que es necesario llevar a cabo para obtenerlos. Dedicaremos una sección al estudio de las aplicaciones en las que pueden ser empleados, principalmente el modelado y el control difuso. No entraremos, sin embargo, en los principios básicos de la Lógica Difusa, que pueden ser consultados en [KY95, Zim96].

1.1 Tipos de Sistemas Basados en Reglas Difusas

En la literatura especializada se suele distinguir entre tres tipos de SBRDs según la forma que presentan las reglas difusas que emplean y según el tipo de entradas y salidas que manejan [Wan94]. Describiremos dichos tipos en las secciones siguientes.

1.1.1 Sistemas Basados en Reglas Difusas Puros

La estructura básica de un SBRD de este primer tipo se recoge en la figura 1.1. Estos sistemas reciben como entrada y devuelven como salida conjuntos difusos. Al no tener que efectuar ninguna transformación sobre las entradas o las salidas, están formados por dos componentes principales: una *Base de Conocimiento*, que almacena el conocimiento disponible sobre el problema en forma de reglas lingüísticas de tipo “*SI-ENTONCES*”, y un *Sistema de Inferencia*, encargado de llevar a cabo el proceso de inferencia sobre las entradas haciendo uso de la información almacenada en la Base de Conocimiento (describiremos este proceso en la siguiente sección).

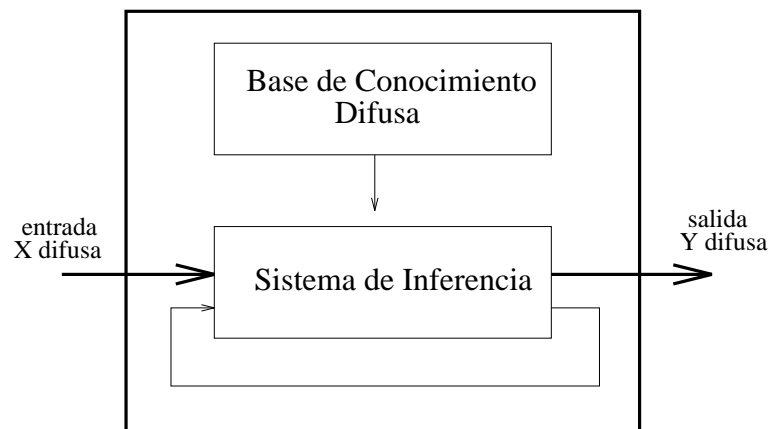


Figura 1.1. Estructura básica de un Sistema Basado en Reglas Difusas puro

Las reglas lingüísticas empleadas presentan la siguiente forma en el caso de trabajar con sistemas con múltiples entradas y una única salida:

$$\text{Si } X_1 \text{ es } A_1 \text{ y } \dots \text{ y } X_n \text{ es } A_n \text{ entonces } Y \text{ es } B,$$

donde X_i e Y son variables lingüísticas de entrada y salida respectivamente, y los A_i y B son etiquetas lingüísticas asociadas con conjuntos difusos que determinan su semántica.

Los SBRDs puros constituyen el modelo esencial de Sistema Difuso, un marco de trabajo general en el cual se cuantifica la información lingüística procedente de los expertos

humanos y se emplean los principios de la Lógica Difusa para hacer un uso sistemático de esa información. Este tipo de SBRDs constituye la derivación natural de las ideas iniciales de Zadeh [Zad73]. Su principal inconveniente es el hecho de que las entradas y salidas sean conjuntos difusos, puesto que en la gran mayoría de los sistemas de ingeniería las entradas y salidas que se manejan son números reales. Para evitar este problema, Takagi y Sugeno, por un lado, y Mamdani, por otro, propusieron otros dos tipos de SBRDs que trabajan con este tipo de entradas y salidas.

1.1.2 Sistemas Basados en Reglas Difusas de tipo Takagi-Sugeno

En lugar de trabajar con reglas difusas del tipo introducido en la sección anterior, Takagi y Sugeno [TS85] propusieron un nuevo modelo de SBRD que empleaba reglas en las que el antecedente estaba constituido por variables lingüísticas o difusas y el consecuente representaba una función de las variables de entrada. La forma más general de este tipo de reglas es la que mostramos a continuación, en la que el consecuente constituye una combinación lineal de las variables contenidas en el antecedente:

$$\text{Si } X_1 \text{ es } A_1 \text{ y } \dots \text{ y } X_n \text{ es } A_n \text{ entonces } Y = p_1 \cdot X_1 + \dots + p_n \cdot X_n + p_0,$$

donde X_i son las variables de entrada del sistema, Y es la variable de salida y los p_i son parámetros reales. En lo que respecta a los A_i , pueden ser bien etiquetas lingüísticas asociadas con conjuntos difusos en el caso en que las X_i sean variables lingüísticas, o bien conjuntos difusos en el caso en que éstas sean directamente variables difusas. Este tipo de reglas suelen denominarse *reglas difusas de tipo TSK*, en alusión a sus creadores [TS85].

La salida de un SBRD TSK que emplee una Base de Conocimiento formada por m reglas de este tipo se obtiene como la media ponderada de las salidas individuales aportadas por cada regla, Y_i , $i = 1 \dots m$, del siguiente modo:

$$\frac{\sum_{i=1}^m h_i \cdot Y_i}{\sum_{i=1}^m Y_i},$$

donde $h_i = T(A_1(x_1), \dots, A_n(x_n))$ es el grado de emparejamiento entre la parte antecedente de la regla i y las entradas actuales al sistema, $x = (x_1, \dots, x_n)$. T es un operador de conjunción que se modela mediante una t-norma.

De este modo, tal y como enuncian sus creadores en [TS85], este SBRD se basa en la división del espacio de entrada (uni o multidimensional) en varios subespacios difusos (uni o multidimensionales) y en definir una relación de entrada-salida lineal en cada uno de dichos subespacios. En el proceso de inferencia, se combinan estas relaciones parciales en el modo comentado para así obtener la relación global de entrada-salida, teniendo en cuenta la dominancia de las relaciones parciales en sus respectivas áreas de aplicación y el conflicto que se presenta en las zonas en las que existe solapamiento.

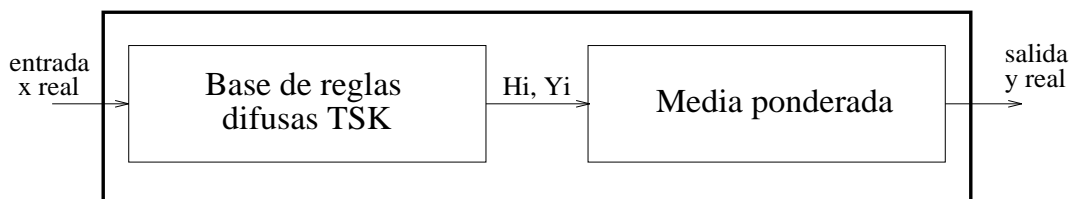


Figura 1.2. Estructura básica de un Sistema Basado en Reglas Difusas TSK

En la figura 1.2 se muestra una representación gráfica de este segundo tipo de SBRDs. Los SBRDs TSK han sido aplicados con éxito a una gran cantidad de problemas prácticos. La ventaja principal de estos sistemas es el hecho de presentar una ecuación del sistema compacta que permite estimar los parámetros p_i empleando métodos clásicos, lo que facilita su diseño. Sin embargo, el mayor inconveniente que tienen asociado está también relacionado con la forma de los consecuentes de las reglas, que al no ser difusos, provocan que el sistema no constituya un marco de trabajo natural para representar el conocimiento experto. Es posible integrar conocimiento experto en estos SBRDs efectuando una pequeña modificación sobre el consecuente de la regla: cuando un experto aporta una regla lingüística con consecuente Y es B , se sustituye dicho consecuente por $Y = p_0$, donde p_0 toma como valor el punto modal del conjunto difuso asociado a la etiqueta B . Este tipo de reglas se suelen denominar *reglas TSK simplificadas* o *de orden cero*. Aun así, no se pueden emplear muchos de los principios de la Lógica Difusa en este tipo de sistemas (por ejemplo, emplear la potencialidad del Razonamiento Aproximado).

En lo que respecta al tipo de modelado que es posible llevar a cabo, los SBRDs TSK constituyen un punto intermedio entre el cualitativo y el difuso. En el caso en que los antecedentes de las reglas estén compuestos por variables lingüísticas, los SBRDs TSK combinan un grado de descripción aceptable (aunque no sea posible interpretar la acción que se lleva a cabo, es posible determinar lingüísticamente de un modo claro cuando dicha acción es aplicada) con un grado de precisión superior al que se puede obtener con un consecuente difuso. En cambio, en el caso en que las variables que conforman los antecedentes sean difusas y no lingüísticas, es claro que el SBRD se reduce a un modelo difuso que persigue la precisión como característica primordial.

1.1.3 Sistemas Basados en Reglas Difusas de tipo Mamdani

El segundo tipo de SBRD que trabaja con entradas y salidas reales fue propuesto por Mamdani [Mam74], el cual fue capaz de plasmar las ideas preliminares de Zadeh [Zad73] en el primer SBRD tangible en una aplicación de control. Este tipo de Sistemas Difusos, los más utilizados desde aquella fecha, se conocen también por el nombre de SBRDs con Fuzzificador y Defuzzificador o, más comúnmente, por el de *controladores difusos* (nombre que ya acuñó Mamdani en sus primeros trabajos [MA75]), ya que su aplicación principal ha sido históricamente el control de sistemas.

Los SBRDs de tipo Mamdani 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: los *Interfaces de Fuzzificación y Defuzzificación*.

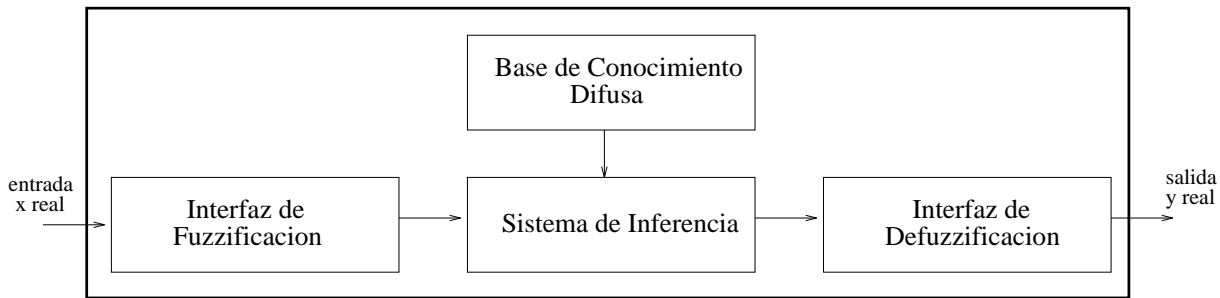


Figura 1.3. Estructura básica de un Sistema Basado en Reglas Difusas de tipo Mamdani

La figura 1.3 muestra la estructura general de los SBRDs de tipo Mamdani. Tal y como comentaremos en la siguiente sección, en la que se estudiará detenidamente cada uno de los componentes de este tipo de sistemas, el 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. El 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 de tipo Mamdani presenta una serie de características muy interesantes. Por un lado, puede ser empleado en aplicaciones reales de ingeniería, puesto que maneja entradas y salidas reales. Por otro, proporciona un marco natural para incluir conocimiento experto en forma de reglas lingüísticas y permite combinar éste 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. Analizaremos estas cuestiones en secciones posteriores de este mismo capítulo.

Con respecto al tipo de modelado que llevan a cabo, los SBRDs de tipo Mamdani fueron inicialmente pensados para realizar un modelado cualitativo. 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.

Desafortunadamente, tal y como comentábamos en la Introducción de este capítulo, los modelos cualitativos obtenidos mediante SBRDs de tipo Mamdani tienen la desventaja de ser menos exactos que los modelos difusos cuando se trabaja con sistemas que presentan

un nivel de complejidad alto. Este hecho se debe a ciertos problemas relacionados con la estructura de las reglas difusas que emplean. En [Bas94], se analizan detenidamente estos problemas y se llega a la conclusión de que la estructura de regla difusa de tipo “*SI-ENTONCES*” basada en el manejo de variables lingüísticas presenta las siguientes limitaciones relacionadas con el empleo de este tipo de variables:

1. Se da una falta de flexibilidad en el SBRD debido a la forma tan rígida en la que se particionan los espacios de entrada y salida.
2. No existe una distinción clara entre el conocimiento experto y la definición de las variables lingüísticas incluidas en las reglas difusas.
3. Cuando las variables de entrada del sistema dependen unas de otras, es muy complicado obtener una partición difusa adecuada de los espacios de entrada.
4. El tamaño de la Base de Conocimiento 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 en 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.

Debido a esta serie de problemas, que provocan que el SBRD obtenido no presente el grado de precisión deseado, en los últimos años ha surgido una nueva variante de los SBRDs de tipo Mamdani, que antepone la precisión del sistema obtenido a su interpretabilidad. Este tipo de sistemas se denominan habitualmente *SBRDs aproximativos de tipo Mamdani* [BD95, CH95, CH97c, Koc96], en contraposición a los que siguen la estructura clásica que se pueden nombrar como *SBRDs descriptivos de tipo Mamdani*. Obviamente, la aplicación principal de los SBRDs aproximativos es el modelado difuso de sistemas.

La estructura de un SBRD aproximativo es prácticamente la misma que la de un descriptivo, es decir, la que mostrábamos en la figura 1.3. La única diferencia radica en el tipo de reglas que se emplean en la Base de Conocimiento. En este caso, dichas reglas no manejan variables lingüísticas sino, directamente, variables difusas. Así, la estructura de las reglas difusas consideradas por los sistemas aproximativos es la siguiente:

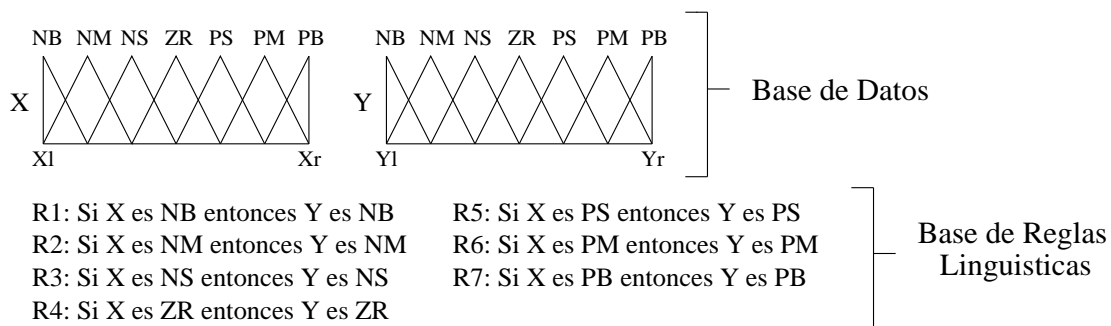
$$\text{Si } X_1 \text{ es } A_1 \text{ y } \dots \text{ y } X_n \text{ es } A_n \text{ entonces } Y \text{ es } B,$$

donde el único cambio que se produce con respecto a la estructura de regla empleada en los SBRDs puros y descriptivos de tipo Mamdani, es el hecho de que las variables de

entrada X_i y la de salida Y son variables difusas en lugar de lingüísticas y, por tanto, los A_i y B son conjuntos difusos sin interpretación directa en lugar de etiquetas lingüísticas.

Por lo tanto, este tipo de sistemas no necesitan del empleo de una Base de Datos que almacene los términos lingüísticos existentes y los conjuntos difusos que determinan la semántica asociada a los mismos. En este caso, la Base de Conocimiento empleada en los SBRDs de los tipos anteriores, que estaba compuesta por la Base de Datos comentada y por una Base de Reglas Lingüísticas, queda reducida a una Base de Reglas Difusas formada 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 1.4 representa gráficamente este hecho.

a) Base de Conocimiento



b) Base de Reglas Difusas

- R1: Si X es entonces Y es
- R2: Si X es entonces Y es
- R3: Si X es entonces Y es
- R4: Si X es entonces Y es

Figura 1.4. Comparación gráfica entre una Base de Conocimiento descriptiva y una Base de Reglas Difusas aproximativa

Finalizaremos esta sección, matizando un poco más la idea comentada sobre el grado de precisión que es posible alcanzar mediante el empleo de los SBRDs aproximativos de tipo Mamdani. Puesto que el nivel de interacción existente entre los conjuntos difusos incluidos en reglas vecinas contenidas en la Base de Reglas Difusas aproximativa es variable y depende de cada pareja de reglas concreta, este tipo de sistemas son muy adecuados para trabajar con problemas complejos en los que pequeños cambios en las entradas tienen asociados grandes cambios en las salidas. Es al afrontar este tipo de problemas cuando los SBRDs aproximativos de tipo Mamdani son más precisos que los descriptivos del mismo tipo.

En cambio, esta característica representa una desventaja en problemas sencillos y lo será tanto más cuantas más reglas contenga la Base de Reglas Difusas. En este caso, existirán más interacciones entre las reglas difusas y los niveles de interacción serán más variados y precisamente, para obtener buenos resultados en problemas de baja complejidad

es necesario un nivel de interacción fijo y constante. Por esta razón, es mejor el uso de SBRDs descriptivos ya que consideran unas particiones difusas globales que definen la semántica de todos los términos empleados en las reglas lingüísticas de la Base de Conocimiento.

1.2 Análisis de las Componentes de los Sistemas Basados en Reglas Difusas de tipo Mamdani

Repasando los conceptos introducidos en la sección anterior, un SBRD de tipo Mamdani está formado por las siguientes componentes:

- una *Base de Conocimiento o de Reglas Difusas*, que contiene las reglas lingüísticas o difusas, según sea el sistema descriptivo o aproximativo, las cuales guían el comportamiento del mismo,
- un *Interfaz de Fuzzificación*, que se encarga de transformar los datos de entrada precisos en valores manejables 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, y
- un *Interfaz de Defuzzificación*, que transforma la acción difusa resultante del proceso de inferencia en una acción precisa que constituye la salida global del SBRD.

Analizaremos detenidamente cada una de estas componentes en las secciones siguientes.

1.2.1 La Base de Conocimiento o Base de Reglas Difusas

La Base de Conocimiento o de Reglas Difusas es la parte esencial del SBRD debido a que las tres componentes restantes del sistema se ocupan de interpretar las reglas contenidas en ella y hacerlas manejables en problemas concretos. Tal y como comentábamos en la sección anterior, la componente que almacena estas reglas se denomina Base de Conocimiento (BC) o Base de Reglas Difusas (BRD) según el SBRD sea descriptivo o aproximativo.

En el primero de los casos, es decir, en los SBRDs descriptivos, la BC está formada a su vez por dos componentes distintos: la Base de Reglas Lingüísticas y la Base de Datos:

- La *Base de Reglas Lingüísticas* (BRL) está formada por un conjunto de reglas lingüísticas de tipo “SI-ENTONCES” que, en el caso de SBRDs con múltiples entradas y una única salida, presentan la estructura introducida en la sección 1.1.1:

$$\text{Si } X_1 \text{ es } A_1 \text{ y } \dots \text{ y } X_n \text{ es } A_n \text{ entonces } Y \text{ es } B,$$

donde X_i e Y son variables lingüísticas de entrada y salida respectivamente, y los A_i y B son etiquetas lingüísticas asociadas a dichas variables. La BRL está compuesta por una serie de reglas de este tipo unidas por el operador *además*, lo que indica, como veremos en la sección siguiente, que todas ellas pueden dispararse ante una entrada concreta.

La estructura de una regla lingüística puede ser más general si se emplea otro conectivo, en lugar del *y*, para relacionar las variables de entrada en el antecedente pero en trabajos como [Wan94] se demuestra que la estructura de la regla mostrada es lo suficientemente general como para incluir otras. Debido a este hecho y a su simplicidad, este tipo de reglas son las más empleadas en la literatura especializada.

- La *Base de Datos* (BD) contiene la definición de los conjuntos difusos asociados a los términos lingüísticos empleados en las reglas de la BRL, 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 difusos a aquellos en los que se definen las variables de entrada y salida del sistema.

En los sistemas aproximativos, sin embargo, esta última componente no es necesaria puesto que las variables que se emplean en las reglas no son lingüísticas sino difusas. La razón es obvia puesto que, al tomar cada variable como valor un conjunto difuso distinto para cada regla y no una etiqueta lingüística, no tiene sentido establecer una relación entre etiquetas y conjuntos difusos. En este tipo de sistemas, la BC queda reducida a una BRD compuesta por un conjunto de reglas difusas aproximativas, del tipo introducido en la sección 1.1.3, conectadas entre si mediante el operador *además*.

1.2.2 El Interfaz de Fuzzificación

El Interfaz de Fuzzificación es una de las dos componentes que permite al SBRD de tipo Mamdani trabajar con entradas y salidas reales. Su tarea es establecer una aplicación que haga corresponder cada valor preciso del espacio de entrada con un conjunto difuso definido en el universo de discurso de dicha entrada. Así, el 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 \mathbf{U} , A' es un conjunto difuso definido sobre el mismo dominio y F es un operador de fuzzificación.

Principalmente, existen dos posibilidades para la elección de F :

1. *Fuzzificación puntual*: A' se construye como un conjunto difuso puntual con soporte x_0 , es decir, con la siguiente función de pertenencia:

$$A'(x) = \begin{cases} 1, & \text{si } x = x_0 \\ 0, & \text{en otro caso} \end{cases}$$

2. *Fuzzificación no puntual o aproximada*: En este caso, $A'(x_0) = 1$ y el grado de pertenencia de los valores restantes de \mathbf{U} va disminuyendo según se alejan éstos de x_0 . Este segundo tipo de operador de fuzzificación permite el empleo de distintos tipos de funciones de pertenencia. Por ejemplo, en el caso de una función de pertenencia triangular, se puede emplear el siguiente:

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

La primera de ellas es habitualmente la más empleada debido a su sencillez.

1.2.3 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 principios de la Lógica Difusa para establecer una aplicación entre conjuntos difusos definidos en $\mathbf{U} = \mathbf{U}_1 \times \mathbf{U}_2 \times \dots \times \mathbf{U}_n$ y conjuntos difusos definidos en \mathbf{V} (donde $\mathbf{U}_1, \dots, \mathbf{U}_n$ y \mathbf{V} son los dominios en los que están definidas las variables de entrada X_1, \dots, X_n y la de salida Y , respectivamente).

El proceso de inferencia difuso 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]:

$$\begin{array}{l} \text{Si } X \text{ es } A \text{ entonces } Y \text{ es } B \\ \underline{X \text{ es } A'} \\ \hline Y \text{ es } B' \end{array}$$

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* representa una relación difusa entre A y B definida en $\mathbf{U} \times \mathbf{V}$. Dicha relación se expresa mediante un conjunto difuso R cuya función de pertenencia $\mu_R(x, y)$ presenta la forma:

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

donde $\mu_A(x)$ y $\mu_B(y)$ son las 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. En la sección siguiente estudiaremos más detenidamente estos operadores.

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 \mathbf{U} a \mathbf{V} y A' es un conjunto difuso definido en \mathbf{U} , entonces el conjunto difuso B' , inducido por A' , viene dado por la composición ... de R y A' ”, es decir:

$$B' = A' \circ R$$

De este modo, cuando la Regla Composicional de Inferencia se aplica sobre reglas cuyo antecedente está formado por n variables de entrada y cuyo consecuente presenta una única variable de salida, toma la siguiente expresión:

$$\mu_{B'}(y) = \text{Sup}_{x \in \mathbf{U}} \{T'(\mu_{A'}(x), I(\mu_A(x), \mu_B(y)))\},$$

donde $\mu_{A'}(x) = T(\mu_{A'_1}(x), \dots, \mu_{A'_n}(x))$, $\mu_A(x) = T(\mu_{A_1}(x), \dots, \mu_{A_n}(x))$, T y T' son operadores de conjunción difusos e I es un operador de implicación.

Puesto que, como comentábamos en la sección anterior, en la mayoría de los casos el Interfaz de Fuzzificación transforma la entrada $x_0 = (x_1, \dots, x_n)$ que recibe el sistema en una serie de conjuntos difusos puntuales A'_1, \dots, A'_n , y debido a la aplicación de las propiedades $T(1, 1) = 1$ y $T(x, 1) = x$, verificadas por las t-normas [GQ91, TV85], la expresión de la Regla Composicional de Inferencia queda finalmente reducida a la forma:

$$\mu_{B'}(y) = I(\mu_A(x_0), \mu_B(y))$$

1.2.4 El Interfaz de Defuzzificación

Del modo de trabajo del Sistema de Inferencia del SBRD de tipo Mamdani descrito en la sección anterior puede extraerse claramente el hecho de que el proceso de inferencia difuso se aplica al nivel de reglas individuales. De este modo, una vez aplicada la Regla Composicional de Inferencia sobre las m reglas que componen la BRL o la BRD, 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, el 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, el 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 modela el operador *también* que relacionaba las reglas de la base:

$$\mu_{B'}(y) = G \{ \mu_{B'_1}(y), \mu_{B'_2}(y), \dots, \mu_{B'_n}(y) \}$$

- 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:* En este segundo modo de trabajo, se 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' , lo que ahorra una gran cantidad de tiempo computacional. Este modo de operación supone una aproximación distinta al concepto representado por el operador *además*.

Históricamente, el primer modo propuesto fue el A, que ya fue empleado por Mamdani en su primera aproximación al control difuso [Mam74]. En los últimos años, la segunda aproximación está siendo más utilizada [CHP97, DHR93, SY93], sobre todo en sistemas en tiempo real que necesitan de una respuesta rápida. En la siguiente sección analizaremos más detenidamente algunos de los operadores de agregación y métodos de defuzzificación existentes.

1.3 Diseño de Sistemas Basados en Reglas Difusas

El buen comportamiento de un SBRD depende directamente de dos factores, la forma en que realiza el proceso de inferencia y la composición de la base de reglas difusas que

maneja (en lo sucesivo, emplearemos el término *base de reglas difusas* cuando queramos referirnos al conjunto de reglas difusas que maneja el sistema, independientemente del tipo y composición de éstas). Por lo tanto, el proceso de diseño de los SBRDs engloba dos tareas principales:

1. *Diseño del Mecanismo de Inferencia*, es decir, la elección de los distintos operadores difusos que se emplearán para realizar el proceso de inferencia.
2. *Obtención de una base de reglas difusas* que contenga información adecuada con respecto al problema que se pretende resolver.

En las dos secciones siguientes analizaremos en profundidad ambas tareas de diseño.

1.3.1 Diseño del Mecanismo de Inferencia

La complejidad de esta primera tarea depende del tipo de SBRD con el que se esté trabajando. Como hemos visto en las secciones anteriores, el mecanismo de inferencia de los SBRDs de tipo Mamdani es más complejo y necesita de la definición de un mayor número de operadores que el empleado en los SBRDs TSK. En este último caso, la única tarea de diseño que es necesario llevar a cabo es la elección del operador de conjunción T empleado para calcular el grado de emparejamiento del antecedente de cada regla con las entradas actuales del sistema, h_i . Este operador se modela habitualmente empleando una t-norma y las elecciones más habituales son el mínimo o el producto.

Cuando se trabaja con sistemas de tipo Mamdani, la complejidad en el diseño del mecanismo de inferencia aumenta en gran medida. En este caso, es necesario definir la composición de tres de los elementos anteriormente introducidos: el Sistema de Inferencia y los Interfaces de Fuzzificación y Defuzzificación. Para ello, se deben tomar las siguientes decisiones [KKS85]:

1. Definir matemáticamente la implicación difusa existente en las reglas lingüísticas o difusas de tipo “*SI-ENTONCES*” contenidas en la base de reglas difusas, es decir, elegir el operador difuso I que se empleará para modelar la implicación.

Existen distintas posibilidades para la elección de este operador. En el primer modelo de SBRD de este tipo, Mamdani empleó la t-norma del Mínimo [Mam74], por lo que varios operadores pertenecientes a esta misma familia han sido posteriormente empleados para dicha tarea [GQ91]. Por otro lado, la familia de las funciones de implicación difusas ofrece una amplia variedad de operadores de implicación clasificados en distintos grupos, dependiendo del modo en que interpretan la implicación difusa [TV85]. Aparte de estas dos familias principales de operadores, algunos autores han sugerido el uso de t-conormas y de operadores de implicación que no pertenecen a ninguna de las familias de operadores difusos conocidas [CK89, CHP97, KKS85].

2. Definir matemáticamente el conectivo *y*, es decir, elegir el operador de conjunción T a usar en el caso de que las reglas de la base presenten más de una variable de entrada. Para esta elección se dispone de distintos operadores de la familia de las t-normas.
3. Definir matemáticamente la composición de relaciones difusas a aplicar en la Regla Composicional de Inferencia. El operador de composición empleado es el *Sup-T*, donde T es una t-norma. Habitualmente, se emplean cuatro tipos de composición basados en otras tantas t-normas: el mínimo, el producto algebraico, el producto acotado y el producto drástico. Las más usadas son las dos primeras [Lee90].
4. Definir matemáticamente el operador de agregación *también*, es decir, elegir el operador G según el modo de defuzzificación que emplee el SBRD. Como ya hemos comentado, en caso de trabajar en modo A, la función del operador de agregación será la de agregar todos los conjuntos difusos individuales resultantes de la inferencia en un único conjunto difuso global. Para este propósito se emplean habitualmente las t-normas y las t-conormas, principalmente el mínimo y el máximo, respectivamente, por su sencillez. En [BD95] se propone una batería de operadores de agregación más sofisticados y se presenta un análisis de sus propiedades, así como un estudio comparativo de su comportamiento.

Por otro lado, en caso de trabajar en modo B, los operadores más habituales son la media, la media ponderada o la selección de un valor característico de los conjuntos difusos, en función de algún grado de importancia de la regla que los ha generado en el proceso de inferencia [CHP97]. Como valores característicos se suelen emplear el *Centro de Gravedad* y el *Punto de Máximo 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 los 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].

5. Definir matemáticamente el método de defuzzificación D a emplear para transformar los conjuntos difusos individuales o globales resultantes del proceso de inferencia en valores precisos de salida. Los más habituales son el *Centro de Gravedad*, el *Centro de Sumas* (aproximación del Centro de Gravedad computacionalmente más rápida de obtener) y la *Media de los Máximos* [DHR93], cuando se trabaja en modo A; y el *Centro de Gravedad* y el *Punto de Máximo Criterio*, al hacerlo en modo B [CHP97].

En la literatura especializada, pueden encontrarse varios estudios que analizan el comportamiento de los operadores difusos existentes para las distintas funciones comentadas. Estos trabajos persiguen aportar, desde diversas perspectivas, una serie de directrices para realizar un buen diseño del Mecanismo de Inferencia de los SBRDs de tipo Mamdani [CK89, CHP97, KKS85].

1.3.2 Obtención de la Base de Reglas Difusas

1.3.2.1 Tareas de diseño para la obtención de la base de reglas difusas

Al igual que en el diseño del Sistema de Inferencia, la generación de la BC o BRD conlleva una serie de tareas de diseño, que varían en función del tipo de SBRD con el que se esté trabajando. Las tareas principales son las siguientes:

1. Selección de las variables relevantes de entrada y salida al sistema, de entre todas las variables disponibles.
2. En el caso de manejar un SBRD con matices descriptivos, ya sea un sistema descriptivo de tipo Mamdani o un sistema TSK que emplee variables lingüísticas en el antecedente de las reglas, definición de la estructura de la BD que contendrá la semántica de los términos que pueden tomar como valor las variables lingüísticas de entrada, en el caso del sistema TSK, o las de entrada y salida, en el caso del de tipo Mamdani. Esto repercute en la aparición de nuevas subtareas de diseño incluidas en ésta:
 - definición de los factores de escala,
 - elección de los conjuntos posibles de términos para cada variable lingüística, lo que permite determinar la granularidad deseada en el sistema,
 - elección del tipo de función de pertenencia a emplear : triangulares, trapezoidales, gaussianas o exponenciales, principalmente [DHR93]. Las dos últimas presentan la ventaja de proporcionar una transición más suave, mientras que las dos primeras la de su simplicidad a la hora de efectuar cálculos computacionales,
 - definición de la función de pertenencia del conjunto difuso concreto asociado a cada etiqueta lingüística.

En lo que respecta a los SBRDs aproximativos, la única tarea a llevar a cabo es la elección del tipo de funciones de pertenencia que se emplearán en las reglas difusas. Se han efectuado distintos estudios para analizar la influencia de la forma de dichas funciones en el comportamiento del SBRD [BFGM93, CHI91], aunque en [DVV97] se demuestra que las funciones trapezoidales pueden aproximar adecuadamente a las demás presentando además la ventaja de su simplicidad.

3. Derivación de las reglas lingüísticas, difusas o de tipo TSK, que compondrán la base de reglas difusas del sistema. Para ello será necesario determinar el número de éstas, así como su composición, mediante la definición del antecedente y el consecuente.

En lo que resta de esta sección supondremos que las variables de entrada y la de salida al sistema han sido previamente determinadas. Esta selección de variables relevantes para

el problema que se está intentando resolver puede ser efectuada por un experto o bien empleando métodos estadísticos, basados en analizar la correlación existente entre las variables de las que se dispone, o combinatorios, que analizan la influencia de conjuntos formados por distintas combinaciones de variables [BD95].

1.3.2.2 Distintas posibilidades para la definición de la base de reglas difusas

Para tomar las restantes decisiones de diseño de la BC o BRD se ha de tener en cuenta la información de la que se dispone y aprovecharla al máximo para obtener un SBRD con el mejor comportamiento posible. En el tipo de aplicaciones de modelado y control en las que se suelen emplear los SBRDs, es posible disponer de dos tipos de información: numérica y lingüística. La primera de ellas se suele obtener a partir de observaciones realizadas sobre el sistema que se desea modelar, mientras que la segunda es aportada directamente a partir de expertos humanos. De este modo, existen dos vías principales para la derivación de la BC o BRD de un SBRD [Wan94]:

1. *Derivación a partir de expertos.* En este primer método, la composición de la BC se efectúa empleando la información experta disponible. El experto o los expertos humanos especifican las etiquetas asociadas a cada variable lingüística, la composición de las reglas de la BRL y la semántica asociada a cada etiqueta. Este primer método es el más sencillo de aplicar en el caso en el que los expertos sean capaces de poner su conocimiento en forma de reglas lingüísticas. Obviamente, sólo puede ser empleado directamente en el caso de trabajar con SBRD descriptivos de tipo Mamdani, aunque como ya hemos indicado anteriormente, es posible transformar reglas lingüísticas en reglas TSK simplificadas para incorporarlas a un sistema difuso de este último tipo.
2. *Derivación a partir de métodos de aprendizaje automático basados en la información numérica existente.*

Aunque la primera vía de definición de la base de reglas difusas es la ideal y fue la primera que se empleó con éxito en algunos problemas, en la mayoría de los casos prácticos, el experto humano no es capaz de expresar su conocimiento sobre el problema en forma de reglas lingüísticas o, simplemente, no existe un experto que pueda aportar información sobre el problema que tratamos de resolver. En otras ocasiones, el experto es capaz de aportar algunas ideas con respecto a la composición de la BRL, es decir, de definir algunas reglas, pero la única información que puede dar sobre la BD se refiere a los dominios de discurso en los que están definidas las variables del problema y las etiquetas lingüísticas necesarias, pero no la forma de las funciones de pertenencia que definen la semántica de estas etiquetas.

Este último problema se resuelve generalmente en el campo de los controladores difusos definiendo una partición difusa primaria de cada variable mediante un proceso de

normalización [DHR93, HMB93, Lee90]. Este proceso consiste en discretizar el dominio de la variable, particionándolo en un número de intervalos igual al número de etiquetas lingüísticas que se consideren, y asociando a cada intervalo un nombre de etiqueta y un conjunto difuso que defina la semántica de ésta. Al no existir conocimiento disponible sobre la forma que deberían presentar dichos conjuntos, lo que se suele hacer es definir una serie de particiones difusas uniformes en las que todos los conjuntos difusos son simétricos y presentan la misma forma. El problema es que este modo de definir la semántica suele acarrear un peor comportamiento del SBRD finalmente obtenido [Lee90].

Los problemas asociados a la derivación de la BC a partir de expertos, en aquellos casos en los que la información experta es insuficiente, han dado lugar a que en los últimos años se hayan desarrollado una gran cantidad de métodos de aprendizaje inductivo para los distintos tipos de SBRDs existentes. Si asumimos que la relación existente entre las variables de entrada y la de salida es lineal o puede ser considerada como tal, estos métodos pueden tomar como base las técnicas clásicas estadísticas de regresión pero esto puede provocar una pérdida de precisión en el sistema obtenido finalmente si dicha suposición no es correcta.

En cambio, cuando se considera también la posibilidad de que existan relaciones no lineales entre las variables del problema, aumenta la complejidad tanto de los aspectos analíticos como de los computacionales en los algoritmos de generación. De este modo, se emplean técnicas de diseño de BCs o BRDs tan diversas como: algoritmos iterativos de generación de reglas basados en el cubrimiento de los ejemplos [WM92] o en la bondad de las reglas con respecto al conjunto de datos existente [BD95, NIT97], variantes del método de mínimos cuadrados [BD95, TS85], métodos de gradiente descendiente [NHW91, NHW92], métodos híbridos entre los dos anteriores [Jan93a], Redes Neuronales [TH91, TSKK92], técnicas de Clustering [YPH93] y Algoritmos Evolutivos [CH95, CHL97a], entre otros. Algunos autores [LAT96, Wan94], se refieren a este tipo de SBRDs que emplean un método automático para el aprendizaje o ajuste de la BC o BRD con el nombre de *Sistemas Difusos Adaptativos*.

Además de manejar las dos técnicas de definición de la BC o BRD por separado, existe la posibilidad de emplear ambas a la vez siempre que se disponga de información de los dos tipos, lingüística y numérica, sobre el problema a resolver. Precisamente, numerosos autores [BD95, Men95, Wan94] destacan la gran ventaja de los SBRDs *al ser los únicos sistemas capaces de combinar a la vez información lingüística y numérica*. No en vano, los SBRDs se pueden construir directamente a partir de conocimiento experto, incorporando la información lingüística en forma de reglas, y, a su vez, pueden hacer uso de la información numérica, en forma de datos de entrada-salida sobre el problema a resolver, en el proceso de ajuste o aprendizaje automático.

Por lo tanto, existen distintas posibilidades de definición de la BC o BRD dependiendo de la información de la que se disponga [BD95]:

1. Los expertos son capaces de definir la BC al completo, es decir, de especificar las etiquetas asociadas a cada variable, los conjuntos difusos que definen la semántica de cada una de estas etiquetas y las reglas lingüísticas que componen la BRL. Aunque pueda parecer extraño, tanto las etiquetas como los conjuntos difusos asociados a las mismas son parte del conocimiento experto. Si un experto proporciona una regla del tipo *Si la temperatura es alta entonces la presión es baja*, debe ser capaz de indicar que significa para él que la temperatura sea alta y la presión baja, especificando las funciones de pertenencia asociadas a ambas etiquetas.
2. Los expertos son capaces de aportar la información anterior en su totalidad o en parte, pero además se dispone de información numérica que es posible emplear para mejorar o completar el conocimiento experto. Pueden plantearse distintas posibilidades:
 - Es posible emplear métodos inductivos de ajuste de la semántica de las etiquetas, es decir, definir la BD a partir de una estimación preliminar de los conjuntos difusos asociados a las mismas, bien aportada por el experto o bien obtenida a partir de una equipartición de los espacios difusos de entrada y salida; y aplicar luego un método automático de ajuste que refine esta definición preliminar basándose en la información existente en el conjunto de datos disponible [Jan93a, Kar91b]. Repasaremos dos métodos de este tipo, basados en el empleo de Algoritmos Evolutivos para llevar a cabo el ajuste, en la sección 2.4.1 del capítulo 2 de esta memoria. Asimismo, presentaremos un nuevo proceso evolutivo de ajuste en el capítulo 3 de la misma.
 - Por otro lado, en algunas ocasiones los expertos son capaces de indicar, desde un punto de vista lingüístico, los posibles estados que se pueden dar en el problema; sin embargo, no son capaces de indicar la acción a llevar a cabo en cada uno de ellos, por lo que los métodos inductivos se emplean para completar la definición de la base de reglas difusas generando los consecuentes desconocidos [BD95].
 - Los expertos pueden también determinar una serie de reglas que dan lugar a una BC incompleta. En este caso, la información numérica puede ser empleada para completar la misma [BD95].
3. Los expertos únicamente son capaces de aportar las variables relevantes para el problema y los términos lingüísticos asociados junto con su semántica, pero no son capaces de determinar las reglas. Si se dispone de información numérica sobre el problema, se hace uso de ella para definir las reglas de la base.
4. No existe conocimiento experto disponible, con lo que es necesario considerar un método de aprendizaje automático basado en la información numérica disponible para construir la BC o BRD.

1.3.2.3 Análisis del proceso de obtención de la base de reglas difusas según el tipo de Sistema Basado en Reglas Difusas empleado

El manejo de los distintos tipos de información disponible y las tareas de diseño para las cuales ésta puede ser empleada dependen del tipo de SBRD que se esté construyendo. Los *sistemas descriptivos de tipo Mamdani pueden manejar ambos tipos*: el conocimiento experto puede ser empleado para definir directamente la BD y la BRL, así como para proporcionar definiciones preliminares que luego pueden ser completadas o mejoradas gracias a la información numérica. Por otro lado, este tipo de sistemas también pueden obtenerse cuando únicamente se dispone de un conjunto de datos de entrada-salida que definen el comportamiento del problema a resolver. En este último caso, se efectúa en primer lugar una definición inicial de la BD a partir de un proceso de normalización, del que resulta una equipartición difusa de los espacios de entrada y salida con la granularidad deseada. Posteriormente, esta definición inicial puede ser mantenida o refinada en el proceso automático de generación de la BRL.

En cambio, en los *SBRDs aproximativos de tipo Mamdani*, la forma de proceder es distinta, ya que la BRD no puede ser definida directamente a partir de conocimiento experto. Aun así, es posible combinar ambos tipos de información o efectuar una definición a partir de un método automático basado en la información numérica. En el primer caso, el conocimiento experto puede ser considerado en el proceso de diseño procediendo de dos modos distintos:

1. Por un lado, es posible emplear este conocimiento para definir una BC descriptiva preliminar y ajustar luego individualmente la semántica de cada regla lingüística obtenida del experto. Mediante este proceso se transforma la BC en una BRD aproximativa teniendo en cuenta la información numérica de la que se dispone [HLV95].
2. Por otro, se pueden incorporar directamente las reglas lingüísticas aportadas por el experto a la BRD que está siendo generada mediante el método automático basado en la información numérica. Para ello, basta con efectuar una simple transformación consistente en sustituir las etiquetas por los conjuntos difusos asociados para convertir la regla difusa descriptiva en aproximativa. El proceso automático de diseño puede presentar también la capacidad de ajustar la definición de los conjuntos difusos de estas reglas [HLV97b], lo que permite obtener sistemas aproximativos con mejor comportamiento.

Finalmente, los *SBRDs de tipo TSK* hacen uso de la información disponible de un modo similar a los aproximativos de tipo Mamdani. Aunque la base de reglas difusas puede ser obtenida directamente a partir de conocimiento experto mediante la transformación de las reglas lingüísticas aportadas por el experto en reglas TSK simplificadas, esto daría lugar a que el sistema así obtenido no presentara buen comportamiento. Además, tendría poco sentido el elegir un sistema TSK cuando la única información de la que se dispone es la aportada por los expertos. Tal y como argumentaban sus propios creadores en [TS85], los

SBRDs TSK están pensados para ser diseñados a partir de información numérica, cuando no se dispone de conocimiento experto o éste es incompleto. En este caso, este último tipo de información puede ser empleada para definir las etiquetas de las variables de entrada al sistema, para especificar la semántica asociada a éstas y para indicar los posibles estados del problema, pero la generación de los parámetros de los consecuentes de las reglas TSK debe ser efectuada empleando un método de aprendizaje automático.

1.4 Propiedades Deseables en una Base de Reglas Difusas

Como ya comentamos en la sección anterior, el comportamiento de un SBRD depende de dos factores, la forma en que realiza el proceso de inferencia y la composición de la BC o BRD que emplea. En lo que respecta al segundo de ellos, en la literatura especializada han ido apareciendo una serie de propiedades deseables en una base de reglas difusas, independientemente de su tipo o del modo en que ha sido obtenida, para que el rendimiento del SBRD sea adecuado [BD95, DHR93, HMB93, Lee90, Ped89]. Dedicaremos cada una de las secciones siguientes a introducir estas propiedades. Como podrá observarse, algunas de ellas no tendrán sentido en el caso de trabajar con reglas difusas de algún tipo concreto, habitualmente reglas de tipo TSK.

1.4.1 Completitud de una Base de Reglas Difusas

Intuitivamente, está claro que un SBRD siempre debería ser capaz de proporcionar una salida ante cualquier entrada que reciba. Esta propiedad se denomina *completitud* y, como puede observarse, no está relacionada con la precisión de la respuesta proporcionada por el sistema sino con el hecho de que ésta se produzca o no.

Desde un punto de vista más formal, la propiedad de completitud en SBRDs no sólo implica que ante una entrada $x_0 \in \mathbf{U}$ que pueda darse realmente en el problema, al menos una de las reglas difusas que componen la base debe devolver como salida un conjunto difuso no vacío sino que, además, la salida difusa global del sistema, es decir, el conjunto difuso obtenido como combinación de las salidas individuales de cada regla, debe ser no vacío [BD95, DHR93]. Como puede observarse, esta definición está pensada para SBRDs cuyo Interfaz de Defuzzificación trabaje en modo A (agregar primero, defuzzificar después) puesto que es en este tipo de sistemas en los que la agregación de una serie de conjuntos difusos individuales no vacíos obtenidos como salida de reglas individuales puede dar lugar a una salida global vacía.

Siguiendo el planteamiento anterior, la propiedad de completitud puede expresarse,

haciendo uso de un valor $\sigma \in (0, 1]$, del siguiente modo:

$$\forall x_0 \in \mathbf{U}, \quad \text{Alt}(S(x_0)) \geq \sigma$$

donde $S(x_0)$ es el conjunto difuso global obtenido como salida del sistema y $\text{Alt}(\cdot)$ representa la altura de un conjunto difuso. Esta propiedad recibe el nombre de σ -completitud y, como puede observarse, es más restrictiva que la definición general introducida en el párrafo anterior puesto que introduce un grado de satisfacción en la completitud.

La propiedad de σ -completitud es de gran utilidad práctica ya que puede ser considerada en el proceso de obtención de la base de reglas difusas. Durante el mismo, se deben seguir añadiendo reglas difusas mientras que una entrada que pueda darse en el problema no esté contemplada en la base o lo esté a un grado inferior al deseado.

1.4.2 Consistencia de una Base de Reglas Difusas

Un conjunto genérico de reglas de tipo “*SI-ENTONCES*” es *consistente* si no contiene contradicciones. La interpretación de este concepto es bastante clara en los sistemas que manejan reglas basadas en Lógica Clásica pero es difícil de analizar en el caso de bases de reglas difusas. De hecho, existen algunas contradicciones en la literatura especializada con respecto a la definición de esta propiedad [DHR93].

En principio, se podría pensar que una base de reglas difusas es inconsistente si contiene reglas que presentan el mismo antecedente y distinto consecuente, pero esta definición genera varias lagunas importantes. Por ejemplo, en [DHR93] los autores se plantean si se pueden considerar inconsistentes dos reglas descriptivas con el mismo antecedente que presentan etiquetas lingüísticas correlativas y solapadas en el consecuente. Así, los autores proponen una definición alternativa en la que dos reglas sólo se consideran inconsistentes cuando presentan el mismo antecedente y sus consecuentes son mutuamente excluyentes.

El estudio de esta propiedad es un campo aún por explorar y parece ser que la forma de afrontarla en el campo de las bases de reglas difusas consiste en relajar su definición introduciendo una gradualidad en la satisfacción de la misma. En [HMB93, Ped89], los autores se basan en la idea de que dos reglas difusas deben ser consideradas como consistentes siempre que existan variaciones similares entre los conjuntos difusos que definen sus antecedentes y consecuentes, y proponen un índice de consistencia para determinar este hecho. En la sección 3.1.1 del capítulo 3 introduciremos otra forma de relajar esta propiedad para su inclusión en procesos inductivos de generación automática de bases de reglas difusas.

1.4.3 Continuidad de una Base de Reglas Difusas

La propiedad de *continuidad* se presenta en [DHR93] aplicada a los SBRDs descriptivos. Para ello, los autores introducen en primer lugar el concepto de reglas vecinas como reglas cuyas celdas son vecinas en la representación en forma de tabla de decisión de la BRL. De este modo, consideran la BC continua siempre que no existan reglas vecinas en las que la intersección de los conjuntos difusos del consecuente sea vacía.

En las BRDs aproximativas, esta propiedad se podría abordar mediante el empleo de un índice similar al utilizado en [Ped89] para medir la inconsistencia, comentado en la sección anterior.

1.4.4 Interacción de una Base de Reglas Difusas

De nuevo, existe cierta confusión en la literatura sobre esta propiedad [DHR93]. Normalmente, el concepto de *interacción* está relacionado con la interpretación de una regla difusa como descriptor de la relación existente entre una serie de variables de entrada y de salida. Suponiendo una regla difusa *Si X es A entonces Y es B*, la cual viene representada por una relación difusa R definida en $\mathbf{U} \times \mathbf{V}$, diríamos que existe interacción en el caso en que la composición de A y R , $A \circ R$, no devuelva B , como podría esperarse [DHR93].

De todos modos, la definición anterior es de poca utilidad en los SBRDs tratados en esta memoria, ya que dichos sistemas manejan entradas reales en lugar de difusas. En estos sistemas, la interacción se define de otra forma [DHR93, HMB93, Lee90, Ped89]. Si la base de reglas difusas está formada por una serie de reglas R_i con la forma *Si X es A_i entonces Y es B_i* , y la entrada que recibe el SBRD es *X es A_i* , entonces cabría esperar que la salida que devuelva sea *Y es B_i* . Lo que ocurre es que realmente la salida puede ser un sub o superconjunto de B_i , dependiendo de la forma en que se hayan definido la composición difusa y el operador de implicación en el Sistema de Inferencia empleado. Este hecho se debe a la interacción que puede darse entre las distintas reglas que componen la base. Por lo tanto, diremos que existe interacción en la base de reglas difusas cuando se da alguno de estos dos últimos casos. En [DHR93, Ped89] se pueden encontrar distintas formulaciones equivalentes de esta propiedad. En la última referencia se presenta además un índice para medir la interacción existente en una base de reglas difusas.

1.4.5 Otras Propiedades asociadas a las Bases de Reglas Difusas

Junto con las cuatro propiedades anteriores, existen otras de menor consistencia teórica pero con un mayor sentido práctico. De entre ellas podemos destacar las dos siguientes:

- *Baja complejidad* [Lee90]: Esta propiedad está relacionada con el número de reglas difusas que forman la base. Es lógico pensar que lo que se pretende es manejar

la menor cantidad de reglas posible, lo que da lugar a una serie de ventajas como, por ejemplo, que el proceso de inferencia sea más rápido. De todos modos, la importancia de esta propiedad depende directamente del tipo de aplicación para la cual se haya diseñado el SBRD. Por ejemplo, en aplicaciones de control, suele primar la rapidez del proceso de inferencia y la simplicidad de la base sobre la precisión de la respuesta, por lo que esta propiedad adquiere una importancia fundamental. Lo mismo ocurre en aplicaciones de modelado cualitativo en las que lo que se persigue es obtener una descripción del sistema real, ya que cuanto menor sea el número de reglas, más claramente interpretable por el ser humano será el modelo obtenido. En cambio, como veremos en la siguiente sección, el objetivo principal en aplicaciones de modelado difuso es que el modelo sea lo más parecido posible al sistema real, con lo que el número de reglas no es un factor excesivamente significativo.

- *Redundancia* [BD95]: La redundancia está directamente relacionada con la propiedad que presentan los SBRDs de que cada estado del sistema pueda estar cubierto por más de una regla debido al solapamiento de los conjuntos difusos de los antecedentes. Esto puede dar lugar a pensar que los SBRDs son, al menos, parcialmente redundantes pero esta suposición no es correcta: reglas cuyas premisas estén completamente cubiertas por otras reglas de la base pueden modificar la salida del sistema.

La existencia de reglas redundantes puede dar lugar incluso a que empeore el funcionamiento global del SBRD, por lo que será necesario estudiar la redundancia que puede darse en la base de reglas difusas con el objetivo de eliminar las reglas innecesarias. Para ello, deberemos analizar la influencia de suprimir una regla de la base sin que empeore el comportamiento del sistema.

En [BD95] se presenta una medida de solapamiento que permite seleccionar aquellas reglas que pueden ser redundantes y un criterio basado en dicha medida que indica cuándo suprimirlas realmente de la base de reglas difusas. En la sección 3.3 del capítulo 3 introduciremos un método para eliminar todas las reglas redundantes de una base de reglas difusas basado en la selección de la combinación de reglas que hace que el sistema presente el mejor funcionamiento posible.

1.5 Aplicaciones de los Sistemas Basados en Reglas Difusas

Como ya indicábamos en la Introducción de este capítulo, las dos principales aplicaciones de los SBRDs son el modelado de sistemas y el control difuso. En esta sección estudiaremos ambos campos de aplicación, analizando las diferencias y similitudes existentes entre ellos, y explicaremos por qué los SBRDs presentan buen comportamiento en los mismos. Asimismo, citaremos algunas aplicaciones concretas dentro de cada grupo.

1.5.1 Diferencias y Semejanzas entre el Modelado y el Control Difuso

De la comparación entre ambas áreas, podemos destacar las siguientes diferencias y semejanzas [BD95]:

1. *Procesos en lazo cerrado y en lazo abierto*: La primera diferencia que existe entre ambos campos es el tipo de procedimiento que se lleva a cabo. El control difuso es un proceso *en lazo cerrado (realimentado) y en tiempo real* consistente en observar en cada instante el valor de las variables de estado del sistema que está siendo controlado y en determinar un nuevo valor de las variables de control que haga que dicho sistema alcance o mantenga el estado deseado. En cambio, el modelado difuso de sistemas es un proceso *en lazo abierto que no tiene por qué llevarse a cabo en tiempo real*.
2. *Tipos de SBRDs empleados*: Mientras que en el control difuso, los SBRDs empleados clásicamente son el descriptivo de tipo Mamdani y, en algunos casos, el TSK; en el modelado difuso, los sistemas más empleados son el aproximativo de tipo Mamdani y el TSK, en ese orden. Únicamente en el modelado cualitativo, que como veremos posteriormente incluye en algunos casos al control difuso, se emplean los SBRDs descriptivos.
3. *Definición de las funciones de pertenencia de los conjuntos difusos manejados en la base de reglas*: En control difuso, generalmente, la definición de las funciones de pertenencia se efectúa a partir de información experta o mediante un proceso de normalización en ausencia de ésta, y se refina posteriormente empleando métodos automáticos o mecanismos de prueba y error en tiempo real basados en ambos casos en el comportamiento en lazo cerrado del proceso. En el modelado difuso, en cambio, esta definición puede efectuarse a partir de distintas vías: información experta, numérica, modelos físicos existentes o cualquier combinación de las anteriores; sin embargo, nunca puede emplearse el método de prueba y error en tiempo real.
4. *Generación de la base de reglas difusas*: Mientras que, como comentábamos en secciones anteriores, las reglas lingüísticas o difusas empleadas en un modelo cualitativo o difuso únicamente pueden obtenerse a partir de la información experta o numérica (o una combinación de ambas), las reglas lingüísticas que componen la BRL de un controlador difuso pueden ser obtenidas a partir de fuentes de información distintas [BD95, Lee90]:
 - A partir de información experta aportada por el operador humano del sistema.
 - A partir de las acciones de control que lleva a cabo dicho operador.
 - A partir de un modelo matemático o difuso del sistema a controlar.
 - A partir de un proceso de aprendizaje automático.

En muchos casos, el modelado de sistemas empleando SBRDs engloba al control difuso ya que casi todos los métodos de diseño de bases de reglas difusas comentados terminan por incluirse dentro de los empleados este campo. El primero de ellos, la derivación a partir de información experta, puede ser considerado como un modelado cualitativo realizado haciendo uso de dicha información, en el que el sistema a modelar es el propio controlador. En lo que respecta al segundo, la derivación a partir de las acciones de control realizadas por el operador humano, suele ser empleado en aplicaciones industriales y, para llevarlo a la práctica, se hace uso de conjuntos de datos operativos de entrada-salida que recogen dichas acciones de control, lo que no es más que un modelado cualitativo del comportamiento del operador. También el cuarto método, la derivación a partir de un proceso de aprendizaje automático, queda reducido a un modelado cualitativo obtenido a partir de información numérica cuando esta información es la que guía al método de aprendizaje.

Por lo tanto, las únicas diferencias se dan en el tercero de los métodos, la derivación a partir de un modelo del sistema, y en el cuarto, en el caso en que el proceso de aprendizaje no esté guiado por datos sino por otros tipos de información, habitualmente también algún tipo de modelo del sistema. Además, aún en el caso de generar la base de reglas difusas empleando información numérica, existe una diferencia principal con respecto a la cantidad de datos disponible. Esta suele ser muy superior en control difuso ya que en este caso suele ser posible realizar una gran cantidad de observaciones tanto a priori como a posteriori, lo que permite que tanto la generación como el refinamiento posterior del SBRD obtenido sean más sencillos. Este ajuste a posteriori está también favorecido por la definición en lazo cerrado del proceso, que permite modificar las reglas a la vista del comportamiento del controlador, lo que no es posible generalmente en el caso de los modelos difusos.

Finalmente, en lo que respecta a la complejidad de la base, ésta debe ser mucho más pequeña en el caso de los controladores difusos, como ya indicábamos en la sección anterior. Aun en el caso en que la velocidad, que suele ser el principal requisito en control, no juegue un papel principal en la aplicación concreta, es conveniente que la base esté compuesta por un número pequeño de reglas difusas ya que este hecho trae consigo otras ventajas tales como la disminución del coste de los dispositivos hardware, factor importante cuando se han de producir controladores difusos en grandes cantidades. En cambio, puesto que el principal requisito en el modelado difuso es la precisión del modelo, el tamaño de la base de reglas difusas no juega un papel importante. En este caso, la precisión de las reglas constituye el criterio principal de diseño y, gracias a la naturaleza del proceso como sistema en lazo abierto, no suelen existir problemas con el tiempo de ejecución que necesita el SBRD para dar una respuesta ante una entrada concreta.

5. *Diseño del Mecanismo de Inferencia:* El papel de este componente del SBRD es mucho más importante en modelado que en control difuso ya que, en muchos casos, son las aplicaciones incluidas en el primer campo las que necesitan de una mayor precisión en la respuesta. La menor importancia del diseño del Mecanismo de Infe-

rencia en control difuso se debe a varias razones pero principalmente al hecho de que, puesto que los controladores difusos presentan una BC muy reducida, la diferencia de funcionamiento existente entre las distintas posibilidades de diseño no es muy significativa. Por otro lado, de nuevo el tema de la velocidad y una mayor sencillez en la implementación en hardware hacen que una gran cantidad de combinaciones sean desechadas directamente por su lentitud o por el coste de su implementación.

1.5.2 Aplicación de los Sistemas Basados en Reglas Difusas al Modelado de Sistemas

1.5.2.1 Justificación del empleo de los Sistemas Basados en Reglas Difusas para modelar sistemas

Tal y como se plantea en [BD95], la primera pregunta que cabe hacerse es: “¿por qué emplear SBRDs para modelar sistemas?”, lo que equivale a decir en la mayoría de los casos: “¿por qué emplear SBRDs para aproximar funciones continuas y no emplear otro tipo de técnicas más clásicas como las de regresión para esta tarea?”. Tradicionalmente, el modo habitual de determinar las relaciones existentes entre un grupo de variables consistía en aplicar un método de regresión para determinar una función, perteneciente a una cierta estructura matemática que verificara la propiedad de cierre, que las aproximara en tanto en cuanto fuera posible. El por qué de emplear funciones de este tipo era una cuestión de funcionamiento: puesto que las soluciones tenían que ser obtenidas mediante cálculos manuales, no era posible usar funciones cuya forma fuese cambiando de estructura durante el proceso de identificación. De este modo, las funciones empleadas presentaban una forma concreta y sus parámetros eran los que tenían que ser estimados mediante algún tipo de técnica (por ejemplo, la de mínimos cuadrados) para aproximar la forma de éstas a los datos disponibles lo máximo posible. Esta forma de proceder no tiene por qué dar lugar a una aproximación razonable y, además, los parámetros de la función no tienen por qué tener una interpretación clara. Que una función presente una expresión analítica más o menos sencilla no tiene por qué implicar que la persona que haga uso de ella conozca la forma que presenta en el espacio. Por ejemplo, incluso cuando se trabaja en el plano, la forma que puede presentar un polinomio de tercer grado es bastante difícil de imaginar.

En cambio, el uso masivo de los ordenadores en los últimos años ha dado lugar a que sea posible desarrollar una descripción que maneje estructuras de otro tipo, lo que presenta una serie de ventajas. Así, las reglas difusas, incluso las aproximativas, tienen siempre una interpretación más clara, ya que cada regla define la función que se desea aproximar en una zona pequeña del espacio. Por lo tanto, un SBRD que trabaja a nivel global está compuesto por una serie de modelos locales, cada una de las reglas difusas que componen su base de reglas difusas, lo que facilita su interpretación ya que permite un análisis por zonas del modelado obtenido. Esta forma de trabajo permite un mayor grado de precisión en la aproximación pero da lugar a que existan más parámetros que estimar.

Por otro lado, el procedimiento de definición de funciones continuas mediante SBRDs es mucho más robusto, ya que un error en la definición de una de las reglas difusas de la base afecta globalmente en un grado muy pequeño a la aproximación obtenida y además lo hace únicamente en el soporte de dicha regla, lo que no ocurre en las funciones cerradas. Por ejemplo, un error en los coeficientes de un polinomio afecta a todo el intervalo en el que está definido y la influencia del error que provoca en la aproximación depende del grado del monomio que presente el coeficiente incorrecto.

Finalmente, como ya hemos indicado, el poder aproximativo de los SBRDs constituye otra ventaja con respecto a las funciones empleadas en las técnicas clásicas. Existen una serie de trabajos que demuestran que distintos tipos de SBRDs, con unas características determinadas, son capaces de aproximar cualquier función bajo unas determinadas condiciones hasta el grado deseado. En este sentido, se dice que los SBRDs son *aproximadores universales* [Buc93, Cas95, Cas96, Kos92, Wan92]. Aunque existen técnicas clásicas, basadas en polinomios y splines, que también verifican esta propiedad, dichas técnicas no son muy empleadas actualmente.

Ahora que ya hemos justificado el uso de los SBRDs para tareas de modelado en contraposición con las técnicas clásicas, el siguiente paso sería analizar si siguen siendo una alternativa interesante a otras técnicas de modelado más actuales, como las *Redes Neuronales* [Lip87, Was93], que han demostrado un buen comportamiento en el campo del modelado de sistemas. En [Wan94] se destacan dos ventajas significativas del uso de los SBRDs sobre el de las Redes Neuronales en tareas de modelado:

- Los parámetros de los SBRDs tienen un sentido real claro, es decir, dichos parámetros determinan las funciones de pertenencia de los conjuntos difusos que manejan las reglas del sistema, independientemente del tipo de éstas, con lo que se conoce en cada momento qué quiere decir el valor de uno determinado. Debido a esto, es posible hacer una buena inicialización de los mismos cuando se pretende estimarlos mediante un método automático basado en la información numérica disponible. Sin embargo, en el caso de las Redes Neuronales, concretamente en el de los *Perceptrones Multicapa*, los parámetros no tienen una relación clara con los datos de entrada-salida, lo que provoca que su inicialización tenga que ser realizada aleatoriamente. Este hecho repercute directamente en la disminución de la velocidad de convergencia de los métodos de entrenamiento de redes, ya que la mayoría están basados en técnicas de gradiente descendiente.
- Por otro lado, los SBRDs proporcionan un marco adecuado para incorporar el conocimiento experto disponible sobre el sistema que se desea modelar. Como hemos comentado en secciones anteriores, este conocimiento puede ser empleado para dar una definición inicial de la base de reglas difusas, definición que será luego mejorada haciendo uso de la información numérica existente. Las Redes Neuronales, en cambio, sólo pueden hacer uso de este último tipo de información a la hora de definir el modelo, lo que puede suponer un inconveniente en aquellos casos en los que se

disponga de una cantidad pequeña de datos pero de bastante conocimiento experto sobre el sistema a modelar.

Las dos ventajas comentadas se centran en el proceso de diseño del modelo. Podemos destacar, además, una última ventaja una vez que éste ya ha sido obtenido: la interpretabilidad. Incluso en los SBRDs aproximativos, que presentaban el menor poder de descripción de todos los existentes, la capacidad de interpretación del modelo difuso es superior al neuronal ya que se es capaz de determinar cuáles son las reglas difusas que están actuando ante una determinada entrada, por lo que se puede conocer el comportamiento del modelo según la zona del espacio en la que se encuentre la entrada que recibe.

Antes de finalizar esta sección, es importante destacar que las afirmaciones anteriores no son ciertas para todos los tipos de Redes Neuronales. Por ejemplo, no son correctas para las *Redes de Base Radial*, ya que se ha demostrado que estas redes son equivalentes a los SBRDs TSK [JS93].

1.5.2.2 Relación del modelado difuso con la identificación de sistemas

Existe una relación directa entre el modelado difuso y la identificación de sistemas. Esta última técnica, perteneciente al campo de la Teoría de Sistemas, se refiere al proceso de obtención de un modelo equivalente al sistema que se desea modelar con respecto a los datos de entrada-salida disponibles.

De este modo, los conceptos empleados clásicamente en el campo de la identificación de sistemas pueden ser trasladados directamente al del modelado difuso de sistemas. Existen dos pasos en el proceso de identificación [Ped96, SY93]: la *identificación de la estructura* y la *identificación de los parámetros*. Esto quiere decir que, para identificar un sistema es necesario, en primer lugar, determinar la estructura que presentará el modelo del mismo y, en segundo, estimar los parámetros para que el modelo obtenido bajo la estructura fijada, se ajuste lo mejor posible a la información numérica de la que se dispone. Cada uno de los dos pasos del proceso de identificación ha de resolver una serie de problemas:

1. *Identificación de la estructura*: La obtención de la estructura del modelo incluye dos tareas: la *determinación de las variables de entrada al modelo*, que a su vez incluye la obtención del conjunto de variables del sistema candidatas a ser variables de entrada al modelo y la selección del subconjunto de éstas que afecta realmente a la salida del sistema en mayor medida; y la *determinación de las relaciones existentes entre las variables de entrada y salida* mediante algún tipo de lenguaje de descripción.
2. *Identificación de los parámetros*: Habitualmente, esta segunda tarea conlleva la estimación de los coeficientes de un modelo funcional de un sistema.

No siempre es necesario llevar a cabo todas las tareas comentadas en los dos puntos anteriores. Por ejemplo, algunas veces, se conocen a priori las variables de entrada y

salida del modelo, con lo que la única tarea a poner en práctica en el primer paso es la determinación de las relaciones existentes entre ellas. Otras, la estructura del modelo está predeterminada, con lo que basta con estimar los parámetros del mismo.

Como puede observarse, existe una analogía entre las tareas de identificación comentadas y las tareas de obtención de la base de reglas difusas, presentadas en la sección 1.3.2.1 [SY93, Ped96]:

1. En la identificación de la estructura, la primera tarea coincide directamente, ya que también está contemplada en el diseño de SBRDs. La segunda, la determinación de las relaciones existentes entre las variables, puede ser asociada con la obtención del número de reglas que forman la base y la composición de éstas (determinación de los antecedentes y consecuentes de cada una de ellas). Además, en el caso de los SBRDs descriptivos, esta segunda tarea incluiría también el determinar la partición de los subespacios difusos de entrada y salida.
2. En lo que respecta a la identificación de los parámetros, la equivalencia es directa con la definición de los mismos en las funciones de pertenencia consideradas en las reglas difusas de cualquiera de los tres tipos de sistemas (Mamdani descriptivo, Mamdani aproximativo y TSK) y de los parámetros existentes en los consecuentes de las de tipo TSK. La única diferencia que existe entre ambos casos radica en el número de parámetros, que es muy superior en los SBRDs.

Así, desde un punto de vista muy general, se podría plantear una equivalencia entre la identificación de la estructura y la obtención de la BRL, y la de los parámetros y la de la BD, en los SBRD descriptivos de tipo Mamdani o en los TSK que emplean variables lingüísticas en el antecedente [SY93]. Esta asociación no es tan directa en el caso de los aproximativos.

Por último, destacaremos que en el caso del modelado difuso, se da la ventaja adicional de que la identificación de la estructura y de los parámetros puede ser realizada por separado (podemos, por ejemplo, definir primero la BRL y luego ajustar su semántica), lo que facilita el proceso de identificación o diseño del modelo difuso o cualitativo. En [SY93] puede consultarse un método de identificación de modelos difusos que sigue esta filosofía, en la que también estarán basados los métodos que presentaremos en los capítulos 4, 5 y 6 de esta memoria.

1.5.2.3 Algunas aplicaciones concretas del modelado difuso de sistemas

Algunos ejemplos prácticos de aplicación en el campo del modelado difuso o cualitativo de sistemas son los siguientes:

- *Aplicaciones al campo de la economía:* Dentro de este campo se han desarrollado aplicaciones para aconsejar en las operaciones comerciales de cambio de moneda

extranjera mediante la predicción de los valores de cambio [Tan95, YYY+91], para predecir la tendencia del mercado de stocks en Shangai en combinación con Redes Neuronales [YG94], para predecir series temporales financieras estadounidenses [Ben94] y para detectar fraudes en las demandas de las compañías de seguros que trabajan con seguros de salud, también en los Estados Unidos [Cox95], entre otras.

- *Aplicaciones a la predicción atmosférica:* Predicción de las precipitaciones y de la temperatura media diaria [BD95].
- *Aplicaciones a la predicción de la demanda de agua y a la gestión de embalses* [BD95].
- *Aplicaciones a la medicina:* En [BD95] puede encontrarse una aplicación al diagnóstico médico en la que el modelo difuso actúa como método de combinación de los diagnósticos de varios médicos. Por otro lado, en [LT96], los autores presentan una aplicación al diagnóstico de una enfermedad dental.
- *Aplicación a la valoración de la calidad del arroz* [NIT97].

1.5.3 Aplicación de los Sistemas Basados en Reglas Difusas al Control de Sistemas

Al igual que hacíamos en la sección anterior con respecto al modelado, lo primero que cabe preguntarse es por qué usar los SBRDs para el control de sistemas. Es conocido el hecho de que el interés que existe actualmente en la Teoría de Conjuntos Difusos y la Lógica Difusa se debe casi en su totalidad a la aplicación tan exitosa que han tenido los controladores difusos en una gran cantidad de productos de consumo y sistemas industriales. En [Wan94], se justifica este éxito desde dos perspectivas, una teórica y otra práctica. Las razones teóricas del éxito del control difuso se deben a los siguientes aspectos:

- Como regla general, una buena técnica de ingeniería debe ser capaz de hacer un uso eficiente de toda la información disponible. La información en control puede provenir de dos fuentes distintas: los sensores, que proporcionan medidas numéricas de las variables del sistema, y los expertos humanos, que proporcionan descripciones lingüísticas sobre el sistema y las instrucciones de control. Los controladores difusos, como SBRDs, constituyen una de las pocas herramientas capaces de incorporar ambos tipos de información, con lo que pueden ser aplicados aún en la ausencia de una de ellas.
- En general, el control difuso es una aproximación independiente del modelo, es decir, no necesita un modelo del sistema a controlar, al igual que varios esquemas clásicos de control tales como el control PID. Las aproximaciones independientes del modelo facilitan el diseño del controlador, ya que la obtención de un modelo matemático de

un sistema es a veces una tarea bastante complicada. Este hecho ha dado lugar a que la importancia de estos enfoques esté aumentando en el campo de la ingeniería de control.

- Los controladores difusos son lo bastante generales como para llevar a cabo cualquier acción de control no lineal (como hemos visto en secciones anteriores, los SBRDs son aproximadores universales), con lo que pueden ser aplicados en el control de sistemas que presenten una dinámica no lineal.

Mientras que las razones anteriores se refieren a la generalidad y el rigor, aspectos exigidos siempre desde un punto de vista teórico, la vertiente práctica suele centrarse en la posibilidad de aplicación. Las razones prácticas del uso creciente del control difuso son las siguientes:

- El control difuso es muy sencillo de entender. Puesto que los controladores difusos emulan la estrategia de control del ser humano, son fáciles de comprender para los no especialistas en control. Esto ha dado lugar a que aumente su aplicación en contraposición a las técnicas clásicas de control que se basan en un complicado aparataje matemático.
- La implementación hardware de los controladores difusos es sencilla y rápida, además de permitir un alto grado de paralelización.
- Desarrollar controladores difusos es barato. Desde un punto de vista práctico, el costo de desarrollo es un factor fundamental para obtener un producto con éxito empresarial. El control difuso es sencillo de entender y puede ser aprendido en un tiempo muy corto, con lo que el "costo software" es bajo. Al ser fácil de implementar, el "costo hardware" también lo es. Todas estas razones dan lugar a que el control difuso sea una técnica con una buena relación rendimiento/coste.

Finalmente, en lo que respecta a las aplicaciones concretas, recordaremos que la primera aplicación industrial de los controladores difusos fue la planta de fabricación de cemento danesa F. L. Smith en 1979 [UK80]. Entre otras más recientes, podemos seleccionar el control de una planta purificadora de agua, del sistema de combustión de una planta incineradora, de la fermentación del licor japonés sake, de ascensores, del sistema de ventilación de un túnel de autopista, de una lavadora totalmente automática, del vuelo de una lanzadera espacial, de la velocidad y la transmisión en automóviles, de reactores nucleares, de sistemas de potencia eléctrica, el sistema de conducción automática de trenes, la ecualización no lineal de canales de comunicación en transmisiones de datos de alta velocidad, etc. [BD95, Ber92, Bon94, Hir93, Lee90].

Capítulo 2

Sistemas Basados en Reglas Difusas Evolutivos

En los últimos años, la investigación en el campo de los Sistemas Difusos ha evolucionado hasta dar lugar a un marco de trabajo más general que contempla la integración de la Lógica Difusa con otras técnicas tales como los Algoritmos Evolutivos, las Redes Neuronales y el Razonamiento Probabilístico. Esta nueva área recibe el nombre de *Computación Flexible*¹ [Bon97].

En concreto, la combinación de la Lógica Difusa y los Algoritmos Evolutivos ha obtenido unos resultados muy prometedores en los últimos años [CHL96b, CHL97a, CHL97b, HV96, Ped97, SSZ97, Tet95]. La definición automática de un Sistema Difuso puede ser afrontada como un proceso de optimización o búsqueda, y los Algoritmos Evolutivos [Bäc96], en particular los Algoritmos Genéticos [Gol89, Hol75, Mic96], están considerados en la actualidad como la técnica de búsqueda global más conocida y empleada. Estos algoritmos presentan la capacidad de explorar y explotar espacios complejos, lo que les permite obtener soluciones muy próximas a la óptima dentro en dichos espacios. Además, la codificación genética que emplean les permite incorporar conocimiento a priori de una forma muy sencilla y aprovecharlo para guiar la búsqueda. Todas estas razones han incrementado el empleo de los Algoritmos Evolutivos para el diseño de Sistemas Difusos a lo largo de los últimos años, lo que ha dado lugar a la creación de los *Sistemas Difusos Evolutivos* [CH95, HM97].

Un Sistema Difuso Evolutivo es un método de diseño automático de Sistemas Difusos basado en Algoritmos Evolutivos. Dentro de este gran grupo, los más conocidos y estudiados son los *Sistemas Basados en Reglas Difusas Evolutivos* (SBRDEs) [CH95, GH97, HM97], en los que se hace uso de los Algoritmos Evolutivos para obtener de un modo automático la totalidad o una parte de la base de reglas difusas del SBRD. Existen distintos tipos según que la tarea de diseño que lleven a cabo sea la definición de las funciones de pertenencia de los conjuntos difusos de las reglas (la Base de Datos (BD), en el caso

¹El término castellano *Computación Flexible* es una traducción del término inglés *Soft Computing*

de los sistemas descriptivos de tipo Mamdani o TSK), la obtención de las propias reglas (la Base de Reglas lingüísticas (BRL), en el caso de los sistemas descriptivos de tipo Mamdani) o ambas a la vez (la Base de Conocimiento (BC) completa en el caso de los sistemas descriptivos de tipo Mamdani o la Base de Reglas Difusas (BRD) en el caso de los aproximativos).

Este segundo capítulo está dedicado al estudio de este tipo de sistemas. Para ello, introduciremos primero la *Computación Evolutiva*, estudiando brevemente los tipos de Algoritmos Evolutivos más empleados. Posteriormente, analizaremos la aplicación clásica de los Algoritmos Evolutivos al aprendizaje de Sistemas Basados en Reglas. Finalmente, nos centraremos en el estudio de los SBRDEs, presentando los tipos existentes, la problemática de diseño asociada y algunos ejemplos.

2.1 Computación Evolutiva

La *Computación Evolutiva* (CE) se basa en el empleo de modelos de procesos evolutivos para el diseño e implementación de sistemas de resolución de problemas basados en el ordenador. Los distintos modelos computacionales que se han propuesto dentro de esta filosofía suelen recibir el nombre genérico de *Algoritmos Evolutivos* (AEs) [Bäc96]. Existen cuatro tipos de AEs bien definidos que han servido como base a la mayoría del trabajo desarrollado en el área: los *Algoritmos Genéticos* (AGs) [Gol89, Hol75], las *Estrategias de Evolución* (EEs) [BS95, Sch95], la *Programación Evolutiva* (PE) [FOW66, Fog91] y la *Programación Genética* (PG) [Koz92].

Un AE se basa en mantener una población de posibles soluciones del problema a resolver, llevar a cabo una serie de alteraciones sobre las mismas y efectuar una selección para determinar cuáles permanecen en generaciones futuras y cuáles son eliminadas. Aunque todos los modelos existentes siguen esta estructura general, existen algunas diferencias en cuanto al modo de ponerla en práctica. Los AGs se basan en operadores que tratan de modelar los operadores genéticos existentes en la naturaleza, como el cruce y la mutación en un punto, los cuales son aplicados a los individuos que codifican las posibles soluciones. En cambio, las EEs y la PE aplican transformaciones basadas en mutaciones efectuadas sobre los padres para obtener los hijos, lo que permite mantener la línea general de comportamiento del individuo en su descendencia. Finalmente, la PG codifica las soluciones al problema en forma de programas, habitualmente codificados en una estructura de árbol, y adapta dichas estructuras empleando operadores muy específicos.

Cada individuo de la población recibe un valor de una medida de adaptación que representa su grado de adecuación al entorno. La selección hace uso de estos valores y se centra en los individuos que presentan mayor valor en la medida. Los operadores de recombinación y/o mutación alteran la composición de dichos individuos, guiando heurísticamente la

búsqueda a través del espacio. Aunque simples desde un punto de vista biológico, este tipo de algoritmos son suficientemente complejos para proporcionar mecanismos de búsqueda adaptativos muy robustos. Los mismos procedimientos pueden ser aplicados a problemas de distintos tipos, sin necesidad de hacer muchos cambios [Fog95].

En las secciones siguientes introduciremos dos de los AEs más conocidos, los AGs y las EEs.

2.1.1 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 han sido aplicados con mucho éxito en problemas de búsqueda y optimización. La razón de gran parte de este éxito se debe a su habilidad para explotar la información que van acumulando sobre el espacio de búsqueda que manejan, desconocido inicialmente, lo que les permite redirigir posteriormente la búsqueda hacia subespacios útiles. La *capacidad de adaptación* que presentan es su característica principal, especialmente en espacios de búsqueda grandes, complejos y con poca información disponible, en los que las técnicas clásicas de búsqueda (enumerativas, heurísticas, ...) no presentan buenos resultados.

La idea básica de estos algoritmos consiste en mantener una población de individuos que codifican soluciones del problema. Dichos individuos emplean una representación genética para codificar los valores de las características parciales que definen las distintas soluciones. Debido a ello, cada individuo recibe el nombre de *cromosoma*, y cada una de sus componentes el de *gen*.

Los cromosomas son generados 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 AEs, 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.

Aunque existen muchas variantes posibles del AG básico, el proceso genérico de funcionamiento está formado por las tres operaciones siguientes:

1. Evaluación de la adaptación de los individuos,
2. creación de una población intermedia empleando un mecanismo de selección y
3. alteración de la estructura de los componentes de esa población mediante los operadores de cruce y mutación.

La figura 2.1, en la que $P(t)$ denota la población en la generación t , muestra la estructura general de un AG básico.

```

Procedimiento Algoritmo Genetico
comienzo (1)
     $t = 0$ ;
    inicializar  $P(t)$ ;
    evaluar  $P(t)$ ;
    Mientras (No condicion de parada) hacer
        comienzo (2)
             $t = t + 1$ ;
            seleccionar  $P'(t)$  a partir de  $P(t - 1)$ ;
            cruzar y mutar  $P'(t)$ ;
             $P(t) = P'(t)$ ;
            evaluar  $P(t)$ ;
        fin (2)
    fin (1)
  
```

Figura 2.1. Estructura básica de un algoritmo genético

En las secciones siguientes, analizaremos distintos aspectos relacionados con los AGs para acabar resumiendo las ideas comentadas en otra sección que recogerá las características principales de estos AEs.

2.1.1.1 Representación y evaluación de las soluciones

El esquema de representación o codificación es un factor clave en la aplicación de los AGs, ya que éstos manipulan directamente una representación codificada del problema y, en consecuencia, el esquema escogido puede limitar de una forma muy severa la ventana desde la cual el AG afronta el problema. Existen distintos esquemas de generales codificación entre los que destacan los siguientes:

1. La *codificación binaria*: Es la más antigua de todas las existentes [Hol75, Gol89]. Se basa 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 [HLV97c], 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 problema del viajante de comercio y del coloreo 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.

Las dos últimas representaciones introducidas requieren de operadores genéticos especiales para trabajar con los cromosomas, ya que los operadores clásicos de cruce y mutación, que serán presentados en la sección 2.1.1.3, bien no tienen aplicación directa, o bien alteran la composición de los cromosomas descendientes de tal modo que las soluciones que codifican no pertenecen al conjunto de soluciones posibles del problema. En dicha sección también presentaremos algunos de los operadores específicos para trabajar con cromosomas con codificación real.

En lo que respecta a la función de adaptación, su importancia es también notable ya que es la encargada de guiar al AG por el espacio de búsqueda. Por esta razón, dicha función debe estar bien diseñada para que sea capaz, no sólo de distinguir de un modo claro los individuos bien adaptados de los que no lo están, sino también de ordenar éstos en función de su capacidad para resolver el problema.

2.1.1.2 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 2.2). El número de veces que es copiado 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' .

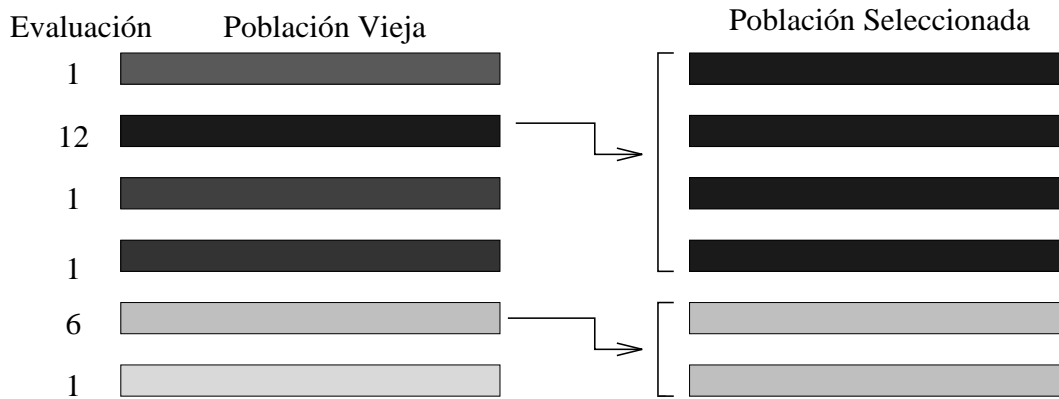


Figura 2.2. Ejemplo de aplicación del mecanismo de selección

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 son seleccionados 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 el o 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].

2.1.1.3 Los operadores de cruce y mutación

Una vez formada la población intermedia mediante la aplicación del mecanismo de selección, llega el momento de aplicar los operadores genéticos de cruce y mutación para alterar la composición de los descendientes que formarán la nueva población.

El operador de cruce: Este operador constituye un mecanismo para compartir información entre cromosomas. Combina las características de dos cromosomas padre para

obtener dos descendientes, con la posibilidad de que los cromosomas hijo, obtenidos mediante la recombinación de sus padres, estén mejor adaptados que éstos. No suele ser aplicado a todas las parejas de cromosomas de la población intermedia sino que se lleva a cabo una selección aleatoria en función de una determinada probabilidad de aplicación, la *probabilidad de cruce*, P_c .

El operador de cruce juega un papel fundamental en el 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 2.3), o el *cruce multipunto*, que procede como el anterior pero trabajando sobre dos o más puntos de cruce.

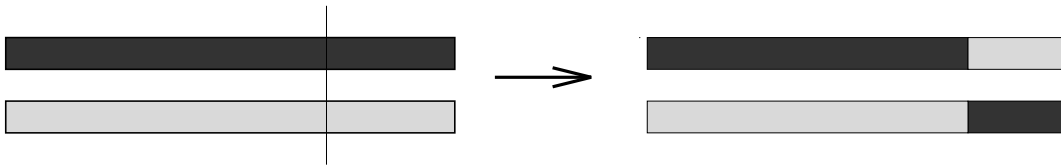


Figura 2.3. Ejemplo de aplicación del operador de cruce simple en un punto

También se pueden emplear ambos operadores cuando se trabaja con el esquema de codificación real, aunque existe una serie de operadores diseñados para su uso específico con esta representación [HLV97c]. Entre estos, destacaremos una familia de operadores que manejan técnicas basadas en Lógica Difusa para mejorar el comportamiento del operador de cruce [HLV97a]. 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 $P(t)$, $C_w^t = (c_1, \dots, c_k, \dots, c_H)$ y $C_v^t = (c'_1, \dots, c'_k, \dots, c'_H)$, 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.

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 y como recoge la figura 2.4.

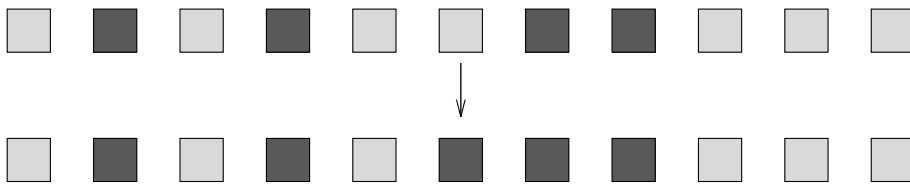


Figura 2.4. Ejemplo de aplicación del operador de mutación

Este operador puede ser trasladado al campo de los AGs con codificación real, de forma que el nuevo valor del gen mutado sea escogido 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 [HLV97c, 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

$$c'_k = \begin{cases} c_k + \Delta(t, c_{kd} - c_k), & \text{si } a = 0 \\ c_k - \Delta(t, c_k - c_{ki}), & \text{si } a = 1 \end{cases}$$

donde a 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.

2.1.1.4 Algoritmos Genéticos para funciones multimodales. Algoritmos Genéticos con nichos

Como hemos comentado en las secciones anteriores, los AGs son conocidos por su capacidad para llevar a cabo procesos de búsqueda en espacios complejos. Aun así, las versiones más generales de este tipo de algoritmos pueden no trabajar de un modo adecuado cuando el espacio de búsqueda es multimodal y presenta muchos óptimos locales. En estos casos, los AGs simples se caracterizan por converger hacia la zona del espacio donde se encuentran el mejor o los mejores óptimos locales, abandonando la búsqueda en las zonas restantes. Este fenómeno se denomina *deriva genética*¹ [DG89] y se debe de evitar por dos razones principales:

- En primer lugar, esta convergencia hacia el que se cree el óptimo más prometedor del espacio puede dar lugar a que finalmente no se alcance el óptimo global debido a que éste no se encuentre en esa zona y el AG no sea capaz de saltar a aquella en la que realmente está situado.
- Por otro lado, debido a esta forma de trabajo, la necesidad que existe en algunos problemas de conocer la localización de varios óptimos de la función no puede ser satisfecha por el AG.

Los conceptos de *nicho* y *especie* fueron introducidos con objeto de evitar este tipo de comportamiento [DG89, Gol89, Hol75]. Como la gran mayoría de los conceptos básicos en el campo de los AGs, ambos son aspectos habituales de la naturaleza que han sido trasladados a dicha área. En la naturaleza, un nicho es una tarea que lleva a cabo un individuo en el entorno y una especie es un grupo de individuos con características similares. La separación del entorno y de los individuos que explotan dicho entorno en distintos subconjuntos es muy común en la naturaleza. De este modo, los AGs con nichos se basan en provocar la formación de subpoblaciones de individuos estables (especies) que exploran zonas parciales del espacio de búsqueda (nichos), obligando a los individuos similares a compartir los recursos disponibles entre ellos.

Una de las formas más habituales para provocar la formación de especies y la creación de nichos se basa en el esquema de *compartición de valores de adaptación entre individuos* [DG89, Gol89], en el que la población del AG se divide en varias subpoblaciones (especies) de acuerdo a la similaridad existente entre los individuos. Estas subpoblaciones crean nichos en dos espacios solución posibles: el de los genes y el de los parámetros

¹El término castellano *deriva genética* es una traducción del término inglés *genetic drift*

decodificados, *compartición genotípica y fenotípica*, respectivamente. Tal y como ocurre en la naturaleza, los individuos de cada nicho comparten la recompensa asociada a dicho nicho entre ellos. Para esa tarea, se define una *función de compartición* que determina la vecindad y el grado de compartición para cada individuo de la población.

La función de compartición está basada en un parámetro σ que determina el radio de los nichos que se definen en cualquiera de los dos espacios solución. Este parámetro está encargado de definir el grado de compartición entre los distintos individuos, que varía en el intervalo $[0, 1]$. Una de las posibles funciones de compartición es la denominada *ley de la potencia*, que presenta la siguiente expresión:

$$P(d) = \begin{cases} 1 - (\frac{d}{\sigma})^\alpha, & \text{si } d < \sigma \\ 0, & \text{en otro caso.} \end{cases}$$

donde d es un valor concreto de una métrica $d(C_i, C_j)$ que determina la distancia existente entre dos individuos cualesquiera. Así, la función trabaja penalizando la recompensa (valor de la función de adaptación original) asociada al cromosoma debido a la presencia de otros individuos en su vecindario. El parámetro real α es el encargado de indicar el grado de penalización a aplicar.

De este modo, el valor de adaptación que se asocia finalmente a cada individuo en un problema de maximización se obtiene dividiendo el valor potencial, procedente de la función de adaptación sin tener en cuenta la compartición, entre el grado de compartición acumulado del siguiente modo:

$$F(C_i) = \frac{f(C_i)}{\sum_{j=1}^n P(d(C_i, C_j))}$$

Tal y como se indica en [DG89], el buen funcionamiento de este esquema de nichos depende directamente del valor del parámetro σ , por lo que su elección debe ser realizada con mucho cuidado. Su valor debe ser igual a la máxima distancia necesaria entre los cromosomas para que se formen el mismo número de nichos que de picos en el espacio de búsqueda multimodal. Por esta razón, el cálculo del valor de σ es un problema complejo si no se conoce bien la forma del espacio solución que se está explorando. En [DG89] se presentan dos métodos alternativos para el cálculo de este valor según se trabaje con un esquema de compartición fenotípico o genotípico.

2.1.1.5 Características principales de los Algoritmos Genéticos

La aplicación de un AG a un problema conlleva el análisis y diseño de los cinco componentes siguientes:

1. Una representación genética de las posibles soluciones al problema.

2. Un mecanismo de creación de la población inicial de soluciones.
3. Una función de adaptación que asocie un valor de adecuación a cada cromosoma.
4. Una serie de operadores genéticos que alteren la composición genética de los descendientes durante el proceso de reproducción, así como un mecanismo de selección de los padres que intervienen en dicho proceso.
5. Una serie de valores para los parámetros del AG: tamaño de la población, probabilidades de aplicación de los operadores genéticos, etc.

Los AGs han sido aplicados con éxito en una gran cantidad de áreas distintas. Algunas de las razones de este éxito son las siguientes:

1. Pueden resolver problemas muy complejos de una forma rápida y sencilla.
2. Pueden ser enlazados fácilmente con simulaciones y modelos existentes.
3. Presentan gran facilidad para ser hibridizados con otras técnicas.

En resumen, *los AGs son muy robustos*. Destacan por su buen comportamiento en problemas difíciles en los que el espacio de búsqueda es grande, discontinuo, complejo y poco conocido. Aunque no garantizan la obtención de la solución óptima al problema, suelen proporcionar soluciones bastante aceptables en un tiempo razonable.

Todas las razones anteriores han dado lugar a que su aplicación haya crecido enormemente a lo largo de los últimos años en campos tan diversos como la optimización de funciones numéricas y la optimización combinatoria [Gol89, Mic96], la Inteligencia Artificial [BB91, For93], la Investigación Operativa [Nis93], las Redes Neuronales [WS92], los Sistemas Difusos [HV96, Ped97], la Robótica [Dav91], la Vida Artificial [BB91], el Aprendizaje Automático [DSG93, Jan93b], etc.

2.1.2 Estrategias de Evolución

Las EEs [BS95, Sch95] fueron propuestas inicialmente por Rechenberg y Schwefel en 1964 con la idea de obtener sistemas que fueran capaces de resolver problemas complejos de optimización con parámetros reales. La representación natural escogida fue un vector de genes con valores reales que era manipulado empleando operadores de mutación que alteraban los valores de los parámetros reales de un modo útil. Por lo tanto, el operador dominante en las EEs es el de mutación.

En la actualidad, existen distintos tipos de EEs, que varían según el número de individuos que manejan y los operadores que emplean. Aunque la mutación sigue siendo el operador fundamental, a raíz de la aparición de las EEs con más de dos individuos se

han desarrollado otra serie de operadores que combinan las características de los padres de un modo similar a como lo haría el operador de cruce de los AGs. En las dos secciones siguientes introduciremos brevemente dos tipos de EEs, la $EE - (1 + 1)$ y la $EE - (\mu, \lambda)$.

2.1.2.1 La Estrategia de Evolución $(1 + 1)$

La primera variante de EE que se propuso, la denominada $EE - (1 + 1)$, estaba basada en el manejo de sólo dos individuos por generación, un padre y un descendiente. El algoritmo hace evolucionar el vector padre aplicando un determinado operador de mutación a cada uno de sus componentes. La fuerza de la mutación que se efectúa viene determinada por un valor σ , la desviación típica de una variable aleatoria distribuida según una normal, que suele denominarse *tamaño de paso*. Este parámetro está asociado al padre y se adapta al mismo tiempo que éste en cada paso del algoritmo. Si la adaptación realizada ha tenido éxito, es decir, si el hijo está mejor adaptado que el padre según el valor de ambos en la función de adaptación, entonces el hijo sustituye al padre en la siguiente generación. El proceso se repite hasta que se verifica una determinada condición de parada, habitualmente que el proceso se estabilice durante un número de generaciones determinado.

El operador de mutación **mut** está formado por dos componentes. El primero de ellos, **mu $_{\sigma}$** , es el encargado de adaptar el valor de la desviación típica σ , mediante la denominada *regla de éxito $\frac{1}{5}$ de Rechenberg*:

$$\sigma' = \mathbf{mu}_{\sigma}(\sigma) = \begin{cases} \frac{\sigma}{\sqrt[n]{c}}, & \text{si } p > \frac{1}{5} \\ \sigma \cdot \sqrt[n]{c}, & \text{si } p < \frac{1}{5} \\ \sigma, & \text{si } p = \frac{1}{5} \end{cases}$$

donde n es la dimensión del vector solución, p es la frecuencia relativa de mutaciones efectuadas con éxito y c es una constante que determina la cantidad en la que se actualiza σ .

El segundo, **mu $_x$** , es el que lleva a cabo la mutación de los componentes del vector de números reales sumando a cada uno de ellos un valor aleatorio z_i distribuido según una normal de media 0 y desviación típica σ' :

$$x' = \mathbf{mu}_x(x) = (x_1 + z_1, \dots, x_n + z_n)$$

donde $z_i \sim N(0, \sigma'^2)$.

2.1.2.2 La Estrategia de Evolución (μ, λ)

Esta segunda variante de EE se basa en hacer evolucionar una población de μ vectores solución para obtener λ descendientes y seleccionar los μ mejores para formar la siguiente población (obviamente, λ ha de ser mayor o igual que μ). En este caso, los descendientes se

obtienen recombinando en primer lugar uno o más padres en un único vector de variables objeto y creando después uno nuevo a partir de éste aplicando mutaciones normales con las mismas o distintas desviaciones típicas a cada variable objeto. La característica principal de este algoritmo es su habilidad para incorporar los parámetros más importantes de la estrategia (desviaciones típicas y coeficientes de correlación de las distribuciones normales que definen la mutación) al proceso de búsqueda, de modo que la adaptación no sólo tenga lugar en las variables objeto sino también en los parámetros de la EE de acuerdo a la topología local del espacio de búsqueda. Esta propiedad se denomina *auto-adaptación*¹ [Bäc96, Sch95].

De este modo, cada individuo \vec{a} de la población está formado por tres vectores distintos, $\vec{a} = (\vec{x}, \vec{\sigma}, \vec{\alpha})$, los cuales representan los valores de las variables objeto, de las desviaciones típicas y de los ángulos de rotación, respectivamente. El vector \vec{x} tiene dimensión n , siendo n el número de variables del problema a resolver. El segundo vector, $\vec{\sigma}$, tiene dimensión n_σ , que puede valer como máximo n (en ese caso, cada variable objeto x_i , $i = 1, \dots, n$, tiene asociado un tamaño de paso σ_i distinto). Finalmente, el valor máximo que puede tomar n_α , la dimensión de $\vec{\alpha}$, es $\frac{(2 \cdot n - n_\sigma) \cdot (n_\sigma - 1)}{2}$. El conjunto de parámetros de la estrategia, formado por las desviaciones típicas y los ángulos de rotación, define completamente la distribución n -dimensional generalizada con media $\vec{0}$. Sin embargo, es posible dar valor cero al parámetro n_α , indicando así que los ángulos de rotación no se consideran en el proceso. Los valores más habituales para ambos parámetros son [BS95, Bäc96]: $(n_\sigma, n_\alpha) = \{(1, 0), (n, 0), (n, \frac{n \cdot (n-1)}{2}), (2, n-1)\}$.

El algoritmo que presentamos a continuación describe de una forma genérica el modo de trabajo de la $EE - (\mu, \lambda)$. El parámetro t especifica el número de la generación en la que se encuentra el algoritmo y $P(t)$ representa la población que se maneja en dicha generación:

1. Inicializar y evaluar $P(0)$. Inicializar el contador de generaciones $t \leftarrow 0$
2. Tomar ζ de los μ individuos de $P(t)$ y recombinarlos λ veces, empleando cualquiera de los siguientes mecanismos de recombinación de genes, $r \in \{0, 1, 2, 3\}$, $i = 1, \dots, n + n_\sigma + n_\alpha$:

$$a'_i = \begin{cases} a_{S,i} ; S \sim U(\{1, \dots, \zeta\}) \text{ igual } \forall i, & r = 0 \\ \frac{\sum_{j=1}^{\zeta} a_{j,i}}{\zeta}, & r = 1 \\ u \cdot a_{S,i} + (1 - u) \cdot a_{T,i} ; S, T \sim U(\{1, \dots, \zeta\}), & r = 2 \\ a_{S,i} ; S \sim U(\{1, \dots, \zeta\}), & r = 3 \end{cases}$$

Esta operación permite obtener los λ individuos que forman $P'(t)$. El nombre de los distintos mecanismos es: sin recombinación ($r=0$), recombinación global intermedia ($r=1$), recombinación local intermedia ($r=2$) y recombinación discreta ($r=3$).

¹El término castellano *auto-adaptación* es una traducción del término inglés *self-adaption*

3. Mutar $P'(t)$ adaptando los λ individuos para obtener λ descendientes que pasan a formar $P''(t)$. Para ello, proceder del siguiente modo con cada uno de ellos:

- 3.1. Mutar los valores de $\vec{\sigma}'$ para obtener el vector mutado $\vec{\sigma}''$:

$$\vec{\sigma}'' = (\sigma'_1 \cdot \exp(z_1 + z_0), \dots, \sigma'_{n_\sigma} \cdot \exp(z_{n_\sigma} + z_0))$$

donde $z_i \sim N(0, \frac{1}{\sqrt{2 \cdot \sqrt{n}}})^2$, $i = 1, \dots, n_\sigma$ y $z_0 \sim N(0, \frac{1}{\sqrt{2 \cdot n}})^2$.

- 3.2. Mutar los valores de $\vec{\alpha}'$ para obtener el vector mutado $\vec{\alpha}''$:

$$\vec{\alpha}'' = (\alpha'_1 + z_1, \dots, \alpha'_{n_\alpha} + z_{n_\alpha})$$

donde $z_i \sim N(0, 0.0873^2)$ ($0.0873 \approx 5^\circ$), $i = 1, \dots, n_\alpha$.

- 3.3. Mutar los valores de \vec{x}' para obtener el vector mutado \vec{x}'' :

$$\vec{x}'' = (x'_1 + \text{cor}_1(\vec{\sigma}'', \vec{\alpha}''), \dots, x'_n + \text{cor}_n(\vec{\sigma}'', \vec{\alpha}''))$$

donde $\text{cor}(\vec{\sigma}'', \vec{\alpha}'')$ es un vector aleatorio de valores correlacionados normalmente distribuidos.

4. Evaluar $P''(t)$ y seleccionar los μ mejores individuos para formar $P(t+1)$.
5. Aumentar en 1 el contador de generaciones $t \leftarrow t+1$
6. Si se da la condición de parada, entonces terminar, en otro caso, volver a 2.

2.2 Aprendizaje de Sistemas Basados en Reglas mediante Algoritmos Evolutivos

Aunque los AEs no son algoritmos de aprendizaje, ofrecen una metodología de búsqueda potente e independiente del dominio que puede ser aplicada a gran cantidad de tareas de aprendizaje. De hecho, los AEs han sido muy utilizados en problemas de aprendizaje automático de reglas de producción a partir de conjuntos de ejemplos [Gre94]. Esto se debe a que los individuos, cuando representan conocimiento, pueden ser tratados al mismo tiempo como datos que son manipulados por el AE y como código ejecutable que lleva a cabo cierta tarea [Mic96].

Existen tres enfoques de aplicación de los AEs al aprendizaje de reglas. De ellos, dos son clásicos y han sido muy usados a lo largo de los últimos años, mientras que el tercero

ha aparecido recientemente tratando de combinar las mejores características y de resolver los distintos problemas de los dos anteriores. Los dos primeros enfoques, *Michigan* y *Pittsburgh*, propuestos respectivamente en [Boo82, HR78] y [Smi80], se diferencian principalmente en el esquema de representación. Mientras que en el enfoque Michigan cada cromosoma codifica una única regla y la solución final al problema está formada por la última población obtenida por el AE al completo, en el enfoque Pittsburgh, los cromosomas corresponden a bases de reglas completas, por lo que la solución al problema de aprendizaje está compuesta finalmente por el mejor cromosoma de la última población. En cambio, en el tercero de los enfoques, el de *Aprendizaje Iterativo de Reglas* (AIR) [GH97, Ven92], cambia un poco la filosofía de trabajo. Al igual que en el enfoque Michigan, cada cromosoma codifica una regla individual, con lo que el proceso evolutivo se hace a nivel de regla y no de base de reglas, pero el único individuo considerado para formar parte de la solución final es el mejor adaptado de la última población. Como veremos posteriormente, este modo de trabajo requiere que el AE sea ejecutado varias veces para completar la solución final al problema de aprendizaje.

En las siguientes secciones analizaremos más detenidamente cada enfoque y presentaremos una pequeña comparativa entre ellos, en la cual analizaremos en qué tipo de problema es más adecuado cada uno de ellos.

2.2.1 El Enfoque Michigan

Los procesos de aprendizaje evolutivo basados en el enfoque Michigan, propuestos inicialmente por Holland, reciben el nombre de *Sistemas Clasificadores*. Este tipo de Sistemas Basados en Reglas presentan mecanismos capaces de procesar reglas en paralelo, generar nuevas reglas mediante procesos adaptativos y probar la eficiencia de las reglas existentes [Mic96].

Como ya hemos comentado, en este primer enfoque cada cromosoma representa una única regla y la base de reglas final está formada por la última población del AE al completo. Este conjunto de reglas es adaptado a lo largo de las distintas generaciones mediante su interacción con el entorno gracias a la acción de una serie de operadores evolutivos que trabajan a nivel de regla y de otra serie de módulos como los de *asignación de mérito*¹ y descubrimiento de reglas.

Aunque existe una amplia variedad estructural y de funcionamiento en los sistemas basados en este enfoque, la organización general de los mismos presenta en la mayoría de los casos la siguiente estructura formada por tres componentes:

1. El *sistema de rendimiento*, encargado de interactuar con el entorno, que contiene la base de reglas y el sistema de producción.

¹El término castellano *asignación de mérito* es una traducción del término inglés *credit assignment*

2. El *sistema de asignación de mérito*, encargado de llevar a cabo el proceso de aprendizaje mediante la modificación de la recompensa asociada a cada regla individual con respecto al comportamiento global del conjunto global de reglas contenidas en la población. Un ejemplo de un sistema de este tipo lo constituye el algoritmo *bucket brigade* presentado por Holland en [Hol86].
3. El *sistema de descubrimiento de reglas*, que genera nuevas reglas individuales a partir de la base global mediante un AE.

En [BGH89] puede encontrarse una descripción más amplia de este tipo de sistemas.

2.2.2 El Enfoque Pittsburgh

El segundo de los enfoques clásicos fue propuesto por De Jong y sus estudiantes [Smi80] mientras trabajaban en la Universidad de Pittsburgh, razón por la cual recibe dicho nombre. Se basa en la codificación de una base de reglas completa en cada individuo existente en la población del AE y en la aplicación de los operadores evolutivos con el objeto de obtener nuevas poblaciones formadas por nuevas bases de reglas candidatas. De este modo, la recompensa o crédito, siguiendo la nomenclatura empleada en el enfoque Michigan, es asociada al conjunto completo de reglas de acuerdo a su interacción con el entorno.

En el caso de emplear un AG para el aprendizaje automático del sistema, el operador de cruce suele ser el encargado de obtener nuevas combinaciones de reglas y el de mutación, de generar reglas distintas a las anteriores. En algunos casos, se trabaja con cromosomas de longitud variable, lo que conlleva la necesidad de emplear operadores genéticos modificados capaces de manejar este tipo de cromosomas.

En [Gre94] se pueden consultar varios sistemas evolutivos de aprendizaje basados en este enfoque.

2.2.3 El Enfoque de Aprendizaje Iterativo de Reglas

En este último enfoque, al igual que en el Michigan, cada individuo de la población codifica una única regla pero, al contrario que éste, sólo el mejor individuo es considerado finalmente para formar parte de la solución, la base de reglas final, siendo descartados los restantes de la última población del AE. De este modo, en el modelo iterativo, el AE soluciona parcialmente el problema del aprendizaje. Para obtener una base de reglas completa, que solucione finalmente el problema, es necesario que el AE sea incluido en un esquema iterativo del tipo del que se muestra a continuación:

1. Ejecutar el AE para obtener una regla para el sistema.

2. Incorporar dicha regla a la base de reglas final.
3. Penalizar de algún modo la regla obtenida.
4. Si la base de reglas obtenida hasta el momento es suficiente para constituir una solución al problema, parar devolviendo dicha base como salida. En otro caso, volver al paso 1.

Una forma muy sencilla y directa de penalizar las reglas que van siendo generadas cuando se está trabajando en un problema de aprendizaje inductivo consiste en eliminar del conjunto de entrenamiento todos aquellos ejemplos que ya estén cubiertos por la base de reglas obtenida hasta el momento.

Como puede observarse, este modo de trabajo provoca la formación de nichos y especies en el espacio de búsqueda (véase la sección 2.1.1.4). De hecho, en [BBM93] se propone un método iterativo de optimización de funciones multimodales, la denominada *Técnica de Nichos Secuenciales*, que permite obtener el número de óptimos deseado empleando un AG con nichos que trabaja de una forma muy similar. El inducir nichos en el espacio de búsqueda parece muy adecuado para el problema del aprendizaje de conceptos, si se considera dicho proceso como el aprendizaje de conceptos multimodales.

Por otro lado, puesto que la generación de reglas se efectúa sin tener en cuenta la cooperación global de la base de reglas obtenida, los sistemas evolutivos de aprendizaje que siguen esta filosofía de trabajo suelen incluir un proceso de postprocesamiento encargado de simplificar dicha base de reglas, por lo que finalmente acaban por presentar una estructura en dos niveles: un proceso de generación y otro de postprocesamiento.

Dentro del AE, la principal diferencia que existe con respecto al enfoque Michigan es que la adaptación de cada cromosoma se calcula individualmente, sin tener en cuenta ningún tipo de cooperación con los demás individuos de la población. Esta forma de trabajo reduce substancialmente el espacio de búsqueda ya que en cada ejecución del AE únicamente se busca el obtener una regla individual.

En [GH97] puede encontrarse una descripción más completa de este último enfoque, así como de varios sistemas evolutivos de aprendizaje automático basados en él.

2.2.4 Análisis de los Tres Enfoques

Cada uno de los tres enfoques presentados tiene asociadas una serie de ventajas e inconvenientes. En lo que respecta al Michigan, ha demostrado ser el más adecuado para su aplicación en procesos de aprendizaje en línea. Este enfoque es más flexible a la hora de trabajar con dominios cambiantes y con procesos de aprendizaje incremental (en los que aparecen nuevos ejemplos de entrenamiento según avanza el tiempo), mientras que el Pittsburgh y el AIR parecen ajustarse mejor a dominios estáticos y al aprendizaje fuera de

línea, en el que todos los ejemplos de entrenamiento están disponibles desde el momento en que comienza el proceso de aprendizaje.

El problema principal del enfoque Michigan es resolver el conflicto existente entre los intereses individuales y colectivos de los clasificadores que forman el sistema. El objetivo final de un Sistema Clasificador es obtener del proceso evolutivo una base de reglas que cooperen adecuadamente para resolver un problema. En un sistema evolutivo de aprendizaje basado en este enfoque, en el que tanto el mecanismo de selección como el de reemplazamiento trabajan a nivel de regla individual, las reglas han de cooperar para proporcionar buenas salidas y recibir así una buena recompensa, además de competir con las demás reglas existentes en la población del AE para poder reproducirse en la siguiente generación.

Este conflicto entre los dos tipos de intereses no se da en los sistemas evolutivos de aprendizaje que siguen el enfoque Pittsburgh, ya que en éstos el proceso de reproducción se hace a nivel de bases de reglas completas, en lugar de a nivel de reglas individuales. Aunque esto constituye una ventaja, acarrea el inconveniente del alto costo computacional (en términos de memoria y tiempo de cálculo) necesario para mantener y evaluar una población formada por una serie de bases de reglas completas. Con ello, los inconvenientes que presenta este último enfoque son, al menos, iguales a las ventajas. Aunque evita el problema de la competición entre clasificadores, necesita de una gran cantidad de recursos computacionales para evaluar una población de bases de reglas.

En cuanto al enfoque AIR, la ventaja que presenta frente a los dos anteriores, es que el primer nivel reduce considerablemente el espacio de búsqueda al perseguir la generación de una sola regla en cada ejecución del AE, lo que permite obtener buenas soluciones en poco tiempo. Aun así, este último enfoque también presenta la desventaja de necesitar muchos recursos de máquina ya que es necesario ejecutar un gran número de veces el AE.

2.3 Sistemas Basados en Reglas Difusas Evolutivos

2.3.1 ¿ Qué es un Sistema Basado en Reglas Difusas Evolutivo ?

Tal y como comentábamos en el capítulo anterior, una de las dos tareas principales de diseño de SBRDs es la obtención de la base de reglas difusas. La problemática existente para generar ésta a partir de expertos ha dado lugar a la aparición de una gran cantidad de métodos de aprendizaje automático que hacen uso de la información numérica (y lingüística, en algunos casos) existente para llevar a cabo esta tarea. Los AEs han demostrado ser una herramienta muy potente para automatizar la definición de la base de reglas difusas, ya que el control adaptativo, el aprendizaje y la auto-organización pueden

ser considerados en muchos casos como procesos de optimización o búsqueda. Este hecho ha dado lugar a que el empleo de estos algoritmos para diseñar SBRDs se haya extendido durante los últimos años. Este tipo de procesos reciben el nombre genérico de *SBRDEs* [CH95, GH97, HM97].

De este modo, un SBRDE es un proceso basado en AEs que lleva a cabo algunas o todas las tareas de diseño asociadas a la obtención de la base de reglas difusas de un SBRD de un modo automático. La figura 2.5 representa gráficamente un sistema de este tipo.

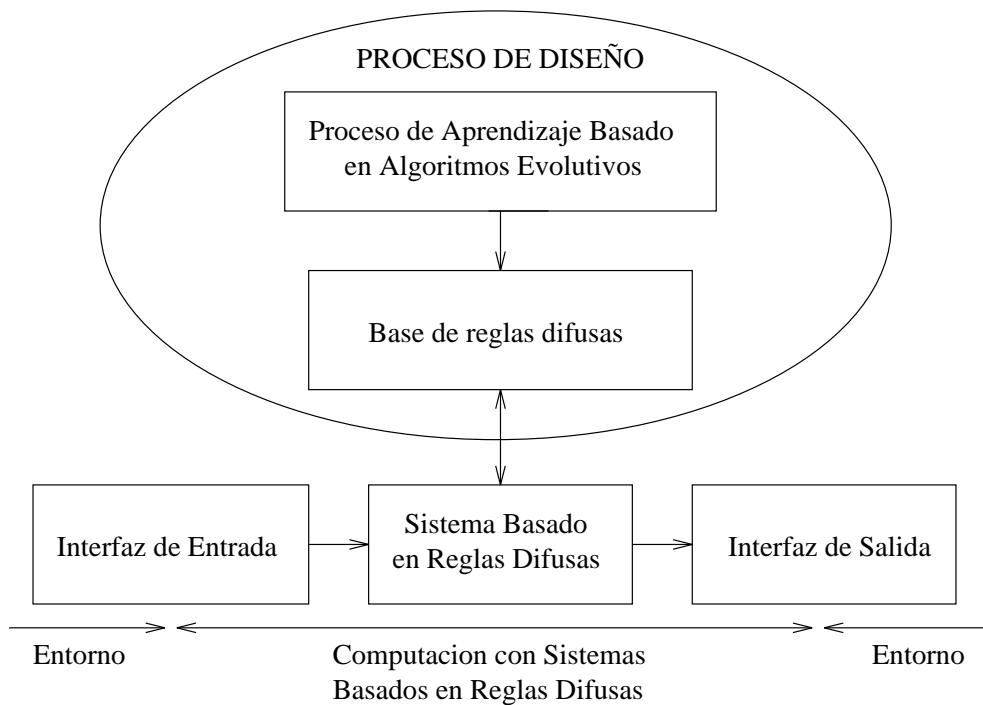


Figura 2.5. Sistemas Basados en Reglas Difusas Evolutivos

2.3.2 Tipos de Sistemas Basados en Reglas Difusas Evolutivos

Es posible clasificar los SBRDEs en distintas familias, según las tareas de diseño asociadas a la obtención de la base de reglas difusas que lleven a cabo [CH95]. Principalmente, los SBRDEs se han aplicado a tareas de definición de los factores de escala y de las funciones de pertenencia de los conjuntos difusos empleados en las reglas (es decir, tareas asociadas a la definición de la BD cuando se trabaja con SBRDs descriptivos) y a la generación de las propias reglas difusas de la base. De este modo, en el caso de los SBRDs descriptivos de tipo Mamdani, que constituyen la familia de SBRDs a la que más se han aplicado los AEs en tareas de diseño de la BC, es posible distinguir tres tipos de SBRDEs:

1. Los que llevan a cabo la *definición evolutiva de la Base de Datos*.

2. Los que desarrollan un *aprendizaje evolutivo de la Base de Reglas Lingüísticas*.
3. Los que desarrollan un *aprendizaje evolutivo de la Base de Conocimiento* al completo.

En [CH95] se introducen varios ejemplos concretos de cada familia, algunos de los cuales serán presentados al final de este capítulo. En [CHL96a] y [CHL97a] se recopila una gran cantidad de referencias bibliográficas sobre el tema, mientras que en la sección 13 de [CHL97b] se puede encontrar una selección de referencias clasificadas de cada familia.

La clasificación introducida varía ligeramente en los dos tipos de SBRDs restantes. En el caso de los SBRDs aproximativos de tipo Mamdani, únicamente existen dos familias, ya que la segunda de las presentadas pierde su sentido en este tipo de sistemas. Los dos grupos posibles son el formado por los SBRDEs que ajustan la forma de los conjuntos difusos empleados en las reglas de una BRD preliminar, los denominados procesos de ajuste; y el formado por los SBRDEs que aprenden directamente la composición de la BRD, los procesos de aprendizaje.

En cambio, en los SBRDs TSK, las tres familias siguen teniendo sentido pero con pequeñas modificaciones. La primera mantiene la filosofía original pero, bien es aplicada únicamente a la parte antecedente de las reglas, o bien ajusta también los parámetros existentes en el consecuente de las mismas. El segundo grupo está formado por aquellos SBRDEs que, tomando como base unas particiones prefijadas para las variables de entrada, aprenden la composición de los antecedentes y los consecuentes de las reglas difusas TSK que formarán la base. Finalmente, la tercera familia está compuesta por los SBRDEs de tipo TSK que llevan a cabo las dos tareas anteriores.

En las secciones siguientes, analizaremos un poco más en profundidad las tres familias comentadas.

2.3.2.1 Definición evolutiva de la Base de Datos

La definición o ajuste de los factores de escala y las funciones de pertenencia de los conjuntos difusos son tareas importantes en el diseño de SBRDs. Existe la posibilidad de parametrizar ambas definiciones y adaptarlas haciendo uso de un AE que maneje dichos parámetros de acuerdo a una función de adaptación.

La función de los factores de escala es establecer una correspondencia tanto entre las entradas que recibe el SBRD y el universo normalizado en el que se definen las funciones de pertenencia; como entre las salidas que devuelve el Sistema de Inferencia, definidas en dicho universo, y el dominio real de las salidas. Este proceso de escalado se efectúa mediante funciones de escala, que pueden ser lineales o no lineales. Es posible parametrizar dichas funciones y afrontar su definición mediante AEs. En [GGP97, Mag96], se pueden encontrar algunos ejemplos de SBRDEs que llevan a cabo esta tarea.

En lo que respecta al ajuste de las funciones de pertenencia, se han propuesto varios

métodos para llevar a cabo la definición de la BD empleando AEs [BN95, HLV95, Kar91b, SKG93]. En todos ellos, cada cromosoma manejado en el proceso evolutivo representa una definición diferente del conjunto completo de funciones de pertenencia consideradas en las reglas difusas de la base. Existen dos posibilidades, según la naturaleza del SBRD sea descriptiva o aproximativa: bien codificar la partición difusa de cada variable lingüística manteniendo el nivel de descripción del sistema, tal y como se hace en [BN95, Kar91b, SKG93], o bien codificar las funciones de pertenencia empleadas en cada regla difusa, ajustando los parámetros de las mismas de un modo local para cada conjunto difuso, con lo que se obtiene un SBRD aproximativo, aún en el caso en el que el sistema inicial fuese descriptivo. Esta última posibilidad es considerada en [HLV95].

En la sección 2.4.1, describiremos dos métodos evolutivos de ajuste basados en cada una de las dos filosofías, los presentados en [HLV95] y [Kar91b].

2.3.2.2 Aprendizaje evolutivo de la Base de Reglas Lingüísticas

Como ya hemos indicado, esta segunda familia únicamente tiene sentido cuando se trabaja con SBRDs con matices descriptivos, ya sean de tipo Mamdani o TSK. Todas las aproximaciones existentes en este grupo, suponen la existencia de una definición previa de la BD, es decir, de un conjunto de funciones de pertenencia que definen la semántica de los términos lingüísticos existentes. Esta definición puede haber sido obtenida a partir de conocimiento experto o mediante una equipartición difusa de los espacios de entrada y salida (éste último sólo en el caso de los SBRDs de tipo Mamdani), tal y como comentábamos en el capítulo anterior. De este modo, la tarea del AE en estos procesos automáticos de diseño es la de generar una base de reglas difusas adecuada para el problema a resolver.

Muchos de los SBRDEs incluidos en este grupo aprenden únicamente los consecuentes de las reglas difusas que forman la base. Para ello, hacen uso de la representación en forma de *tabla de decisión* de la base de reglas difusas. Una BRL de un SBRD descriptivo de tipo Mamdani formada por reglas con n variables en el antecedente y sólo una en el consecuente puede ser representada mediante una tabla de decisión n -dimensional en la que cada dimensión corresponde a una variable de entrada. Cada dimensión tendrá asociado un vector que contendrá las posibles etiquetas lingüísticas de la variable correspondiente y cada casilla de la tabla almacenará la etiqueta que toma como valor la variable de salida ante la combinación de entradas que la generan. Es decir, cada casilla de la tabla representará una regla lingüística de la base.

Las primeras aproximaciones que se realizaron, por ejemplo las presentadas en [HT94, Kar91a, KB93, Thr91], se limitaban a codificar la base de reglas en forma de tabla de decisión en el cromosoma, por lo que el proceso evolutivo sólo se llevaba a cabo sobre los consecuentes de las reglas, siguiendo un enfoque Pittsburgh. En muchos casos, la única diferencia que existía entre ellos era la capacidad de aprender también el número de reglas óptimo para la base. Esta característica la presentaban únicamente aquellos métodos que incluían un valor que representaba la ausencia de consecuente para una regla con un

antecedente concreto, es decir, la ausencia de la regla correspondiente en la BRL. En el caso en que el AE no dispusiera de dicho valor de código, todas las reglas de la tabla eran automáticamente incluidas en la BRL aunque no fuesen necesarias. De los cuatro métodos citados, los presentados en [Kar91a, KB93] tienen este inconveniente mientras que los dos restantes, los incluidos en [HT94, Thr91], sí poseen la capacidad de decidir el número de reglas.

Posteriormente, se han desarrollado otros SBRDEs más refinados dentro de esta familia. En [Bon93], Bonarini presenta un proceso evolutivo de diseño basado en una variante del enfoque Michigan para el aprendizaje de la BRL, por lo que no codifica tablas de decisión, sino reglas individuales en cada individuo de la población. Por otro lado, el SBRDE presentado por Hoffmann y Pfister en [HP95] mantiene el enfoque Pittsburgh, pero aporta una serie de novedades tales como el empleo de una estructura jerárquica de inferencia, y un AG desordenado¹ [Gol89] que posibilita el manejo de cromosomas de longitud variable con codificación no dependiente de la posición. Este esquema de codificación permite representar la BRL de una forma más adecuada, lo que facilita su diseño.

En la sección 2.4.2, presentaremos un método basado en tablas de decisión [Thr91] y el método de Bonarini [Bon93].

2.3.2.3 Aprendizaje evolutivo de la Base de Conocimiento

Esta última familia ha sido la más prolífica en número de contribuciones desarrolladas en los últimos años. Podemos encontrar aproximaciones muy diversas para la definición completa de la base de reglas difusas, tales como las presentadas en [CV93, GP96, HLV97b, KKK94, KS95, LP94, LT93b, MV96, NL94, PB93, VM95a, VM95b]. Dentro de este gran grupo de trabajos, hay procesos que presentan cromosomas de longitud variable [CV93, LP94], algunos que hacen uso de una gran cantidad de conocimiento experto con objeto de mejorar el proceso de aprendizaje [HLV97b, LT93b], mientras que otros fijan el número de reglas a priori [KS95, PB93].

Por otro lado, Carse y sus colegas [CFM96] dividen esta tercera familia en dos subgrupos dependiendo de la simultaneidad en el desarrollo de las dos tareas de diseño consideradas: la definición de las funciones de pertenencia y de las reglas de la base. Principalmente, distinguen entre procesos que aprenden la base de reglas completa en varios pasos (como los presentados en [GP96, HLV97b, KKK94]), o simultáneamente en un único paso (caso de los introducidos en [CV93, KS95, LT93b, LP94, MV96, NL94, PB93, VM95a, VM95b]).

En la sección 2.4.3, introduciremos tres SBRDEs incluidos en esta familia: el presentado por Lee y Takagi en [LT93b] para el diseño de SBRDs TSK, y los dos presentados por Magdalena y Velasco en [MV96, VM95a, VM95b], basados respectivamente en los enfoques Michigan, para diseñar SBRDs aproximativos de tipo Mamdani, y Pittsburgh, para sistemas descriptivos del mismo tipo.

¹El término castellano *Algoritmo Genético desordenado* es una traducción del término inglés *Messy Genetic Algorithm*

2.3.3 El Problema de Cooperación-Competición

Al igual que cualquier otra técnica de diseño de SBRDs, el objetivo de un SBRDE es la obtención de un SBRD capaz de resolver el problema planteado de la mejor forma posible. Para llevar a cabo esta tarea, los SBRDEs combinan en el proceso de diseño las principales características del sistema a obtener, un SBRD, y de la herramienta empleada para ello, un AE.

Como vimos en el primer capítulo, una de las características más interesantes de un SBRD es el razonamiento interpolativo que lleva a cabo. Esta característica juega un papel primordial en el buen funcionamiento de los SBRDs, y es consecuencia de la *cooperación entre las reglas que componen la base de reglas difusas del sistema*. Esto se debe a que la salida que proporciona el SBRD no suele ser fruto de la acción de una única regla sino de la acción conjunta de varias reglas difusas que se han activado al emparejarse sus antecedentes con la entrada del sistema en algún grado.

Por otro lado, el aspecto fundamental de los AEs es la *competición entre los miembros de la población que representan posibles soluciones al problema*. En este caso, este comportamiento se debe a la mecánica de la selección natural en la que se basa el AE.

De este modo, los SBRDEs trabajan *promoviendo la competición para obtener finalmente la mejor cooperación posible*, al combinar las dos características anteriores. Esta forma de trabajo parece muy adecuada para afrontar el diseño de SBRDs, ya que los distintos miembros de la población compiten entre ellos para proporcionar la solución final: la base de reglas difusas que presente el mejor nivel de cooperación posible entre las reglas que la constituyen. El problema que se plantea es cómo llevar a la práctica la forma de trabajo indicada y se denomina *problema de cooperación-competición* (PCC) [Bon96].

La dificultad en la resolución del problema anterior depende directamente del enfoque evolutivo de aprendizaje seguido por el SBRDE. Los tres enfoques genéricos de aprendizaje de Sistemas Basados en Reglas introducidos en la sección 2.2 pueden ser empleados sin alteración ninguna para la construcción de SBRDEs, y cada uno de ellos presenta una problemática distinta a la hora de resolver el PCC que analizamos brevemente a continuación.

En el **enfoque Michigan**, la resolución del PCC es bastante difícil. El hecho de que el proceso evolutivo sea efectuado a nivel de regla individual en los SBRDEs basados en este enfoque, da lugar a que no sea fácil obtener un nivel de cooperación adecuado entre reglas difusas que están compitiendo entre sí. Para ello, es necesario disponer de una función de adaptación capaz de medir tanto la bondad de la regla individual, como la calidad de su cooperación con las restantes de la población para proporcionar la mejor acción posible en la salida. Como se indica en [Bon96], no es nada fácil diseñar una función de adaptación de este tipo.

Por otro lado, los SBRDEs que siguen el **enfoque Pittsburgh** son capaces de resolver correctamente el PCC. En este caso, el AE lleva a cabo la evolución sobre bases de reglas

difusas completas, y el valor de la función de adaptación asociado a cada individuo se calcula teniendo en cuenta la salida que devuelve realmente el SBRD codificado en el cromosoma, al recibir una determinada entrada. De este modo, cada vez que se evalúa la adaptación de un individuo, se mide el nivel de cooperación existente entre las reglas que componen la base de reglas difusas que codifica, por lo que el SBRDE es capaz de adaptar correctamente la población para conseguir que las reglas difusas de la base obtenida finalmente cooperen de la mejor forma posible.

Desafortunadamente, este enfoque presenta el inconveniente de tener que trabajar con espacios de búsqueda muy grandes, lo que dificulta el encontrar soluciones óptimas. Los parámetros del SBRD considerados en el proceso de aprendizaje automático condicionan la dimensión y las propiedades del espacio de búsqueda, dando lugar a que la resolución del problema sea más lenta o más rápida o, incluso, factible en menor o mayor grado. Como se indica en [SK94], las investigaciones realizadas han demostrado que la longitud del cromosoma y la complejidad del problema juegan un papel fundamental en estos factores.

Este inconveniente es más habitual en el diseño de SBRDEs pertenecientes a la tercera familia comentada, es decir, cuando el proceso evolutivo de aprendizaje afronta la generación de la base de reglas difusas completa. En estos casos, la representación genética debe incluir una gran cantidad de parámetros distintos para definir la base de reglas difusas, lo que provoca que el tamaño de los individuos aumente en gran medida. Este hecho se acentúa aún más cuando se afronta el diseño de SBRDs aproximativos. El uso de definiciones distintas de las funciones de pertenencia en cada regla difusa da lugar a que el número de parámetros de la base de reglas difusas se incremente, por lo que el espacio de búsqueda se hace más complejo, provocando incluso que el problema pueda llegar a ser computacionalmente intratable.

En [Bon96], Bonarini analiza detenidamente el PCC y concluye que las reglas difusas que cooperan adecuadamente para producir un SBRD robusto presentan distintos antecedentes, mientras que las que compiten entre sí son las que tienen el mismo antecedente. Bajo esta suposición, propone un SBRD llamado ELF (que introduciremos en la sección 2.4.2.2), basado en una **modificación del enfoque Michigan**, que toma lo mejor de los dos enfoques anteriores. ELF divide la población de reglas empleada habitualmente en el enfoque Michigan en varias subpoblaciones formadas por reglas difusas que se emparejan con el mismo estado. El proceso hace que las reglas existentes en una subpoblación compitan para proponer la mejor acción, y fomenta la cooperación entre las distintas subpoblaciones para generar finalmente el mejor SBRD posible. Esta forma de trabajo constituye una solución intermedia que resuelve correctamente el PCC y reduce de un modo significativo la dimensión del espacio de búsqueda. Los dos objetivos son satisfechos gracias a la formación de nichos y especies en la población.

Por último, los SBRDEs basados en el **enfoque AIR** tratan al mismo tiempo de resolver correctamente el PCC y de reducir el espacio de búsqueda. Para ello, estos procesos siguen la forma de trabajo habitual de particionar los problemas y dividen el proceso evolutivo de aprendizaje en, al menos, dos subprocesos: el de generación de reglas

difusas y el de postprocesamiento, por lo que reciben el nombre de *SBRDEs en varias etapas* [GH97]. Así, estos sistemas *resuelven el PCC en dos etapas empleando dos procesos que actúan a dos niveles distintos: en el primero, predomina la competición entre reglas difusas y en el segundo la cooperación entre las reglas generadas en el anterior*. Con ello, el *proceso evolutivo de generación de reglas* fuerza la competición entre reglas, como los basados en el enfoque Michigan, *para obtener una base de reglas difusas compuesta por las mejores reglas posibles para cada estado*, pero no la cooperación entre ellas, que sólo es débilmente considerada por el criterio de penalización de reglas. En cambio, el *proceso evolutivo de postprocesamiento* fuerza la cooperación entre las reglas difusas generadas en el proceso anterior, refinando o eliminando las reglas redundantes o innecesarias para generar la *mejor base de reglas difusas posible*.

Como puede observarse, los SBRDEs en varios niveles y ELF trabajan de forma similar. El modo de operación iterativo seguido por el proceso evolutivo de generación de reglas, comentado en la sección 2.2.3, provoca la formación de nichos y reduce considerablemente el tamaño del espacio de búsqueda. El método de postprocesamiento también trabaja con un espacio de búsqueda más reducido, ya que únicamente maneja las reglas difusas generadas en el proceso anterior. La principal diferencia entre las dos filosofías de trabajo es que los SBRDEs en varios niveles descomponen la solución del PCC en dos fases, en las que predominan la competición y la cooperación, respectivamente, mientras que ELF resuelve el problema en un único paso.

2.3.4 Aspectos Clave en el Diseño de Sistemas Basados en Reglas Difusas Evolutivos

Independientemente del tipo de SBRDE con el que se esté trabajando y de la aplicación a la que se pretenda destinar el SBRD diseñado finalmente, hay una serie de cuestiones de gran importancia referentes al diseño del AE empleado, que han de ser consideradas a la hora de construir un SBRDE. Muchas de ellas son generales en el sentido de que han de ser consideradas cuando se afronta la resolución de cualquier problema empleando AEs, aunque otras son específicas para el problema de la definición de la base de reglas difusas de un SBRD. En [SK94], se recogen todas ellas:

- Definición del tamaño de la población que manejará el AE.
- Elección del esquema de representación que se empleará para codificar las posibles soluciones, es decir, los parámetros de definición de la base de reglas difusas considerados en el proceso evolutivo de aprendizaje.
- Definición de la forma en la que se provocará la aparición de nichos en el espacio de búsqueda, en el caso de trabajar con un AE que presente esta propiedad.
- Elección o diseño de la estrategia de selección de descendientes y de los operadores evolutivos a aplicar, así como de los valores de los parámetros asociados a los mismos.

- Especificación del modo en el que se empleará el conocimiento disponible sobre el problema para mejorar el proceso evolutivo.
- Definición del criterio de parada del AE.
- Generación de la población inicial de individuos del AE.
- Diseño de una función de adaptación que mida el grado de adecuación de la regla difusa o base de reglas difusas codificada en el individuo.

A continuación, comentaremos brevemente los aspectos más relevantes de entre los anteriores [SK94, HM97, MV97]:

1. *Elección del esquema de representación:* Dentro de esta tarea de diseño, se han de resolver dos subtareas: la elección del esquema de codificación a emplear entre la amplia gama de posibilidades existente: codificación binaria, *k*-aria, real, basada en orden, expresiones en Lisp, árboles, etc. ; y el posicionamiento de la información en la cadena que representa el individuo, es decir, el orden en el cual se codifican los parámetros de la base de reglas difusas que se pretenden optimizar.

En lo que respecta a la primera decisión, el esquema más empleado clásicamente al trabajar con AGs ha sido el binario, pero en los últimos años las ventajas de la codificación real se están imponiendo paulatinamente. Esta primera subtarea es fundamental, ya que la longitud de la cadena que representa al individuo, la cual depende del esquema de codificación empleado y del número de parámetros de la base de reglas difusas que se desean aprender, determina directamente el tamaño y la complejidad del espacio de búsqueda.

2. *Diseño de los operadores evolutivos:* Esta tarea consiste en la definición de los operadores evolutivos que permiten generar nuevas soluciones potenciales. El éxito de un proceso de búsqueda necesita de la existencia de un equilibrio adecuado entre la exploración y la explotación del espacio de posibles soluciones, y dicho equilibrio depende directamente de los operadores evolutivos diseñados.

Puesto que los operadores evolutivos trabajan sobre una representación codificada de una definición de una base de reglas difusas, es necesario que exista una cierta compatibilidad entre ellos y la estructura de los individuos. Esta compatibilidad puede ser llevada a la práctica de dos formas distintas: bien, trabajando con un esquema de codificación clásico (por ejemplo, el binario en el caso de un AG) y utilizando el conjunto de operadores habitualmente empleados para el mismo; o bien, haciendo uso de una representación no clásica para codificar las posibles soluciones, y diseñando operadores evolutivos específicos que permitan trabajar adecuadamente con los nuevos genomas. De este modo, la definición del conjunto de operadores evolutivos debe ser afrontada a la vez que se decide el esquema de representación a emplear.

3. *Diseño de la función de adaptación*: La función de adaptación es la encargada de guiar la búsqueda a través del espacio, por lo que debe ser capaz de distinguir entre buenas y malas soluciones, asociando mejores valores a las primeras. En el área de los SBRDEs, la composición de la función de adaptación depende, en líneas generales, de dos factores: el esquema evolutivo de aprendizaje seguido por el AE y la aplicación a la que se destina el SBRD que se está diseñando.

El primero de ellos determina si los individuos de la población del AE codifican reglas difusas individuales o bases de reglas difusas completas. De este modo, la función de adaptación debe tener en cuenta esta característica para asociar adecuadamente las recompensas a los individuos. Por otro lado, el hecho de que el sistema final vaya a ser empleado en una aplicación de modelado o de control determina también la forma de esta función. En el primero de los casos, el índice de rendimiento empleado para definirla suele estar basado en medidas de error que calculan la diferencia existente entre la salida deseada y la que devuelve actualmente el SBRD codificado en el individuo. En el segundo, se suele emplear el propio sistema a controlar, en el caso del aprendizaje en línea, o un modelo del mismo, cuando se realiza un aprendizaje fuera de línea; y se definen índices de rendimiento que determinan la bondad de la actuación del SBRD evaluado en el control de dicho sistema o modelo.

4. *Especificación del modo en que se empleará el conocimiento disponible*: Como destacábamos en el primer capítulo, una de las ventajas principales de los SBRDs es que son los únicos sistemas capaces de trabajar con dos tipos de información: numérica y lingüística, y de hacer uso de ambas en el proceso de diseño del propio sistema. Los AEs presentan además la ventaja de su flexibilidad, que les permite incorporar el conocimiento disponible sobre el problema y mejorar así el proceso de búsqueda. De este modo, el empleo de un AE para diseñar un SBRD constituye un marco muy adecuado para aprovechar toda la información disponible en el proceso de diseño.

Como ya hemos comentado anteriormente, el conocimiento a priori puede estar presente en formas muy distintas cuando se afronta la generación de la base de reglas difusas de un SBRD [SK94]: conocimiento sobre las variables lingüísticas existentes en el problema, sobre la forma de las funciones de pertenencia, sobre los intervalos de definición de éstas, existencia de reglas difusas totalmente definidas, conocimiento sobre el número de reglas que pueden componer la base, sobre la existencia de reglas simétricas, existencia de datos de entrada-salida que definan el comportamiento del problema, etc. Este conocimiento a priori puede ser incorporado al AE de diversas formas, tales como las siguientes:

- Manteniendo las reglas difusas existentes en la base de reglas difusas y empleándolas para calcular la acción y la medida de rendimiento. Usar el AE para generar el resto de las reglas de la base.
- Haciendo uso de la simetría existente en el problema para reducir el espacio de búsqueda a la mitad, persiguiendo obtener únicamente la definición de la mitad de los parámetros asociados a la base.

- Empleando el conocimiento disponible para generar la población inicial de soluciones que manejará el AE. El trabajar de esta forma permite que el AE converja mucho más rápidamente hacia buenas soluciones ya que, en el caso en el que esta información no esté disponible, la población inicial se genera aleatoriamente.
5. *Definición del esquema de formación de nichos*: La capacidad que presentan algunos AEs para permitir la formación de nichos en el espacio de búsqueda puede aportar numerosas ventajas en el diseño de la base de reglas difusas de un SBRD. El esquema de nichos puede ser empleado tanto para mejorar el proceso de obtención de nuevas soluciones por parte del AE [Bon96], como para permitir la generación de distintas definiciones finales de la base de reglas difusas [KS95]. Consideraremos ambas posibilidades en los procesos evolutivos de aprendizaje presentados en los capítulos 4 y 5 de esta memoria.

Dentro de las dos posibilidades de esquemas de formación de nichos existentes (véase la sección 2.1.1.4), el esquema fenotípico es el más empleado ya que trabaja directamente con las reglas difusas en lugar de con su genoma, lo que permite afrontar la definición de la base de reglas difusas de un modo más adecuado. En los capítulos 4 y 5, emplearemos el esquema genotípico con el objeto de permitir la generación de distintas definiciones de la base de reglas difusas.

2.4 Algunos Ejemplos Concretos de Sistemas Basados en Reglas Difusas Evolutivos

Finalizaremos este capítulo presentando brevemente algunas aproximaciones concretas de SBRDEs existentes en la literatura especializada. Las tres secciones siguientes están dedicadas, respectivamente, a cada una de las familias de SBRDEs introducidas anteriormente.

2.4.1 Sistemas Basados en Reglas Difusas Evolutivos para la Definición Automática de la Base de Datos

Dentro de la primera familia, introduciremos un SBRDE para el diseño de la BD de un SBRD descriptivo de tipo Mamdani, el propuesto por Karr en [Kar91b], y otro para el ajuste aproximativo de las funciones de pertenencia de una base de reglas difusas descriptiva o aproximativa, presentado por Herrera, Lozano y Verdegay en [HLV95].

2.4.1.1 El método evolutivo de definición de la Base de Datos de Karr

La aproximación de Karr [Kar91b] está basada en la existencia de particiones difusas primarias de los distintos espacios en los que se definen las variables de entrada y de salida existentes en el sistema. Emplea un AG para definir la semántica de los distintos conjuntos de términos lingüísticos, es decir, para aprender la forma de los conjuntos difusos asociados a cada una de las etiquetas lingüísticas definidas en las particiones difusas. Es, por tanto, un proceso de diseño para SBRDs descriptivos de tipo Mamdani, aunque podría ser adaptado para permitir la definición de la semántica de las variables de entrada en un SBRD TSK con matices descriptivos en el antecedente.

Con objeto de llevar a cabo la tarea comentada, Karr emplea un AG simple con codificación binaria, esquema de selección proporcional, cruce simple y mutación aleatoria. El esquema de codificación binario permite representar todas las funciones de pertenencia asociadas a las distintas etiquetas lingüísticas de los conjuntos de términos en un único cromosoma.

Karr considera funciones de pertenencia triangulares para definir la semántica de las etiquetas de la BD. Para la definición de cada triángulo, tiene en cuenta únicamente los dos puntos extremos del soporte de cada conjunto difuso, dejando fijo el punto central con valor de pertenencia 1 (la moda). De este modo, la evolución que lleva a cabo el AG permite alterar la forma de los triángulos (cuando se modifica la anchura de la base) o trasladar éstos (cuando ambos puntos son desplazados a través del eje X) libremente. El proceso varía en los triángulos situados en los extremos del intervalo, los cuales requieren únicamente el uso de un punto (con valor de pertenencia 0) para su definición. Su punto modal coincide con el valor del extremo correspondiente del universo de discurso de la variable. En este caso, solamente es posible llevar a cabo una operación sobre estos triángulos consistente en alterar la anchura de su base en una de las dos direcciones del eje X, es decir, agrandarlos o estrecharlos. En la figura 2.6 se muestra un ejemplo del tipo de partición difusa considerada.

La restricción impuesta sobre las funciones de pertenencia obliga a que las funciones asociadas a las etiquetas de los extremos mantengan triángulos rectos, mientras que las asociadas a los términos interiores emplean triángulos isósceles. Es claro el hecho de que esta restricción impide que el AG obtenga funciones de pertenencia mal definidas.

Las codificaciones parciales de las funciones de pertenencia existentes en las particiones difusas asociadas a cada variable del sistema se unen para formar un cromosoma que representa la definición completa de la BD en una única cadena binaria. Puesto que cada punto se representa mediante un número fijo de bits, los cromosomas son de longitud fija, es decir, todos los individuos de la población tienen la misma longitud.

Finalmente, Karr no define explícitamente una función de adaptación, pero introduce algunos aspectos a considerar para construirla. Su idea para la medida de la adecuación de un individuo concreto se basa en medir el rendimiento del SBRD que maneja una BC formada por la BRL original y la BD codificada en el mismo en el problema concreto a

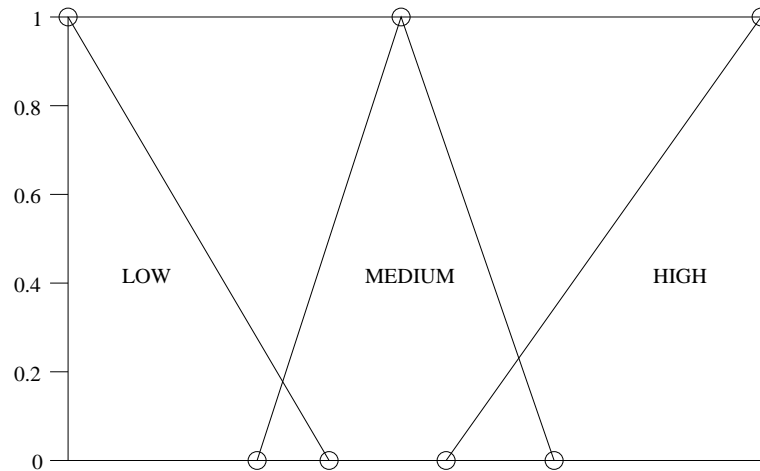


Figura 2.6. Un ejemplo del tipo de partición difusa empleada en el proceso evolutivo de definición de la BD de Karr

resolver.

2.4.1.2 El método evolutivo de ajuste de funciones de pertenencia de Herrera, Lozano y Verdegay

En [HLV95], se presenta un proceso evolutivo para el ajuste de las funciones de pertenencia manejadas en las reglas de una base de reglas difusas de tipo Mamdani previamente obtenida. El proceso es aplicable independientemente de la naturaleza descriptiva o aproximativa de dicha base, y devuelve como resultado un SBRD aproximativo de tipo Mamdani. Cada cromosoma codifica una definición completa de la base de reglas difusas ya que almacena individualmente las T reglas difusas que la componen, incluyendo los conjuntos difusos que cada una de ellas emplea. El SBRDE sigue, por tanto, un enfoque Pittsburgh.

El AG diseñado para el proceso está basado en un esquema de codificación con parámetros reales y usa el muestreo universal estocástico como procedimiento de selección y la mutación no uniforme de Michalewicz. Con respecto al cruce, se emplea el operador max-min-aritmético, propuesto por los autores. La descripción de los distintos operadores puede consultarse en la sección 2.1.1.

El método emplea funciones de pertenencia trapezoidales a las que asocia una representación paramétrica basada en una tupla formada por cuatro valores reales. Supongamos que la siguiente regla es la i -ésima de la base de reglas difusas:

$$\text{Si } X_1 \text{ es } A_{i1} \text{ y } X_2 \text{ es } A_{i2} \text{ y } \dots \text{ y } X_n \text{ es } A_{in} \text{ entonces } Y \text{ es } B_i$$

En ese caso, las funciones de pertenencia asociadas a los A_{ij} y a B_i (sean éstos etiquetas lingüísticas con una semántica asociada o bien, directamente, conjuntos difusos) se

representan mediante las tuplas $(c_{ij}, a_{ij}, b_{ij}, d_{ij})$ y (c'_i, a'_i, b'_i, d'_i) , respectivamente, según la equivalencia reflejada en la figura 2.7.

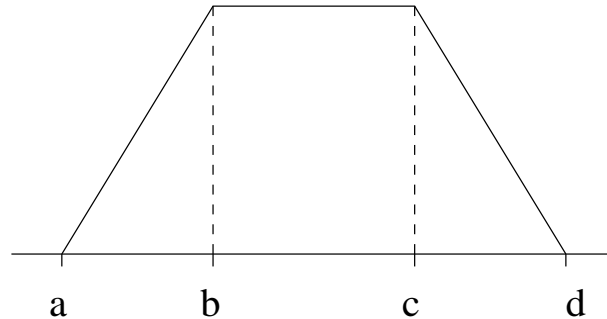


Figura 2.7. Representación de un conjunto difuso trapezoidal en forma de tupla con cuatro valores

Así, cada regla difusa se codifica en una porción de cromosoma C_{ri} , $i = 1, \dots, T$, del siguiente modo:

$$C_{ri} = (c_{i1}, a_{i1}, b_{i1}, d_{i1}, \dots, c_{in}, a_{in}, b_{in}, d_{in}, c'_i, a'_i, b'_i, d'_i)$$

Así, la definición completa de la base de reglas difusas se codifica en un cromosoma C_r de la forma:

$$C_r = C_{r1} C_{r2} \dots C_{rT}$$

Tal y como puede verse, cada individuo codifica el conjunto de reglas difusas original y la diferencia entre ellos radica en las funciones de pertenencia empleadas en cada regla difusa. Puesto que cada una de ellas se codifica en una porción de cromosoma, si partimos de una base de reglas descriptiva, el proceso evolutivo puede cambiar un conjunto difuso en una regla y no en otras en las que aparezca o, en el caso más extremo, la forma de dicho conjunto difuso puede ser distinta en cada una de las reglas en las que aparece. La semántica de los antiguos términos lingüísticos pasa a depender de este modo de la regla concreta en la que aparezcan dichos términos, con lo que la base de reglas difusas finalmente obtenida presentará un comportamiento aproximativo.

La generación de la primera población se hace a partir de la definición de la base de reglas difusas inicial. Esta primera definición es codificada directamente en un cromosoma notado por C_1 , mientras que los individuos restantes son generados asociando un intervalo de variación, $[c_h^i, c_h^d]$, a cada gen c_h de C_1 , $h = 1 \dots (n+1) \cdot T \cdot 4$. Cada intervalo de variación constituye el intervalo de ajuste de la variable correspondiente, $c_h \in [c_h^i, c_h^d]$.

Si $(t \bmod 4) = 1$ entonces c_t es el valor situado más a la izquierda del soporte del número difuso. Dicho número se define mediante cuatro parámetros $(c_t, c_{t+1}, c_{t+2}, c_{t+3})$ y los intervalos de variación son los siguientes:

$$\begin{aligned}
c_t &\in [c_t^i, c_t^d] = \left[c_t - \frac{c_{t+1} - c_t}{2}, c_t + \frac{c_{t+1} - c_t}{2} \right] \\
c_{t+1} &\in [c_{t+1}^i, c_{t+1}^d] = \left[c_{t+1} - \frac{c_{t+2} - c_{t+1}}{2}, c_{t+1} + \frac{c_{t+2} - c_{t+1}}{2} \right] \\
c_{t+2} &\in [c_{t+2}^i, c_{t+2}^d] = \left[c_{t+2} - \frac{c_{t+3} - c_{t+2}}{2}, c_{t+2} + \frac{c_{t+3} - c_{t+2}}{2} \right] \\
c_{t+3} &\in [c_{t+3}^i, c_{t+3}^d] = \left[c_{t+3} - \frac{c_{t+4} - c_{t+3}}{2}, c_{t+3} + \frac{c_{t+4} - c_{t+3}}{2} \right]
\end{aligned}$$

La figura 2.8 muestra la representación gráfica de estos intervalos. De este modo, se crea una población de cromosomas con C_1 como primer individuo y los restantes inicializados aleatoriamente, variando cada gen en su intervalo de ajuste respectivo.

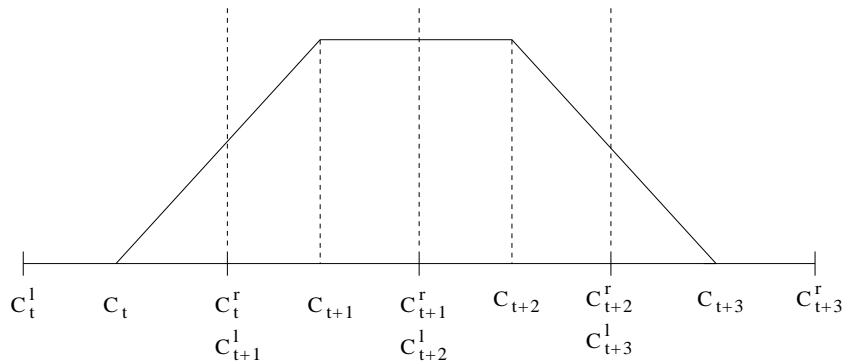


Figura 2.8. Intervalos de variación

La función que mide el grado de adaptación de un cromosoma se define haciendo uso de un conjunto de entrenamiento, E_{TDS} , y una medida de error concreta, el error cuadrático medio. De este modo, el valor de adecuación que se asocia a cada individuo se obtiene calculando el error cometido por el SBRD aproximativo que emplea la BRD codificada en el cromosoma, al tomar como entrada los distintos valores almacenados en el conjunto de entrenamiento. La función de adaptación se representa mediante la expresión siguiente:

$$E(C) = \frac{1}{2|E_{TDS}|} \sum_{e_k \in E_{TDS}} (ey^k - S(ex^k))^2$$

2.4.2 Sistemas Basados en Reglas Difusas Evolutivos para el Aprendizaje de la Base de Reglas Lingüísticas

En lo que respecta a esta segunda familia, analizaremos el método de Thrift [Thr91], que emplea la codificación en forma de tabla de decisión para generar la BRL de un SBRD descriptivo de tipo Mamdani; y el de Bonarini [Bon93], que sigue una variante del enfoque Michigan para llevar a cabo la misma tarea.

2.4.2.1 El método evolutivo de generación de la Base de Reglas Lingüísticas de Thrift

Este método, como muchos otros de este grupo, está basado en codificar todas las celdas de la tabla de decisión en los cromosomas. Así, Thrift [Thr91] establece una aplicación entre el conjunto de etiquetas asociado a la variable de salida del sistema y un conjunto creciente de enteros (que contiene un elemento más que el anterior y tiene al 0 como primer elemento) que representa el conjunto de valores posibles para los genes. Vamos a ver un ejemplo para clarificar el concepto. Sea $\{NG, NP, CE, PP, PG\}$ el conjunto de términos asociado a la variable de salida, podemos representar la ausencia de valor para la misma empleando el símbolo $-$. Se establece una aplicación entre el conjunto completo constituido añadiendo este símbolo al conjunto de términos y el conjunto $\{0, 1, 2, 3, 4, 5\}$. Con ello la etiqueta NG se asocia con el valor 0, NP con el 1, \dots , PG con el 4 y el símbolo nulo $-$ con el 5.

De este modo, el AG empleado usa un esquema de codificación entera. Cada cromosoma se construye uniendo las codificaciones parciales asociadas a cada etiqueta lingüística contenida en las celdas de la tabla de decisión. Un gen que toma el alelo $-$ representa la ausencia de la regla difusa contenida en la celda correspondiente de la BRL.

El AG propuesto utiliza un esquema de selección elitista y los operadores genéticos empleados son de distinta naturaleza. Mientras que el operador de cruce es el cruce multipunto clásico (efectuado en dos puntos), el operador de mutación ha sido diseñado específicamente para este proceso. Cuando se aplica sobre un alelo distinto al símbolo nulo, se añade o resta uno o se cambia al valor nulo. Cuando el valor anterior del gen es el símbolo nulo, se selecciona un nuevo valor aleatoriamente.

Por último, la función de adaptación está también basada en una medida específica de la aplicación. La adaptación de un individuo se determina aplicando el SBRD de tipo Mamdani que emplea una BC formada por la BD definida previamente y BRL codificada en su genotipo al sistema a controlar, partiendo de distintos estados y calculando la convergencia del proceso hacia el punto de equilibrio deseado.

2.4.2.2 El método evolutivo de generación de la Base de Reglas Lingüísticas de Bonarini

El método que propone Bonarini [Bon93], denominado ELF (Evolutionary Learning of Fuzzy Rules, es decir, aprendizaje evolutivo de reglas difusas), es bastante distinto a las otras aproximaciones englobadas en este grupo. El autor considera el alto costo computacional necesario para obtener una BRL mediante un AG con una población de individuos que codifiquen la tabla de decisión completa, es decir, asociado al enfoque Pittsburgh, y decide diseñar un SBRDE basado en el enfoque Michigan.

Con objeto de llevar a cabo esta tarea, el AG empleado por Bonarini trabaja con una población de reglas. Esta nueva codificación permite aprender la estructura completa

de las reglas difusas de tipo Mamdani, es decir, sus antecedentes, sus consecuentes y el número óptimo de reglas que formarán parte de la BRL. El autor desea aprender con ELF los estados que se darán en el sistema a controlar y determinar los que son irrelevantes para el mismo, obteniendo así BRLs óptimas para la aplicación.

La función de adaptación juzga el estado alcanzado cada vez que se disparan las reglas. Cada individuo de la población, es decir, cada regla, posee información asociada referente a varias cuestiones: con qué nivel ha sido juzgada (su fuerza), cuándo fue generada, cuándo fue disparada por última vez y en cuánto contribuyó a las acciones llevadas a cabo anteriormente por el SBRD, que en este caso es empleado en una aplicación de control. ELF modifica la fuerza asociada a la regla difusa de acuerdo con el rendimiento de la acción a la que contribuye. Este rendimiento es determinado empleando la función de adaptación.

Otra característica importante de este método, asociada al empleo del enfoque Michigan, es que está diseñado para ser ejecutado en un entorno real. En primer lugar, el proceso selecciona aquellas reglas cuyo antecedente presenta un grado de emparejamiento superior a un umbral proporcionado por el diseñador del SBRD, de entre las que pueden ser disparadas en el estado actual del sistema. Las reglas incluidas en este conjunto compiten entre ellas para proporcionar la mejor acción de control para el estado actual.

Sobre este conjunto de individuos se aplican varios operadores genéticos. Algunas de estas reglas se consideran lo suficientemente evaluadas, puesto que han contribuido en más de un cierto grado a las acciones pasadas. Si varias de ellas poseen un grado de fuerza bajo, eso quiere decir que las acciones de control que proporcionaron no funcionaban correctamente, por lo que son sustituidas por otras que funcionen mejor. Los consecuentes de las reglas del conjunto son modificados con una probabilidad inversa a su fuerza. Este último paso conforma el operador de mutación del AG y permite que ELF localice nuevas reglas en un entorno próximo a otras que ha aprendido anteriormente. Como comentábamos en la sección 2.3.3, esto constituye una diferencia con respecto al enfoque Michigan clásico y provoca la creación de nichos y especies en el espacio de búsqueda del problema.

Si ha transcurrido un amplio intervalo de tiempo desde la última modificación efectuada sobre las reglas, esto quiere decir que la población de reglas que se comportan adecuadamente ante el estado actual del sistema se ha estacionado, y todas ellas poseen más o menos la misma fuerza. En este caso, ELF selecciona la peor regla y muta sus consecuentes para continuar la búsqueda, tratando de localizar una configuración mejor. Si existen pocas reglas que puedan ser disparadas en el estado actual (el parámetro que indica cuándo el número de reglas es suficiente cambia su valor dinámicamente), ELF genera una nueva regla que lo cubra y propone aleatoriamente una acción de control. Este es el único mecanismo que introduce nuevos antecedentes y se denomina *detector de cubrimiento*. Este proceso puede introducir, con una probabilidad dada, símbolos de tipo "no interviene" como valor para alguna de las variables de entrada de la nueva regla generada. Las reglas que presentan este tipo de símbolos en el antecedente se emparejan con distintos estados y compiten con distintos grupos de reglas, uno por cada uno de los estados con

los que se emparejen. Son individuos, por tanto, encuadrados en varias especies y nichos.

La situación dual se da cuando existen demasiadas reglas que casan con un estado concreto. En este caso, el operador genético que se aplica se denomina *eliminador de reglas* y simplemente elimina de la población la peor de las reglas cuyo antecedente se empareja con dicho estado.

Una vez que se ha llevado a cabo la selección y se han aplicado los operadores genéticos, ELF emplea el SBRD con la BRL codificada en la población en el estado actual. La función de adaptación evalúa entonces el nuevo estado obtenido, proporcionando la debida recompensa. Existe un proceso de compartición en la distribución de esta recompensa: cada individuo obtiene una parte de la misma de acuerdo con la contribución de la regla que codifica la acción de control aplicada. Mediante este proceso se modifica la fuerza asociada a las reglas codificadas en la población.

Por último, varios aspectos merecen ser destacados. Por un lado, Bonarini propone medidas específicas de la aplicación para definir la función de adaptación. Por otro, ELF devuelve varias BRLs distintas como salida del proceso de aprendizaje. Este hecho se debe a que cada vez que el rendimiento del sistema es superior a un valor "satisfactorio" definido por el diseñador, se almacena la BRL actual y se efectúan modificaciones sobre la población actual con objeto de obtener una solución mejor.

2.4.3 Sistemas Basados en Reglas Difusas Evolutivos para el Aprendizaje de la Base de Conocimiento

Para finalizar, presentaremos tres SBRDEs distintos pertenecientes a la última de las familias existentes. En primer lugar, describiremos el funcionamiento del método propuesto por Lee y Takagi en [LT93a, LT93b], que permite obtener una base de reglas difusas completa para un SBRD de tipo TSK. Posteriormente, nos centraremos en dos SBRDEs presentados por Magdalena y Velasco en [MV96, VM95a, VM95b], los cuales llevan a cabo las tareas de diseño necesarias para obtener la BC y la BRD de un SBRD de tipo Mamdani con naturaleza descriptiva y aproximativa, siguiendo respectivamente el enfoque Pittsburgh y el Michigan.

2.4.3.1 El método evolutivo de Lee y Takagi para el aprendizaje de la Base de Conocimiento de un Sistema Basado en Reglas Difusas de tipo TSK

En [LT93a, LT93b], Lee y Takagi introducen un método evolutivo de diseño que permite aprender automáticamente una BC completa para un SBRD de tipo TSK. De este modo, la definición de los consecuentes de las reglas consiste en el aprendizaje de los parámetros p_i que se emplean para combinar linealmente los valores de entrada para obtener las salidas.

Las funciones de pertenencia consideradas en los antecedentes de las reglas son triangulares aunque los autores indican que el método puede trabajar con cualquier tipo de función de pertenencia parametrizada, tales como las gaussianas, acampanadas, trapezoidales o sigmoidales.

Se supone una partición difusa del espacio de entrada. Cada función de pertenencia triangular de los conjuntos difusos primarios asociados a los distintos términos lingüísticos se representa mediante tres parámetros. El primero es el punto central, es decir, el punto modal. Únicamente el centro del triángulo asociado al primer conjunto difuso primario viene especificado por una posición absoluta, mientras que los parámetros asociados a los demás representan la distancia existente entre el punto central del triángulo actual y el del anterior. Los dos parámetros restantes representan respectivamente los puntos derecho e izquierdo de la base del triángulo. Ambos presentan grado de pertenencia 0.

El AG empleado está basado en una codificación binaria. Las funciones de pertenencia se codifican agregando los valores binarios de los tres parámetros asociados en una subcadena binaria. Para codificar cada parámetro se usan ocho bits. La BD completa es codificada agregando las codificaciones parciales de las funciones de pertenencia asociadas a cada uno de los conjuntos difusos primarios de las variables de entrada una detrás de otra. La última parte del cromosoma se construye codificando los parámetros p_i asociados a cada combinación de los valores de entrada y uniéndolos en una nueva subcadena binaria. Estos valores son también codificados empleando ocho bits. De este modo, cada cromosoma representa una BC TSK completa. El número de reglas que forman parte de la misma dependerá del número de conjuntos difusos primarios asociados a cada una de las variables de entrada y será igual al producto de éstos. La codificación empleada permite decidir el número óptimo de reglas difusas que formarán la base del siguiente modo: aquellos términos lingüísticos para los cuales el punto central de la función de pertenencia asociada cae fuera de un límite concreto, obtenido a partir del conocimiento que se posee sobre el sistema, provocan que la regla en cuestión no forme parte de la base. Así, todas las reglas cuyo antecedente sea una combinación de los valores de entrada válidos codificados en la primera parte del cromosoma formarán parte de la base de reglas difusas del sistema. Tomemos como ejemplo un cromosoma que codifique la BC de un sistema con dos variables de entrada que posean asociadas respectivamente m y n etiquetas lingüísticas válidas. El número total de reglas que puede poseer la base de reglas difusas que codifica es $m \cdot n$.

Para el caso del problema de control propuesto, el problema del Péndulo Invertido, la función de adaptación se basa en optimizar dos criterios distintos. Por un lado, una medida específica de la aplicación (concretamente, en el ejemplo propuesto se emplea una medida de la convergencia del sistema y el SBRD TSK se ejecuta desde varios estados iniciales distintos). Por otro, los cromosomas que codifican BCs TSK con un gran número de reglas son penalizados para obtener otros con menos reglas.

Por último, es muy importante destacar el hecho de que los autores proponen dos modos distintos de incorporar el conocimiento previo que se posee sobre el sistema al SBRDE presentado, con objeto de mejorar su comportamiento. Por un lado, es posible

incorporar conocimiento mediante las configuraciones iniciales de los parámetros de la BC TSK. Este conocimiento se emplea para generar la población inicial del AG. De este modo, los individuos que la constituyen no son generados aleatoriamente sino que algunos de ellos son obtenidos mediante una equipartición de los espacios de entrada empleando un número distinto de términos lingüísticos. Este conocimiento puede ser empleado también para fijar inicialmente valores apropiados para los parámetros p_i de los consecuentes de las reglas.

Por otro lado, el conocimiento previo puede ser incorporado mediante una representación estructural de las BCs. Por ejemplo, en un problema que presente una naturaleza simétrica como el del control del Péndulo Invertido, el SBRD puede ser restringido con objeto de particionar de una forma simétrica el espacio de entrada alrededor del origen. Esto reducirá el tamaño del espacio de búsqueda puesto que el número de funciones de pertenencia se reduce a la mitad.

2.4.3.2 El método evolutivo de Magdalena y Velasco para el aprendizaje de la Base de Conocimiento de un Sistema Basado en Reglas Difusas descriptivo de tipo Mamdani

En [MV96, VM95a, VM95b], Magdalena y Velasco presentan un SBRDE basado en el enfoque Pittsburgh para generar la definición completa de la BC de un SBRD descriptivo de tipo Mamdani con múltiples entradas y salidas.

El sistema está basado en un AG en el que cada cromosoma codifica una definición distinta de la BD y la BRL del SBRD. El método permite el manejo de hasta tres tipos de información diferentes en la definición de la BD: el conjunto de parámetros asociado, la forma de las funciones de pertenencia y los intervalos de normalización:

1. El conjunto de parámetros define el número de variables de entrada (N) y de salida (M) del sistema, así como el número de términos lingüísticos asociado a cada una de ellas. Se codifican en una subcadena formada por dos vectores de números reales, \vec{m} y \vec{n} , en la que el valor contenido en cada posición representa el número de términos asociado a la variable correspondiente de entrada, en el caso del vector \vec{m} , o de salida, en el caso de \vec{n} .
2. Las funciones de pertenencia consideradas son trapezoidales y están definidas en el intervalo $[-1, 1]$, por lo que la codificación de las mismas se realiza mediante la representación en forma de tupla con cuatro valores definidos en dicho intervalo. De este modo, la forma de las funciones de pertenencia se codifica en un vector real de dimensión $4 \cdot L$, donde $L = L_a + L_c$ es el número total de etiquetas existentes en la BD y L_a , L_c son el número de etiquetas asociadas a las variables de entrada y de salida, respectivamente.
3. Finalmente, los intervalos de normalización o factores de escala son los encargados de transformar el intervalo genérico de definición $[-1, 1]$ al intervalo $[v_{min}, v_{max}]$

asociado a cada variable concreta. Se codifican en un vector real de dimensión $2 \cdot (N + M)$.

En lo que respecta a la definición de la BRL, incluye únicamente la especificación de las reglas difusas que forman parte de la base. Estas reglas presentan una estructura un poco más general que la introducida en el capítulo 1, ya que permiten que una variable tome como valor distintas etiquetas lingüísticas conectadas mediante el operador o . Para representar este nuevo tipo de reglas difusas, se emplea una codificación basada en dos subcadenas binarias: una de longitud L_a para especificar el antecedente de la regla y otra de longitud L_c para el consecuente. La presencia del valor 1 en una posición concreta de cualquiera de las dos subcadenas indica que la variable en cuestión toma el término codificado en esa posición como valor. Finalmente, la definición completa de la BRL se codifica mediante un cadena binaria de longitud variable formada por la concatenación de un número máximo L_r de subcadenas del tipo de las anteriores. Gracias a esta representación, el SBRDE presenta la capacidad de aprender el número óptimo de reglas en la BRL.

A la vista del esquema de representación introducido hasta el momento, observamos que, finalmente, el cromosoma está formado por dos partes basadas cada una de ellas en un esquema de codificación distinto. La parte asociada a la definición de la BD emplea una codificación real, mientras que la asociada a la BRL maneja una binaria. Los autores destacan el hecho de que la modularidad existente en el esquema de representación provoca que el SBRDE presente la versatilidad de permitir adaptar la parte que se desee dentro del código genético. Así, aunque en principio el SBRDE está encuadrado dentro de la tercera familia, puede ser empleado como un proceso evolutivo de diseño de la primera familia, en el caso en el que el genoma codifique únicamente todos o alguno de los tipos de información asociados a la definición de la BD, o de la segunda familia, cuando éste sólo incluya la definición de la BRL.

En lo que respecta a los operadores evolutivos, el SBRDE propuesto emplea los siguientes:

- *Selección*: El operador de selección es específico y trabaja a dos niveles. En primer lugar, copia directamente un grupo de miembros de la población actual a la nueva. Este grupo, denominado *élite de la población*, está formado por un conjunto de los individuos mejor adaptados de la misma, conjunto que puede estar constituido por un número fijo de individuos, un porcentaje del tamaño de la población o por todos aquellos que superen un determinado umbral en su valor de adaptación.

En segundo lugar, se aplica una versión modificada del muestreo universal estocástico sobre la población completa, en la que los individuos de mayor adaptación tienen asociada una probabilidad mayor de ser copiados una única vez en la nueva población. Esta redefinición del operador clásico permite mantener un nivel adecuado de diversidad en la población.

- *Cruce*: El cruce se efectúa independientemente en las dos partes que componen el cromosoma. En primer lugar, se aplica un cruce en un punto sobre la parte que define la BRL. Puesto que ésta presenta longitud variable, el punto de corte se genera aleatoriamente en ambos padres. Posteriormente, el proceso de cruce de la parte asociada a la BD considera qué reglas han cambiado de cromosoma al efectuar el cruce sobre la BRL y trabaja en consecuencia para obtener dos descendientes con la mejor forma posible.
- *Mutación*: Al igual que el operador de cruce, el de mutación se aplica de un modo independiente sobre las dos partes del cromosoma. Sobre la parte de reglas se aplica el operador de mutación clásico, el cual cambia el valor del bit mutado al otro valor posible en el alfabeto binario. En cambio, sobre la parte de la BD se aplican operadores de mutación adecuados para la codificación real empleada.
- *Reordenación de reglas*: Aunque la ordenación de las reglas en el cromosoma no afecta al comportamiento del SBRD codificado, ya que el operador *también* es conmutativo y asociativo, esta ordenación puede sesgar el comportamiento del operador de cruce. Debido a esta razón, los autores proponen un operador de reordenación de reglas que se aplica a cada descendiente del cruce con una probabilidad definida a la entrada del proceso. El operador se basa en generar aleatoriamente un punto de cambio dentro de la parte de reglas del cromosoma, e intercambiar la posición de los dos trozos de cromosoma que inicialmente estaban situados a la derecha y a la izquierda de dicho punto.

Finalmente, la función de adaptación está basada en una medida del rendimiento del SBRD codificado en el cromosoma al ser aplicado al problema que está siendo resuelto.

2.4.3.3 El método evolutivo de Magdalena y Velasco para el aprendizaje de Bases de Reglas Difusas aproximativas de tipo Mamdani

El segundo de los SBRDEs introducido en [MV96, VM95a, VM95b] está basado en el enfoque Michigan, trabaja con reglas difusas aproximativas de tipo Mamdani, y presenta una estructura y un funcionamiento muy similares a los Sistemas Clasificadores clásicos. Este SBRDE está formado por los siguientes componentes:

1. Un *Módulo de Observación*, encargado de recibir y generar las entradas para el sistema.
2. Una *BRD global* compuesta por varias BRDs parciales, formadas por reglas aproximativas de tipo Mamdani que manejan funciones de pertenencia trapezoidales y presentan un peso asociado, cada una de ellas preparada para actuar en un macroestado del mundo real. Esta descomposición facilita el proceso de aprendizaje, al trabajar en distintos subespacios de búsqueda de dimensiones pequeñas en lugar

de en uno más general y de mayor dimensión. Los autores indican que la generación inicial de estas BRDs parciales puede ser llevada a cabo de un modo aleatorio, a partir de expertos o empleando algún tipo de algoritmo de aprendizaje.

3. Un *Sistema de Inferencia Difuso*, el cual realiza el proceso de razonamiento difuso empleando reglas difusas aproximativas con un peso asociado.
4. Un *Interfaz de Defuzzificación*, que aplica el Centro de Gravedad después de ponderar los consecuentes obtenidos de cada regla individual por el peso de la regla, su grado de emparejamiento y el valor de activación de la BRD parcial a la que pertenece (grado de pertenencia del estado actual del sistema al macroestado representado por la BRD parcial correspondiente).
5. Un *Módulo de Aprendizaje Adaptativo*, que altera el peso asociado a las reglas difusas existentes en la BRD de acuerdo al hecho de que éstas intervengan o no en la acción devuelta por el sistema
6. Un *AG*, encargado de crear nuevas reglas difusas.

El AG propuesto presenta la característica de no ser ejecutado en cada iteración del proceso global, sino cada cierto intervalo de tiempo, cuando una BRD ha sido utilizada una serie de veces. Está basado en un mecanismo de selección aleatorio según el peso de las reglas (a mayor peso, mayor probabilidad de selección), un operador de cruce uniforme y dos operadores de mutación distintos: uno suave, que cambia la forma del conjunto difuso, y otro fuerte, que altera el término en sí. Emplea además un operador de inserción que añade las nuevas reglas difusas generadas al *Limbo*, una zona especial en la que el funcionamiento de éstas es evaluado sin afectar a la salida proporcionada por el SBRD antes de ser insertadas en la BRD global.

La función de adaptación empleada por el AG se encarga de medir la bondad de las reglas individuales existentes en la población genética. Para ello, se basa en dos criterios, una medida del comportamiento global del sistema, que se estudia con respecto a los últimos n casos, y otra de la influencia parcial de la regla concreta en dicho comportamiento.

Así, cada cierto tiempo, el AG genera nuevas reglas difusas y las introduce en el Limbo, al mismo tiempo que algunas de las reglas existentes en el Limbo pasan a formar parte de la BRD global del sistema. Tanto el propio Limbo como las reglas difusas contenidas en él tienen asociados una serie de parámetros que determinan el destino final de cada regla. Las reglas que no demuestran un buen comportamiento durante su estancia en el Limbo, acaban siendo eliminadas. En cambio, las que sí lo acaban presentando, son insertadas en la BRD parcial correspondiente, siempre que lo permita un mecanismo de filtrado encargado de evitar un número excesivo de copias de las mismas reglas en ésta.

El Módulo de Aprendizaje Adaptativo, en cambio, sí actúa en todas las iteraciones del SBRDE global. Su función es la de incrementar o decrementar el peso asociado a las reglas difusas existentes tanto en la BRD como en el Limbo de acuerdo al comportamiento

que presenten. Este peso es incrementado cuando la regla presenta un valor positivo en la función de adaptación y decrementado, en caso contrario.

Finalmente, destacaremos que el AG permite la generación de reglas a partir de ejemplos. Esto facilita la obtención de una BRD inicial capaz de llevar a cabo el control de procesos en tiempo real partiendo de una definición aleatoria de la misma, y permite el refinamiento de ésta mediante un proceso de adaptación en línea a partir de ese momento.

Capítulo 3

Una Metodología para el Diseño Automático de Sistemas Basados en Reglas Difusas Mediante Algoritmos Evolutivos

Como ya hemos señalado anteriormente a lo largo de esta memoria, la segunda tarea que es necesario llevar a cabo para diseñar un SBRD, la obtención de la base de reglas difusas, no es fácil de efectuar cuando no se dispone de suficiente conocimiento experto. En los últimos años, se han desarrollado una gran cantidad de métodos de aprendizaje automático que tratan de resolver el problema haciendo uso de la información numérica disponible y de las técnicas más variadas. Gran parte de estos procesos presentan una serie de inconvenientes tales como el hecho de que no aprovechan la información experta disponible o que solamente permiten el diseño de un tipo concreto de SBRD.

Debido a estas razones, parece adecuado el desarrollo de una metodología más general que permita la generación automática de la base de reglas difusas en el caso en el que éstas no puedan ser obtenidas directamente a partir de la información experta existente. Esta metodología debe posibilitar la creación de procesos de aprendizaje automático que permitan diseñar SBRDs de todos los tipos existentes y que aprovechen la información experta disponible para mejorar la definición de la base de reglas difusas. Además, debe establecer una estructura genérica y una serie de recomendaciones de diseño que sean, a la vez, lo suficientemente flexibles como para que sea sencillo el construir procesos concretos de aprendizaje a partir de estas guías y lo suficientemente robustas como para que dichos procesos permitan el diseño de SBRDs con el mejor comportamiento posible.

El objetivo del tercer capítulo de esta memoria es el de presentar una metodología de este tipo basada en AEs. En él, vamos a sentar las bases de dicho modelo evolutivo de aprendizaje, que tendrá en cuenta las características principales de los dos elementos considerados en el proceso de diseño (el sistema a diseñar, el SBRD, y la técnica con-

siderada para llevar a cabo dicho diseño, el AE) para resolver del modo más adecuado el problema principal que se presenta al combinar ambos en un SBRDE, el problema de cooperación-competición (PCC), y para refinar las definiciones de las bases de reglas difusas obtenidas. Con este objeto, en el paradigma evolutivo de aprendizaje presentado se tendrán en cuenta las siguientes características de cada sistema:

- Por parte del SBRD, se tendrán en cuenta las propiedades más importantes asociadas a las bases de reglas difusas, presentadas en el capítulo 1, las cuales aseguran el buen comportamiento del sistema.
- En lo que respecta al AE, en la metodología se tomará como base el enfoque de Aprendizaje Iterativo de Reglas (AIR), estudiado en el capítulo 2, y se introducirán una serie de mejoras para resolver mejor el PCC. Entre dichas mejoras podemos destacar: el llevar a cabo las dos tareas posibles, simplificación y refinamiento de reglas, en la fase de postprocesamiento, dividiendo ésta en dos etapas; el tratar de afrontar parte de la cooperación entre reglas difusas en la primera fase del aprendizaje y el diseñar de un modo más adecuado los componentes de los AEs empleados en las distintas etapas del SBRDE.

Así, integrando los aspectos comentados con otros particulares que introduciremos a lo largo del capítulo actual, los SBRDEs que se construyan a partir de la metodología evolutiva de aprendizaje refinarán el proceso de diseño y permitirán generar buenas definiciones para la base de reglas difusas del SBRD, independientemente del tipo de éste.

La estructura de este capítulo es la que comentamos a continuación. En primer lugar analizaremos los aspectos básicos del modelo evolutivo de aprendizaje. Posteriormente, dedicaremos una sección a describir los aspectos generales asociados a cada uno de los tres procesos individuales que componen los SBRDEs construidos a partir de la metodología propuesta. Finalmente, efectuaremos un pequeño análisis de los distintos parámetros que manejan estos SBRDEs en varias etapas, estudiando la influencia de los mismos en la composición final de la base de reglas difusas aprendida.

3.1 Aspectos Básicos de la Metodología Evolutiva de Aprendizaje

En esta sección introduciremos las bases que rigen el paradigma evolutivo de aprendizaje de bases de reglas difusas propuesto, analizando en profundidad los siguientes aspectos:

- Propiedades exigidas a la base de reglas difusas aprendida.

- Mecanismos para resolver adecuadamente el PCC.
- Estructura de los SBRDEs en varias etapas construidos según la metodología presentada.
- Manejo de la información disponible para el diseño de la base de reglas difusas.

3.1.1 Propiedades Exigidas a la Base de Reglas Difusas Aprendida

Como comentábamos en la sección 1.4 del primer capítulo, la base de reglas difusas de un SBRD debe verificar una serie de propiedades para que dicho sistema presente un rendimiento adecuado. Diseñaremos nuestro proceso evolutivo de aprendizaje exigiendo que la base de reglas difusas obtenida satisfaga dos de ellas, la *completitud* y la *consistencia*. Consideraremos la verificación de estas propiedades en función del comportamiento de la base de reglas difusas sobre el conjunto de ejemplos de entrenamiento del que se dispone, E_p , el cual será empleado para realizar el proceso de aprendizaje automático y estará compuesto por p vectores numéricos de valores de entrada-salida que presentarán la estructura siguiente:

$$e_l = (ex_1^l, \dots, ex_n^l, ey^l) \quad , \quad l = 1, \dots, p$$

Analizamos a continuación la forma en que se exigirá la satisfacción de ambas propiedades:

1. *Completitud de la base de reglas difusas*: Como ya hemos comentado en esta memoria, la propiedad de completitud se refiere al hecho de que un SBRD debe ser capaz de proporcionar una salida ante cualquier entrada que reciba. Esta propiedad es independiente del valor concreto de dicha salida, simplemente considera que ésta no sea vacía. Puesto que en aprendizaje inductivo, es interesante considerar no sólo este último hecho, sino también que la salida proporcionada por el sistema ante una entrada sea lo más parecida posible a la salida real, la cual se conoce, vamos a modificar la propiedad de completitud para que considere este hecho. De este modo, consideraremos que una base de reglas difusas es completa cuando cubra todos los ejemplos del conjunto de entrenamiento.

Para incorporar dicha propiedad en los SBRDEs construidos a partir del modelo evolutivo propuesto, emplearemos una formulación matemática, basada en el uso de un parámetro real $\tau \in (0, 1]$ [HLV97b]. Sea una base de reglas difusas R formada por T reglas difusas de tipo Mamdani, R_i , con la estructura

$$R_i : \text{Si } X_1 \text{ es } A_{i1} \text{ y } \dots \text{ y } X_n \text{ es } A_{in} \text{ entonces } Y \text{ es } B_i,$$

donde X_i e Y son variables de entrada y salida respectivamente, y los A_{ij} y B_i son bien etiquetas lingüísticas o bien directamente conjuntos difusos (reglas descriptivas y aproximativas, respectivamente). Definiremos el *grado de compatibilidad* entre la regla R_i y el ejemplo $e_l \in E_p$ de la forma:

$$R_i(e_l) = *(A_i(ex^l), B_i(ey^l))$$

$$A_i(ex^l) = *(A_{i1}(ex_1^l), \dots, A_{in}(ex_n^l))$$

donde $*$ es una t-norma. Consideraremos que la base de reglas difusas R es τ -*completa* cuando se verifique que

$$C_R(e_l) = \bigcup_{i=1..T} R_i(e_l) \geq \tau \quad , \quad l = 1, \dots, p$$

Por otro lado, definiremos también el *grado de cubrimiento* $GC_R(\cdot)$ de la base de reglas difusas R sobre un ejemplo $e_l \in E_p$ de la forma [HLV97b]:

$$GC_R(e_l) = \sum_{i=1}^T R_i(e_l)$$

y exigiremos que dicho valor sea superior al de un parámetro ϵ en todos los ejemplos del conjunto de entrenamiento:

$$GC_R(e_l) \geq \epsilon, \quad l = 1, \dots, p.$$

De este modo, consideraremos que la base de reglas difusas R es adecuada cuando satisfaga las dos condiciones presentadas, es decir, cuando verifique la *propiedad de τ -completitud* y presente un buen *grado de cubrimiento*. Los valores de los parámetros τ y ϵ pueden ser especificados por el diseñador del sistema e influirán en la composición de la base de reglas difusas obtenida, como comentaremos en la sección 3.5.

2. *Consistencia de la base de reglas difusas*: Ya hemos indicado que es necesario relajar esta propiedad para trasladarla al campo de las bases de reglas difusas y que existen distintas formas para hacerlo. En nuestro caso, llevaremos a cabo esta relajación mediante los conceptos de *ejemplo positivo* y *negativo* [GP97, HLV97b]. Un ejemplo es considerado positivo para una regla difusa cuando se empareja con el antecedente y el consecuente de ésta. En cambio, será considerado negativo para la regla cuando se empareje con su antecedente y no con su consecuente. En principio, una regla difusa será inconsistente cuando presente ejemplos negativos.

Sin embargo, podría ser ilógico considerar inconsistente una regla que presentase un número bajo de ejemplos negativos y uno muy alto de ejemplos positivos. Para

evitar este hecho, en [GP97] se introduce la definición de *k-consistencia*, que relaja el concepto de consistencia haciendo uso de los cardinales de los conjuntos de ejemplos positivos y negativos de una regla difusa. Sean

$$E^+(R_i) = \{e_l \in E_p / R_i(e_l) \geq 0\} \text{ y } E^-(R_i) = \{e_l \in E_p / R_i(e_l) = 0 \text{ y } A_i(e_l) > 0\}$$

los conjuntos de ejemplos positivos y negativos de la regla R_i , respectivamente, y $n_{R_i}^+ = |E^+(R_i)|$ y $n_{R_i}^- = |E^-(R_i)|$ los cardinales de ambos. Dado un parámetro $k \in [0, 1]$, se dice que la regla R_i es *k-consistente* cuando presenta un número de ejemplos negativos menor o igual que un porcentaje $100 \cdot k$ del número de ejemplos positivos, es decir:

$$R_i \text{ es } k\text{-consistente sii } n_{R_i}^- \leq k \cdot n_{R_i}^+$$

De este modo, la forma de incorporar la verificación de esta propiedad en la metodología propuesta se basará en potenciar la generación de reglas *k-consistentes*. Aquellas reglas que no verifiquen esta propiedad para un valor concreto de k , especificado por el diseñador del proceso, serán penalizadas duramente y tendrán muy pocas posibilidades de formar parte de la base de reglas difusas final.

3.1.2 Mecanismos para Resolver Adecuadamente el Problema de Cooperación-Competición

Como indicábamos en la Introducción de este capítulo, la metodología evolutiva de aprendizaje presentada está basada en el enfoque AIR, por lo que los procesos evolutivos de aprendizaje construidos a partir de la misma serán capaces de resolver directamente el PCC de una forma adecuada. Los SBRDEs en varias etapas basados en el enfoque AIR se caracterizan por resolver el PCC en al menos dos fases, en las que predominan respectivamente la competición y la cooperación entre reglas. El proceso evolutivo de generación de reglas difusas que compone la primera fase trabaja fomentando la competición entre dichas reglas mediante la creación de nichos en el espacio de búsqueda, con el fin de obtener un conjunto formado por las mejores reglas difusas posibles que representen el conocimiento contenido en el conjunto de entrenamiento. En cambio, el propósito de la segunda fase es el obtener la mejor base de reglas difusas y se lleva a cabo buscando el mejor nivel de cooperación posible entre las reglas generadas en la primera etapa. Dependiendo del SBRDE en varias etapas, esta operación se lleva a cabo refinando o simplificando el conjunto de reglas previo. Como puede observarse, trabajando de este modo se reduce enormemente el tamaño del espacio de búsqueda, lo que permite diseñar SBRDs con mejor rendimiento.

En la metodología propuesta se extiende la forma de trabajo habitual de los procesos evolutivos de aprendizaje en varias etapas basados en el enfoque AIR existentes hasta el momento [GP97, HLV97b, Ven92], teniendo en cuenta una serie de criterios de diseño

agrupados en varios bloques según la etapa del SBRDE en la que se consideren. Presentaremos estos criterios en las siguientes secciones.

3.1.2.1 Aspectos de diseño asociados al proceso de generación de reglas difusas

Nuestro modelo evolutivo de aprendizaje se basa en una serie de aspectos novedosos que pueden ser incorporados en la primera etapa del aprendizaje con objeto de mejorar el comportamiento de los SBRDEs construidos a partir del mismo:

- Se permite que el proceso de generación se construya empleando distintos tipos de algoritmos y no sólo a partir de un AG como se hacía en los diferentes procesos evolutivos de aprendizaje basados en el enfoque AIR existentes [GP97, HLV97b, Ven92]. Es posible emplear un algoritmo inductivo o una EE en lugar del habitual AG manteniendo en cualquier caso el mecanismo básico de acción seguido por la primera etapa de este tipo de procesos. El método genera la mejor regla difusa con respecto al estado actual del conjunto de entrenamiento, la almacena en la base de reglas difusas resultante y la penaliza para iteraciones posteriores eliminando los ejemplos ya cubiertos por la base obtenida hasta el momento. La única diferencia radica en la duración del proceso de generación, que es menor en el caso de implementar la primera fase del aprendizaje por medio de una EE en lugar de por medio de un AG, y muy inferior en el caso de considerar un algoritmo inductivo.
- La forma de trabajo habitual empleada por los SBRDEs basados en el enfoque AIR, presentados hasta el momento en la literatura especializada [GP97, HLV97b], no considera la cooperación entre las reglas difusas obtenidas en la primera etapa. Dichas reglas van siendo generadas sin tener en cuenta cómo cooperarán con las ya existentes en la base. De este modo, las reglas nuevas pueden interactuar insuficiente o excesivamente con las generadas anteriormente, provocando que el SBRD obtenido presente un mal rendimiento. Esta característica puede ser considerada con el propósito de mejorar el comportamiento de los SBRDEs construidos a partir del modelo evolutivo de aprendizaje propuesto. Así, cuando planteemos la construcción de procesos evolutivos de diseño de SBRDs aproximativos de tipo Mamdani o de SBRDs de tipo TSK, consideraremos el empleo de algún tipo de técnica que permita afrontar parte del problema de cooperación en la primera etapa del aprendizaje.

En el primero de los casos, pondremos dicha técnica en práctica haciendo uso de un criterio de nichos basado en el esquema de compartición fenotípico que permite generar la mejor regla difusa en cada iteración teniendo en cuenta tanto la bondad de dicha regla como lo bien que coopera con las anteriormente generadas. En el segundo caso, en cambio, lo que emplearemos será una función de adaptación basada en una medida local de error, la cual potenciará la generación de reglas difusas TSK que se ajusten mejor a aquellos ejemplos que caen en su radio directo de influencia, es decir, aquellos que se emparejan en mayor grado con sus antecedentes. Este modo de

generación permitirá que las reglas difusas TSK que están siendo generadas cooperen mejor con las demás reglas de la base para proporcionar una salida en aquellos ejemplos que se emparejan con sus antecedentes en un grado menor.

3.1.2.2 Aspectos de diseño asociados al proceso de postprocesamiento

La tarea de la segunda etapa considerada habitualmente en los SBRDEs basados en el enfoque AIR es la de obtener la definición final de la base de reglas difusas mediante la eliminación de las reglas innecesarias o el refinamiento de las reglas obtenidas en la primera fase. Con objeto de mejorar el comportamiento de los SBRDs diseñados, en nuestra metodología evolutiva se propone llevar a cabo ambas tareas: la simplificación de la base de reglas difusas obtenida en la primera fase y el refinamiento de dichas reglas mediante el ajuste de las funciones de pertenencia que emplean.

Para facilitar la puesta en práctica de ambas tareas, dividimos el proceso de postprocesamiento en dos etapas distintas: el *proceso evolutivo de simplificación o selección* y el *proceso evolutivo de ajuste*, tal y como se hacía en el SBRDE en varias etapas presentado en [HLV97b]. De este modo, el PCC es nuevamente descompuesto en subproblemas más pequeños para afrontar su resolución de un modo más adecuado. El proceso de simplificación se encarga de seleccionar el subconjunto de reglas difusas que presenta el mejor nivel de cooperación de entre las generadas en la fase anterior, pero no altera la definición de las funciones de pertenencia empleadas en éstas, por lo que trabaja sobre un espacio de búsqueda de menor dimensión. Posteriormente, el proceso de ajuste refina esta definición pero trabajando únicamente bien sobre las reglas difusas obtenidas como salida del proceso anterior, o bien sobre las particiones difusas iniciales por lo que, de nuevo, se reduce el tamaño del espacio de búsqueda.

Además, en el modelo evolutivo de aprendizaje presentado en este capítulo, se consideran una serie de aspectos de diseño adicionales para mejorar la estructura de ambos procesos:

- El método que constituye la segunda fase del SBRDE en varias etapas presenta dos características muy importantes. Por un lado, está constituido por un AE basado en el enfoque Pittsburgh, el cual recordemos era el enfoque que mejor resolvía el PCC pero presentaba el inconveniente de tener que manejar un espacio de búsqueda muy amplio. Este inconveniente queda resuelto al particionar el problema de diseño en tres etapas, por lo que el proceso de simplificación trabaja únicamente sobre las reglas generadas en la primera fase y no altera la definición de las funciones de pertenencia. De este modo, aprovecha al máximo la potencialidad que presenta el enfoque Pittsburgh en la resolución del PCC manejando un espacio de búsqueda de tamaño reducido.

Por otro lado, el proceso provoca la aparición de nichos en el espacio de búsqueda, en este caso mediante un enfoque genotípico, lo que permite generar varias soluciones al problema, es decir, distintas bases de reglas difusas formadas por las combinaciones

de las reglas obtenidas en la primera fase que mejor comportamiento presentan. Por esta razón, nos referiremos a él con el nombre de *proceso de multisimplificación*.

- El tercer proceso estará también constituido por un AE que manejará de nuevo un espacio de búsqueda más reducido al tener que trabajar únicamente con las funciones de pertenencia y no con la estructura de las reglas que componen la base de reglas difusas.

Así, el proceso evolutivo de ajuste se aplicará a las distintas bases de reglas difusas obtenidas como salida del proceso evolutivo de multisimplificación, y la mejor adaptada será la proporcionada como salida del proceso global de aprendizaje. De este modo, una base de reglas difusas que no presente el mejor comportamiento después del segundo proceso puede convertirse en la de mejor rendimiento después de la aplicación del tercero debido a que las nuevas funciones de pertenencia den lugar a que las reglas difusas cooperen de un modo más adecuado.

3.1.2.3 Aspectos de diseño asociados a la composición de los Algoritmos Evolutivos considerados

Por último, en la metodología presentada se tiene también en cuenta la importancia de otros factores asociados al diseño de los AEs empleados en cada etapa de los SBRDEs construidos para facilitar la resolución del PCC mediante la realización de una búsqueda adecuada sobre el espacio de soluciones posibles. En la sección 2.3.4 del capítulo anterior presentamos los cinco aspectos claves en el diseño de SBRDEs que permitían la resolución del PCC de una forma adecuada independientemente del esquema evolutivo de aprendizaje empleado. A continuación, indicaremos el tratamiento de estos factores en el modelo evolutivo de aprendizaje propuesto:

1. *Elección del esquema de representación:* El esquema de codificación a emplear dependerá de la etapa en la que actúe el AE y del tipo de información que se represente en el individuo. Principalmente, el más habitual será el esquema de codificación real, puesto que los parámetros que definen las funciones de pertenencia son valores reales, lo que provoca que dicha representación sea la más natural para codificar ese tipo de información. Cuando lo que se desee codificar en el individuo sean las etiquetas que componen una regla lingüística, en la metodología propuesta se sugiere el empleo del esquema de codificación k -ario, en el que se consideran tantos símbolos posibles en el alfabeto como etiquetas pueda tomar como valor una variable concreta.

Con las dos especificaciones anteriores tenemos solucionado el problema de la representación de las reglas difusas aproximativas y descriptivas de tipo Mamdani, así como de los antecedentes de las reglas difusas de tipo TSK. Faltaría por decidir cuál será la codificación a emplear en el caso de los parámetros reales p_i que definen el consecuente de estas últimas. En principio, podría parecer coherente emplear un

esquema de codificación real debido a la naturaleza de dichos valores, pero esta elección entraña un problema: en la mayoría de las ocasiones, los operadores evolutivos empleados en el SBRDE necesitan conocer el intervalo en el que está definida cada componente del individuo, y los intervalos de definición de los parámetros existentes en el consecuente de una regla difusa TSK son desconocidos. Para solventar este problema, en nuestro modelo se propone el empleo de un esquema a medida para representar dichos parámetros, la *codificación angular*, que presentaremos en secciones posteriores.

Finalmente, el esquema escogido para el proceso de simplificación será el binario, ya que la selección del subconjunto de reglas difusas con mejor nivel de cooperación se reduce a un problema de optimización; en concreto a un problema de programación lineal 0 – 1, por lo que dicho esquema resulta ser el más adecuado.

Por otro lado, siempre se tendrá en cuenta la máxima de “representar la mayor cantidad de información en individuos que presenten el menor tamaño posible”. De este modo, se considerará la combinación de los esquemas de representación anteriormente comentados en el caso en el que sea necesario representar distintos componentes de la base de reglas difusas en los individuos de la población genética. Por ejemplo, cuando se desee codificar las etiquetas lingüísticas que componen una regla junto con los conjuntos difusos que definen su semántica, se hará uso de un individuo formado por dos partes distintas: una basada en una codificación entera para representar la etiqueta asociada a cada variable y otra en un esquema real para codificar la forma de las funciones de pertenencia.

2. *Diseño de los operadores evolutivos*: Como indicamos en el capítulo anterior, la definición de los operadores evolutivos a emplear depende del esquema de representación considerado y debe ser efectuada de la mejor manera posible para obtener un equilibrio adecuado entre la exploración y la explotación que el AE efectúa sobre el espacio de búsqueda. Ante esta problemática, existían dos posibles alternativas de diseño en el campo de los SBRDEs: emplear la representación binaria y trabajar con los operadores clásicos, o hacer uso de una representación distinta y diseñar operadores específicos para la misma.

En nuestra metodología, apostamos por esta última vía puesto que, en nuestra opinión, permite resolver de un modo más correcto el PCC al poder trabajar con una representación más adecuada de las soluciones al problema. De este modo, en cada una de las fases que compondrán los distintos SBRDEs en varias etapas construidos a partir del modelo evolutivo se tendrá en cuenta el esquema de representación más adecuado, y se diseñarán operadores capaces de trabajar con individuos que sigan dicho esquema y de realizar una búsqueda adecuada sobre el espacio de posibles soluciones.

Así, nuestro modelo se basa en el empleo de operadores evolutivos para codificación real que permitan obtener un equilibrio adecuado entre las propiedades de exploración y explotación, como los operadores de cruce max-min-aritmético y de

mutación no uniforme propuestos en la sección 2.1.1.3 del capítulo anterior, entre otros. Además, se propone la inclusión de la $EE - (1 + 1)$ como un operador más del AG, para potenciar la explotación de las mejores soluciones obtenidas en cada generación. Este tipo de algoritmos híbridos, agrupados bajo el nombre de *búsqueda local genética*, han demostrado recientemente tener un buen comportamiento en problemas de optimización combinatoria [YI96].

Finalmente, en nuestra metodología se fomenta el empleo de distintos operadores evolutivos que, combinados, potencien respectivamente la exploración y la explotación en el caso de trabajar con individuos que codifican distintos tipos de información, como ocurría en el ejemplo comentado en el punto anterior.

3. *Diseño de la función de adaptación*: La función de adaptación es uno de los elementos más importantes, ya que debe ser capaz de guiar adecuadamente la búsqueda que lleva a cabo el AE por el espacio de soluciones. Debido a esta razón, en la metodología propuesta se potenciará el que las funciones de adaptación empleadas en las distintas etapas de los SBRDEs construidos estén basadas en múltiples criterios, para que así distingan adecuadamente entre buenas y malas soluciones.

Por lo tanto, la composición de la función de adaptación a emplear dependerá también de la etapa del SBRDE en la que se vaya a aplicar. En la primera etapa, se trabajará con criterios frecuentistas de cubrimiento de los ejemplos existentes en el conjunto de entrenamiento, que hagan que la base de reglas difusas generada finalmente verifique las propiedades comentadas en la sección 3.1.1. En las etapas restantes, se combinarán criterios de este tipo con medidas del error cometido por el SBRD codificado sobre el conjunto de entrenamiento. Los primeros darán lugar a que la base de reglas difusas mantenga la satisfacción de las propiedades comentadas, mientras que los segundos permitirán obtener el mejor nivel de cooperación posible entre las reglas difusas que las componen.

Cuando no sea posible trabajar con criterios frecuentistas, es decir, en los procesos evolutivos de aprendizaje de bases de reglas difusas de tipo TSK, este tipo de criterios será sustituido por medidas de error. En estos casos, se considerarán distintos tipos de medidas de error con objeto de mejorar la resolución del PCC. Así, la etapa de generación de reglas empleará medidas locales que evalúen la bondad de las reglas individuales mientras que las etapas posteriores emplearán de nuevo medidas globales para evaluar el nivel de cooperación de la BC TSK codificada en el individuo. *El empleo de medidas locales en la fase de generación de reglas difusas permite además afrontar parte del problema de cooperación en dicha fase.*

4. *Especificación del modo en que se empleará el conocimiento disponible*: Ya hemos indicado que el conocimiento disponible a priori sobre el problema puede ser empleado de muy diversas formas para mejorar el proceso de diseño del SBRD. En la metodología propuesta, se consideran dos formas distintas de emplear este conocimiento:

- *Incorporación de las definiciones parciales obtenidas a partir del conocimiento experto disponible en el proceso evolutivo de aprendizaje:* Debido a la importancia de este punto, lo comentaremos en profundidad en la sección 3.1.4.
- *Empleo del conocimiento disponible para la generación de la población inicial de individuos que maneja el AE:* En todos los casos en los que una etapa del SBRDE sea implementada mediante un AE, se hará uso de este conocimiento para inicializar de una forma adecuada los individuos que componen la primera población, lo que permite al AE converger más rápidamente hacia buenas soluciones.

En la primera etapa, la información numérica existente en el conjunto de entrenamiento se empleará directamente para generar las reglas difusas que compondrán la población inicial. En cambio, en las etapas restantes se hará uso de las definiciones parciales obtenidas en las fases anteriores. Así, las poblaciones iniciales de los procesos de multisimplificación y ajuste incluirán un individuo que codifique al completo la base de reglas difusas obtenida en la fase anterior (o las particiones difusas iniciales empleadas, en los procesos de ajuste de SBRDs descriptivos), mientras que en este último proceso, se considerarán también las definiciones iniciales de las funciones de pertenencia para especificar los intervalos de variación de cada gen, y se inicializarán aleatoriamente los restantes individuos de la población en dichos intervalos.

5. *Definición del esquema de formación de nichos:* En la metodología evolutiva de aprendizaje propuesta, se considerarán las dos posibilidades de empleo del esquema de formación de nichos para el diseño de SBRDs. Por un lado, ya hemos indicado que se hará uso de un criterio basado en el esquema de compartición fenotípico para mejorar el proceso de obtención de soluciones, es decir, para afrontar parte del problema de la cooperación entre reglas difusas en la primera etapa del aprendizaje. Por otro lado, se empleará el otro esquema de compartición existente, el genotípico, para permitir la generación de distintas definiciones de la base de reglas difusas. Analizaremos más detenidamente ambas propuestas en las secciones 3.2 y 3.3.

3.1.3 Estructura de los Sistemas Basados en Reglas Difusas Evolutivos en Varias Etapas

A la vista de los conceptos introducidos en la sección anterior, la estructura general de los SBRDEs en varias etapas construidos a partir del modelo propuesto será la siguiente:

1. *Un proceso de generación de reglas difusas* que permita obtener un conjunto de reglas difusas de cualquier tipo, que represente el conocimiento existente en el conjunto de entrenamiento de un modo adecuado. En todos los casos, este proceso estará formado por dos componentes: un *método de generación de reglas difusas*,

cuya composición depende del tipo de SBRD considerado, y un *método iterativo de cubrimiento* del conjunto de entrenamiento.

El primero de ellos está constituido por un algoritmo inductivo o evolutivo, guiado por una función de adaptación basada, bien en una serie de criterios frecuentistas de la bondad de las reglas, en el caso de SBRDs descriptivos de tipo Mamdani, y en un criterio de nichos para fomentar la cooperación entre las reglas generadas (este último en el caso de SBRDs aproximativos de tipo Mamdani), o bien en una medida local de error que recoge en un solo criterio las dos tareas que lleva a cabo los dos grupos de criterios anteriores, cuando el SBRD a diseñar sea de tipo TSK.

El segundo método, el método iterativo de cubrimiento, es el encargado de poner en práctica el modo de trabajo habitual en la primera etapa de los procesos evolutivos de aprendizaje basados en el enfoque AIR. Así, este proceso penaliza las reglas obtenidas por el método de generación considerando el cubrimiento que producen sobre los ejemplos del conjunto de entrenamiento, y eliminando del mismo los ejemplos ya cubiertos.

2. *Un proceso genético de multisimplificación* para la selección de reglas, siempre que sea necesario simplificar la base de reglas difusas obtenida en el proceso de generación. Este segundo proceso presenta menos flexibilidad a la hora de su puesta en práctica, ya que la necesidad de una representación binaria para el problema de selección de reglas obliga a que tenga que ser implementado mediante un AG. De este modo, la metodología se basa en el uso de un AG con codificación binaria con una función de compartición genotípica y una función de adaptación basada en dos criterios, una medida de error que permita evaluar el rendimiento del SBRD que emplee la base de reglas difusas codificada en el cromosoma y un criterio que penalice la no verificación de la propiedad de completitud por parte de ésta.

Este proceso permite obtener varias definiciones de bases de reglas difusas formadas por las distintas combinaciones de las reglas generadas en la fase anterior que mejor nivel de cooperación presenten.

3. *Un proceso evolutivo de ajuste* basado en cualquier tipo de AE y en una función de adaptación del tipo de la empleada en la fase anterior o que considere únicamente una medida de error para evaluar el rendimiento del SBRD codificado en el cromosoma. Este último proceso proporciona la definición final de la base de reglas difusas ajustando la definición previa de la forma de las funciones de pertenencia empleadas en las reglas manejadas en los dos procesos anteriores.

El tipo de ajuste a efectuar depende de la naturaleza de la base de reglas difusas que maneje el SBRD diseñado. De este modo, cuando se trabaja con un SBRD descriptivo de tipo Mamdani, el ajuste se lleva a cabo de un modo global sobre las particiones difusas asociadas a cada variable lingüística, mientras que en el caso de los SBRDs aproximativos de tipo Mamdani, las funciones de pertenencia se ajustan individualmente para cada regla. En cambio, en el caso de los SBRDs de tipo TSK, la parte antecedente de las reglas se ajusta del mismo modo que en el caso de los

SBRDs descriptivos, mientras que lo que se efectúa sobre la parte consecuente es un refinamiento de la definición inicial de los parámetros p_i obtenida en las fases anteriores.

La definición de la base de reglas difusas que presente el mejor comportamiento tras el ajuste, de entre todas las generadas en el segundo proceso, será la que constituirá la salida final del SBRDE completo.

Como hemos indicado, aunque esta estructura genérica será la que habitualmente presenten los SBRDEs construidos a partir de la metodología propuesta, existe la posibilidad de que éstos estén constituidos únicamente por dos fases, un proceso de generación de reglas difusas y un proceso de ajuste de las funciones de pertenencia. Esta estructura alternativa se dará en aquellos casos en los que la base de reglas difusas obtenida en el proceso de generación no necesite ser simplificada debido a que dicho proceso, aún trabajando a nivel de regla individual, haya sido capaz de obtener un nivel de cooperación adecuado entre las reglas que la componen. Esto provocará que el proceso de ajuste sea empleado solamente para refinar la cooperación entre las reglas generadas en la primera fase, y que no se obtengan finalmente distintas definiciones de la base tras la ejecución de dicho proceso.

3.1.4 Manejo de la Información Disponible para el Diseño de la Base de Reglas Difusas

Una de las características principales de los SBRDs es que son capaces de incorporar en su proceso de diseño los dos tipos de información de los que se suele disponer a la hora de resolver un problema, información de tipo lingüístico y numérico. Por otro lado, como comentamos en el capítulo 2, los AEs presentan la ventaja de su facilidad para incorporar conocimiento a priori sobre el problema, y emplearlo para acelerar el proceso de búsqueda y permitir la obtención de mejores soluciones.

Por tanto, empleando el paradigma evolutivo de aprendizaje automático propuesto, deberíamos ser capaces tanto de aprovechar los dos tipos de información, en el caso en el que ambos estén disponibles, como de permitir la generación de la base de reglas difusas del SBRD en el caso en que únicamente se disponga de un conjunto de datos de entrada-salida que reflejen el comportamiento del sistema que se desea modelar. Así, los SBRDEs construidos a partir de la metodología presentada en este capítulo podrán trabajar de diferentes formas, dependiendo del tipo y la cantidad de información disponibles. A continuación, expondremos las distintas posibilidades comenzando por el caso en el que se dispone de la mínima información posible y finalizando en el que existe conocimiento experto suficiente para obtener una definición preliminar de la base de reglas difusas al completo:

1. *La información disponible se reduce a un conjunto de ejemplos:* En este caso, el

SBRDE en varios niveles construido a partir de la metodología propuesta es aplicado en su totalidad. Para ello, en el caso de trabajar con un SBRD descriptivo de tipo Mamdani o de tipo TSK, se efectúa una definición preliminar de la BD, seleccionando un número adecuado de etiquetas lingüísticas para cada variable de entrada, y asociando a cada una de ellas un conjunto difuso obtenido de una equipartición difusa del espacio de entrada en el primer caso, y de los espacios de entrada y salida en el segundo.

2. *Además de disponer del conjunto de ejemplos, el conocimiento experto permite especificar los términos lingüísticos existentes y los conjuntos difusos asociados:* En este segundo caso, se procede de un modo exactamente igual al anterior, pero se hace uso de la definición existente de la BD desde la primera etapa del SBRDE.
3. *Se dispone de una base de reglas difusas incompleta y de un conjunto de ejemplos:* Adicionalmente, en el caso de trabajar con SBRDs descriptivos de tipo Mamdani o de tipo TSK, puede o no existir una definición previa de la BD como en el caso anterior. Se procede, en primer lugar, incorporando las reglas lingüísticas obtenidas a partir de los expertos a la base de reglas difusas a generar en el primero de los procesos que constituye el SBRDE en varias etapas (sea ésta de tipo Mamdani descriptivo o aproximativo, o de tipo TSK), la cual estaba inicialmente vacía en los dos casos anteriores. Tras estudiar el cubrimiento que dichas reglas provocan sobre los ejemplos del conjunto de entrenamiento y eliminar del mismo los ejemplos ya cubiertos, se ejecuta el proceso iterativo de generación hasta obtener un conjunto de reglas difusas que contenga todo el conocimiento existente en el conjunto de entrenamiento. Finalmente, se ejecutan las restantes etapas que componen el SBRDE.
4. *Se dispone de una definición inicial completa de la base de reglas difusas y de un conjunto de ejemplos.* Finalmente, en el caso en el que el experto humano haya sido capaz de proporcionar una definición completa de la base, o ésta haya sido obtenida a partir de otro tipo de método de aprendizaje automático, es posible hacer uso únicamente del proceso evolutivo de ajuste para refinar la definición de las funciones de pertenencia empleadas en las reglas difusas, con objeto de obtener un SBRD final con un mejor rendimiento.

3.2 El Proceso de Generación de Reglas Difusas

Como comentábamos en secciones anteriores, el proceso de generación de reglas difusas está compuesto por el *método de generación de reglas difusas* y el *método iterativo de cubrimiento*. Dedicaremos las dos secciones siguientes a describir las características generales de ambos métodos y sus implementaciones particulares en los SBRDEs que presentaremos en los capítulos 4, 5 y 6 de esta memoria.

3.2.1 El Método de Generación de Reglas Difusas

Este primer método es el encargado de obtener la mejor regla difusa en cada momento con respecto al estado actual del conjunto de entrenamiento. Como ya se ha indicado, la composición de este proceso dependerá del tipo de regla difusa con el que se esté trabajando. De este modo, existen las siguientes posibilidades:

- *Diseño de SBRDs descriptivos de tipo Mamdani:* En este caso, la primera fase del SBRDE en varias etapas no tiene que trabajar con las funciones de pertenencia, que vienen definidas por las particiones difusas iniciales, sino únicamente con las etiquetas lingüísticas en sí. Esto reduce enormemente el espacio de búsqueda, por lo que no es necesario emplear un AE para llevar a cabo la generación de la mejor regla lingüística. En este caso, se propone el uso de un algoritmo inductivo para poner en práctica el método de generación. Dicho algoritmo empleará una función de selección de reglas lingüísticas que deberá estar basada en una serie de criterios frecuentistas y que debe permitir ordenar dichas reglas con respecto a su comportamiento.

En el capítulo 4 de esta memoria, introduciremos un algoritmo inductivo para generar reglas lingüísticas a partir de ejemplos, que será empleado para implementar este método. Posteriormente, presentaremos los criterios frecuentistas considerados en la función de selección de reglas lingüísticas empleada por dicho algoritmo.

- *Diseño de SBRDs aproximativos de tipo Mamdani:* La situación es distinta cuando se manejan reglas aproximativas de este tipo, ya que en este caso sí es necesario que el método de generación de reglas difusas sea capaz de obtener las funciones de pertenencia empleadas en las reglas que genere. Por esta razón, el espacio de búsqueda presenta una dimensión mucho mayor que en el caso anterior, con lo que es recomendable emplear algún tipo de AE. La función de adaptación estará basada en los mismos criterios frecuentistas empleados para las reglas descriptivas y en el criterio de nichos comentado, que permite afrontar parte de la cooperación entre reglas difusas en la primera fase.

En el capítulo 5 de esta memoria presentaremos dos posibilidades para el diseño de este método basadas en el empleo de una $EE - (1 + 1)$, y de un AE híbrido compuesto por un AG que emplea una $EE - (1 + 1)$ como operador genético.

- *Diseño de SBRDs de tipo TSK:* Por último, en el caso en que el SBRD a diseñar sea de tipo TSK y presente matices descriptivos, la situación es intermedia, ya que el método de generación no tiene que aprender la semántica asociada a las variables lingüísticas del antecedente de las reglas, pero sí tiene que obtener una definición preliminar de los parámetros reales p_i que definen el consecuente. De este modo, de nuevo se recomienda el uso de un AE con un esquema de representación que combine una codificación entera para representar las etiquetas lingüísticas existentes en los antecedentes de las reglas, y una codificación angular, que introduciremos en la sección 3.2.1.4, para los parámetros que definen los consecuentes. Dicho algoritmo

estará guiado, en este caso, por una función de adaptación basada en una medida local de error que permita afrontar la cooperación en esta fase del aprendizaje.

En el capítulo 6 describiremos un método de generación de reglas difusas TSK a partir de ejemplos, basado en una $EE - (\mu, \lambda)$ que sigue las ideas comentadas.

En las tres secciones siguientes presentaremos algunos de los criterios que permiten diseñar las funciones de adaptación empleadas en cada implementación del método de generación de reglas difusas. La cuarta sección estará dedicada a introducir la codificación angular empleada para representar los consecuentes de las reglas difusas TSK en el tercer caso de diseño.

3.2.1.1 Los criterios frecuentistas considerados para la generación de reglas lingüísticas

Los criterios frecuentistas considerados para formar parte de la función multicriterio de selección de reglas lingüísticas deben ser tales que las reglas que presenten mejores valores en la misma sean prometedoras para formar parte de una BC que presente buen comportamiento; además, se deben de verificar las propiedades de completitud, cubrimiento y consistencia. Entre otros, podemos considerar los siguientes criterios:

- a) **Alto valor de frecuencia** [HLV97b]: La frecuencia de una regla difusa, R_i , sobre un conjunto de ejemplos, E_p , se define de la forma:

$$\Psi_{E_p}(R_i) = \frac{\sum_{l=1}^p R_i(e_l)}{p}$$

donde p es el número de ejemplos de E_p , y $R_i(e_l)$ es el grado de compatibilidad entre la regla R_i y el ejemplo e_l introducido en la sección 3.1.1.

- b) **Alto grado de cubrimiento medio sobre los ejemplos positivos** [HLV97b]: El conjunto de ejemplos positivos para la regla R_i con grado de compatibilidad mayor o igual que ω se define de la forma:

$$E_{\omega}^{+}(R_i) = \{e_l \in E_p / R_i(e_l) \geq \omega\}$$

con $n_{\omega}^{+}(R_i)$ igual a $|E_{\omega}^{+}(R_i)|$. El *grado de cubrimiento medio* sobre $E_{\omega}^{+}(R_i)$ puede definirse como:

$$G_{\omega}(R_i) = \sum_{e_l \in E_{\omega}^{+}(R_i)} R_i(e_l) / n_{\omega}^{+}(R_i)$$

c) **Penalización debida a la no satisfacción de la propiedad de k -consistencia:**

Este último criterio penaliza aquellas reglas que presentan un número excesivo de ejemplos negativos con respecto al número de ejemplos positivos a grado ω . Dicho criterio penaliza, por tanto, la no verificación de la propiedad de k -consistencia. La *función de penalización sobre el conjunto de ejemplos negativos de la regla R_i* viene representada por la expresión:

$$g_n^-(R_i) = \begin{cases} 1, & \text{si } n_{R_i}^- \leq k \cdot n_{\omega}^+(R_i) \\ \frac{1}{n_{R_i}^- - k \cdot n_{\omega}^+(R_i) + \exp(1)}, & \text{en otro caso} \end{cases}$$

Es importante destacar el hecho de que el conjunto de ejemplos negativos para una regla se obtiene trabajando sobre el conjunto de entrenamiento E_p inicial y no sobre el actual, del que ya se han podido eliminar algunos ejemplos. Esto se debe a que, si se calculase a partir del conjunto actual, podría darse el caso de que se generasen reglas difusas que fuesen inconsistentes con otras ya obtenidas anteriormente, debido a no haber considerado la influencia de los ejemplos ya eliminados en su generación.

3.2.1.2 El criterio de baja interacción entre nichos considerado para la generación de reglas aproximativas de tipo Mamdani

• **Justificación del empleo del criterio de baja interacción entre nichos:**

Este nuevo criterio se emplea en combinación con otros del tipo de los presentados en la sección anterior para formar la función de adaptación empleada en el método de generación de reglas difusas en el caso de estar diseñando un SBRD aproximativo de tipo Mamdani. Tiene su razón de ser, al posibilitar la resolución de parte de la cooperación entre reglas difusas en la primera fase del aprendizaje, como veremos a continuación.

Puesto que la función de adaptación empleada en el método de generación de reglas aproximativas de tipo Mamdani está basada inicialmente en una serie de criterios frecuentistas, la competición producida por dicho método da lugar a que las reglas difusas aproximativas codificadas en los distintos individuos tiendan a ampliar los soportes de los conjuntos difusos que emplean, para extender su área de aplicación y cubrir más ejemplos, obteniendo así una mayor recompensa de la función de adaptación. Esto hace que se obtenga finalmente una BRD en la que las reglas difusas aproximativas interactúan de un modo excesivo. Como consecuencia, dichas reglas no cooperan adecuadamente y el SBRD diseñado no hace un uso correcto de su capacidad de razonamiento interpolativo.

Teniendo en cuenta estos hechos, en nuestra metodología se propone el fomentar la aparición de un nuevo tipo de nichos en el proceso de generación de reglas difusas, aparte del que el propio proceso induce por su forma habitual de trabajo basada en el enfoque AIR (véase la sección 3.2.2). Este nuevo tipo de nichos, inducido ahora por el método de generación, se basa en compartir la recompensa asociada a la regla que está siendo generada cuando se encuentra dentro del nicho formado por alguna de las generadas anteriormente. Por lo tanto, este criterio de nichos tenderá a estrechar los soportes de los conjuntos

difusos considerados en la regla que está siendo generada actualmente, para así evitar la penalización que recibe cuando interactúa demasiado con las obtenidas anteriormente.

Como puede observarse, el mejor esquema de compartición para llevar a cabo la tarea comentada parece ser el fenotípico, ya que con él se trabaja directamente con las reglas difusas en sí y es más sencillo el penalizar un nivel de interacción excesivo mediante la función de compartición. La regla difusa aproximativa generada en cada iteración del método de cubrimiento formará un nicho en el espacio de soluciones y el tamaño del mismo dependerá de la forma de los conjuntos difusos que emplee dicha regla, como veremos a continuación. Cuando se proceda a generar una nueva regla en las iteraciones posteriores, los individuos existentes en la población manejada por el método de generación de reglas difusas que estén situados en uno de esos nichos estarán obligados a compartir su recompensa con las reglas difusas aproximativas que provocan la aparición de éstos.

De este modo, *una función de adaptación que combine algunos criterios frecuentistas y un criterio de penalización de nichos de este tipo permitirá obtener reglas con el mejor comportamiento posible que presenten un nivel de interacción adecuado con las reglas vecinas ya generadas en las ejecuciones anteriores.* Gracias a ello, las reglas difusas aproximativas que compongan la BRD generada finalmente cooperarán adecuadamente, con lo que el SBRD diseñado presentará un mejor rendimiento.

• **Diseño del criterio de baja interacción entre nichos:**

Centrándonos ya en el diseño de un criterio concreto que nos permita poner en práctica el comportamiento deseado, lo primero a tener en cuenta es el inconveniente más importante en los esquemas de compartición clásicos: la necesidad de conocer *dónde* está cada nicho y *cuál es su tamaño* para así poder llevar a cabo el proceso de compartición de las recompensas entre los individuos. Como comentábamos en la sección 2.1.1.4 del segundo capítulo, este requisito suele ser satisfecho asumiendo que si dos individuos se encuentran lo suficientemente cerca, a una distancia conocida como *radio de nicho*, entonces deben compartir sus recompensas entre sí. Aunque existen algunos métodos que permiten determinar este radio, su cálculo es una tarea muy complicada en la gran mayoría de los casos y, además, se obliga a que todos los nichos presenten el mismo tamaño, lo que no es siempre recomendable.

Afortunadamente, en nuestro caso es posible determinar la localización y el tamaño de los distintos nichos existentes. Puesto que trabajamos en el espacio fenotípico, cada individuo representa una regla formada por n variables de entrada y una de salida. Considerando que los conjuntos difusos asociados a cada variable son triangulares, podemos pensar en el centro del nicho inducido por la regla en cuestión como un punto en el espacio $(n + 1)$ -dimensional cuyas coordenadas corresponden a los puntos modales de las funciones de pertenencia triangulares. Dos individuos compartirán la recompensa que reciban cuando exista algún tipo de interacción entre los conjuntos difusos asociados a la misma variable en las dos reglas que codifican. De este modo, el algoritmo no considera un valor fijo de radio de nicho como en el esquema clásico de compartición, sino que el tamaño

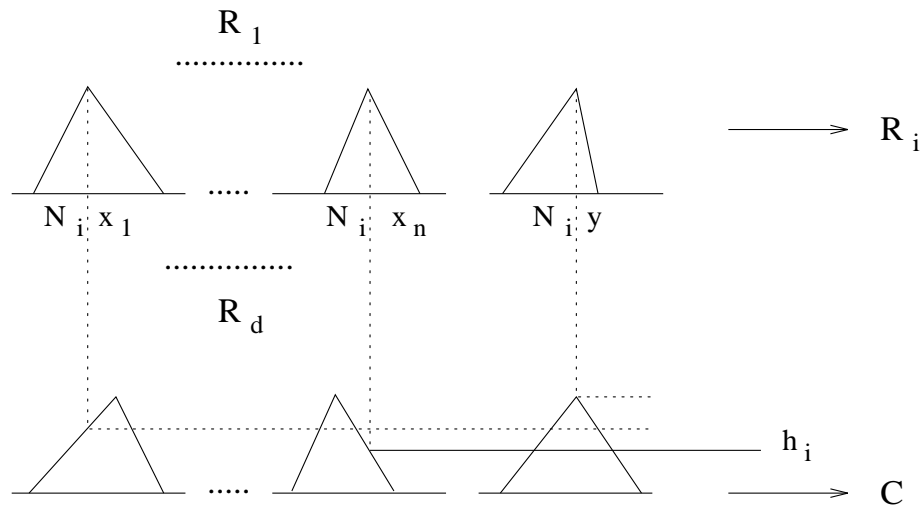


Figura 3.1. Interacción entre la regla actual y las generadas anteriormente

del nicho depende de la forma de las funciones de pertenencia empleadas en las reglas generadas en las distintas iteraciones del método de cubrimiento.

Sean $N_i = (N_{ix}, N_{iy})$, $i = 1, \dots, d$, los centros de dichas reglas (nichos), donde d es el número de iteraciones del método de cubrimiento realizadas, y sea C el individuo que codifica la regla difusa aproximativa que está siendo adaptada. El *criterio de baja interacción entre nichos*, $BIEN(\cdot)$, penaliza el valor de la función de adaptación asociado a C , haciendo uso de la medida de interacción entre nichos, $IEN(\cdot)$, del siguiente modo:

$$BIEN(C) = 1 - IEN(C)$$

$$IEN(C) = \text{Max}_i\{h_i\}, \quad i = 1, \dots, d$$

$$h_i = *(A_1(N_{ix_1}), \dots, A_n(N_{ix_n}), B(N_{iy}))$$

$$C \sim \text{Si } x_1 \text{ es } A_1 \text{ y } \dots \text{ y } x_n \text{ es } A_n \text{ entonces } y \text{ es } B$$

donde $*$ es una t-norma. La figura 3.1 representa gráficamente una situación en la que existe interacción entre la regla codificada en C y alguna de las reglas generadas hasta el momento.

De este modo, el criterio $BIEN(\cdot)$ está definido en el intervalo $[0, 1]$. Tomará su máximo valor, por lo que no se penalizará la regla codificada en C , cuando dicha regla no interactúe con ninguna de las generadas hasta ahora. En cambio, tendrá asociado el valor mínimo, que repercute en la máxima penalización posible, cuando la regla codificada sea exactamente igual a alguna de las ya obtenidas.

Por otro lado, la composición concreta de este criterio depende directamente de la medida de interacción entre nichos $IEN(\cdot)$ empleada. Considerando distintas definiciones para esta última, es posible obtener varios criterios $BIEN(\cdot)$ con filosofías diferentes.

3.2.1.3 La medida local de error considerada para la generación de reglas difusas TSK

Como ya hemos comentado anteriormente a lo largo de este capítulo, la función de adaptación a emplear en el método de generación de reglas difusas de tipo TSK no puede hacer uso de criterios frecuentistas como los mostrados en las secciones anteriores debido a la propia composición del consecuente de este tipo de reglas. Por esta razón, los criterios a considerar en estos casos se basan en medidas de error que reflejen el grado de aproximación obtenido por la regla codificada sobre los ejemplos del conjunto de entrenamiento.

Con objeto de poder afrontar parte de la cooperación entre las reglas difusas generadas en la primera fase, en la metodología presentada se sugiere el empleo de una medida de error local para que las reglas generadas se ajusten en mayor medida a los ejemplos de la base que más se emparejen con ellas, dando lugar a que el ajuste de los que se emparejan en un grado menor con sus antecedentes se efectúe mediante la acción combinada de distintas reglas difusas de la base.

De este modo, el diseño de la función de adaptación se puede realizar empleando como único criterio una medida de error de este tipo, ya que permite satisfacer los dos objetivos que se plantean: obtener reglas difusas TSK que presenten un buen comportamiento y que cooperen adecuadamente entre sí.

Como ejemplo de un criterio de este tipo podemos emplear el siguiente, propuesto en [YG95]:

$$\sum_{e_l \in E} h^l \cdot (ey^l - S(ex^l))^2$$

donde E es un conjunto formado por los ejemplos existentes en E_p que están contenidos en el subespacio difuso definido por los antecedentes de la regla codificada en el individuo C , es decir, aquellos ejemplos de E_p que se emparejan con los antecedentes de la regla actual a cualquier grado; $h^l = T(A_1(ex_1^l), \dots, A_n(ex_n^l))$ es el grado de emparejamiento existente entre las componentes de entrada, ex^l , de cada uno de esos ejemplos y los antecedentes de dicha regla (T es una t-norma); y, por último, $S(ex^l)$ es la salida que devuelve la regla difusa TSK codificada en C al recibir ex^l como entrada.

3.2.1.4 Un nuevo esquema para la representación de los consecuentes de las reglas difusas TSK: La codificación angular

Como ya hemos comentado, cuando se considera el diseño de SBRDs de tipo TSK empleando AEs se plantea el problema de que es necesario conocer los intervalos en los que se definen las distintas variables para así poder obtener una solución. Esta información es necesaria para definir el esquema de representación de las posibles soluciones y adaptarlas mediante los operadores evolutivos. Desafortunadamente, estos intervalos de definición

son desconocidos en el aprendizaje de los parámetros asociados a los consecuentes de las reglas difusas de tipo TSK.

Normalmente, los distintos autores que se han enfrentado con este problema [LS95, LT93a, LT93b, Wie94, YG95] lo han resuelto fijando unos valores suficientemente grandes para los límites inferior y superior de dichos intervalos de definición. Esta no es una mala solución, ya que el poder de búsqueda de los AEs les permite obtener buenas soluciones trabajando de este modo, pero presenta el inconveniente de que no se considere el espacio de soluciones al completo, por lo que puede que no sea posible encontrar la mejor solución del problema debido a que el valor óptimo de alguno de los distintos parámetros esté situado fuera de los intervalos considerados.

En esta sección, vamos a proponer un nuevo esquema de representación denominado *codificación angular*, para solventar el problema de una forma más adecuada. El esquema está basado en codificar los valores de los ángulos que forman los consecuentes de una regla difusa TSK en el espacio, en lugar de los valores de las tangentes, que son los codificados habitualmente. Este cambio permite que todas las variables estén definidas en el mismo intervalo fijo y que sea posible representar todo el espacio posible de soluciones de una forma sencilla.

Es conocido el hecho de que la relación parcial definida por los consecuentes de una regla difusa TSK determina un hiperplano en el hiperespacio correspondiente. Por ejemplo, en el caso de trabajar con un SBRD TSK con una única variable de entrada, cada consecuente de las reglas TSK presenta la forma $Y = p_1 \cdot X + p_0$, la cual representa una línea recta en una parte del plano (dicha parte está definida por el antecedente de la regla, que determina el área de actuación de la misma). A la vista de este hecho, sabemos que el valor real p_1 no es otra cosa que la tangente del ángulo existente entre dicha recta y el eje X . De este modo, en el caso de codificar el valor de este ángulo en lugar del de la tangente haciendo uso de la función

$$C : \mathbb{R} \rightarrow \left(-\frac{\pi}{2}, \frac{\pi}{2}\right) \\ C(x) = \arctan(x) ,$$

todos los valores posibles del parámetro p_1 están definidos en el intervalo abierto $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$. La figura 3.2 recoge gráficamente algunos ejemplos de la idea comentada.

Como puede observarse en dicha figura, mediante el empleo de intervalos muy pequeños se representa una zona muy amplia del espacio solución. Por ejemplo, considerando el intervalo real $[-20, 20]$ para los parámetros p_i , estamos trabajando con el intervalo angular $[-87.13^\circ, 87.13^\circ]$. Por ello, únicamente no consideramos del orden de un 3.3 por ciento del espacio de búsqueda (más o menos 5.74°). Esta conclusión justifica el hecho de que los SBRDEs que consideraban intervalos de tamaño fijo para el aprendizaje de los consecuentes de las reglas difusas TSK consiguieran buenos resultados. De todos modos, parece más apropiado el trabajar con el espacio de búsqueda al completo cuando se afronta una búsqueda en pos de una solución óptima.

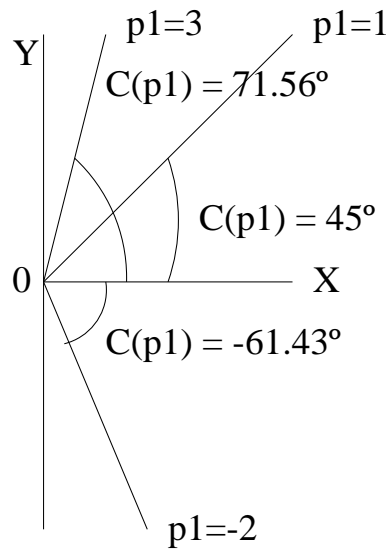


Figura 3.2. Ejemplos de aplicación de la codificación angular

Finalmente, cuando se trabaja en el plano, el parámetro p_0 determina el desplazamiento de la recta sobre el eje Y con respecto al origen, tal y como se observa en la figura 3.3. Puesto que los valores de este parámetro pueden variar en las distintas reglas difusas TSK que componen la base, la consideración de un intervalo fijo de definición no es una buena solución cuando se pretende afrontar su aprendizaje mediante un AE, por lo que la codificación angular se convierte en una herramienta poderosa para resolver el problema. En este caso, no existe una interpretación geométrica de la codificación empleada (téngase en cuenta que p_0 no se corresponde con la tangente de ningún ángulo en el hiperespacio concreto); únicamente hacemos uso de la codificación angular para establecer una aplicación entre un intervalo con extremos no definidos, \mathbb{R} , y otro que si los tiene, $(-\frac{\pi}{2}, \frac{\pi}{2})$.

De este modo, la transformación presentada permite efectuar una búsqueda adecuada sobre el espacio de soluciones, haciendo uso de un AE para aprender los valores de éstos parámetros. Sin embargo, para utilizar adecuadamente dicha transformación, será necesario considerar su naturaleza no lineal, que repercutirá en los dos factores siguientes:

- *Generación de la población inicial del AE:* Para efectuar la generación aleatoria de la población inicial de soluciones, se ha de usar una distribución de probabilidad no uniforme. Este hecho se debe a que, al ser no lineal la transformación considerada, una distribución uniforme en el genotipo daría lugar a una no uniforme en el fenotipo, lo que sesgaría la distribución de los individuos de la población inicial en el espacio de soluciones del problema.
- *Diseño de los operadores evolutivos:* En el caso en el que los operadores evolutivos considerados realicen operaciones aritméticas o manejen distribuciones de probabilidad, se va a dar un hecho similar al comentado en el punto anterior. Estas

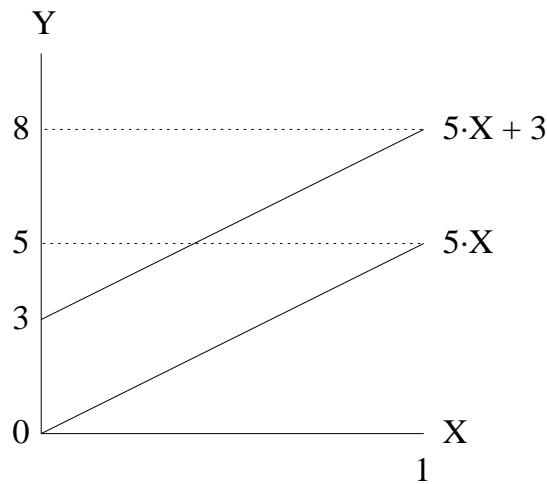


Figura 3.3. Interpretación geométrica del parámetro p_0

operaciones perderán por completo sus propiedades originales al ser transformadas no linealmente. Por esta razón, será necesario diseñar adecuadamente los operadores evolutivos para que trabajen directamente sobre un genotipo angular, evitando este tipo de problemas.

En el capítulo 6 de esta memoria, introduciremos un método de generación de consecuentes de reglas difusas TSK, basado en una $EE - (\mu, \lambda)$, que tendrá en cuenta las ideas comentadas.

3.2.2 El Método Iterativo de Cubrimiento

Este segundo componente del proceso de generación está basado en un proceso iterativo que permite generar un conjunto de reglas difusas de cualquier tipo, que represente el conocimiento incluido en el conjunto de entrenamiento. El algoritmo iterativo considerado fue presentado en [HLV97b]. En cada iteración, se ejecuta el método de generación para obtener la mejor regla difusa en el momento actual, se considera el cubrimiento de dicha sobre el conjunto de ejemplos, y se elimina del mismo aquellos ejemplos cubiertos a grado ϵ , valor especificado por el diseñador del SBRD. El algoritmo empleado se muestra a continuación:

1. *Inicialización:*

- (a) *Introducir los valores de los parámetros considerados.*
- (b) *En el caso en que existan reglas lingüísticas obtenidas a partir de conocimiento experto:*
 - i. *Introducir dichas reglas en el conjunto final de reglas difusas B^g .*

- ii. Inicializar el grado de cubrimiento del ejemplo e_l , $GC(e_l)$, $l = 1, \dots, p$, considerando el cubrimiento de las reglas contenidas en B^g sobre los ejemplos existentes en el conjunto E_p .
 - iii. Eliminar de E_p aquellos ejemplos en los que $GC(e_l) \geq \epsilon$.
- (c) En caso contrario:
- i. Inicializar el conjunto final de reglas difusas $B^g \leftarrow \emptyset$.
 - ii. Inicializar $GC(e_l) \leftarrow 0$, $l = 1, \dots, p$.
2. Ejecutar el método de generación para obtener la mejor regla difusa R_r con respecto al estado actual del conjunto de ejemplos E_p . Introducir R_r en B^g .
3. Para cada ejemplo $e_l \in E_p$ hacer
- (a) $GC(e_l) \leftarrow GC(e_l) + R_r(e_l)$,
 - (b) Si $GC(e_l) \geq \epsilon$ entonces eliminar e_l de E_p .
4. Si $E_p = \emptyset$ entonces Parar. En caso contrario, volver al paso 2.

Ya hemos indicado que la primera fase de los SBRDEs en varias etapas basados en el enfoque AIR está basada en fomentar la aparición de nichos en el espacio de búsqueda. Para ello, el método de cubrimiento procede eliminando ejemplos del conjunto de entrenamiento, con lo que anula la recompensa asociada a las zonas del espacio en las que se encontraban dichos ejemplos. Esto constituye una modificación de alto nivel sobre la función de adaptación, ya que traslada el foco de la búsqueda, de una zona del espacio a otra, fomentando el que se produzca una buena exploración del espacio de búsqueda.

Como puede observarse, esta forma de trabajo asegura la generación de reglas difusas en cada zona del espacio en la que el problema esté definido, es decir, en todas aquellas en las que existen ejemplos. Además, en el caso en el que se considere una función de adaptación basada en criterios frecuentistas en el método de generación, la combinación del modo de trabajo seguido por el método iterativo de cubrimiento y de estos criterios, asegura la verificación de las propiedades de consistencia y completitud por parte de la base de reglas difusas obtenida como salida de la primera fase del SBRDE.

Por otro lado, cuando el método de generación se encarga de obtener reglas difusas aproximativas de tipo Mamdani y emplea el criterio de baja interacción entre nichos, se efectúa además una modificación de bajo nivel en la función de adaptación, que permite obtener una distribución adecuada de las reglas en cada nicho existente en el espacio de soluciones; esto traduce en un buen nivel de interacción entre dichas reglas. En el momento en que se ha localizado un nicho en una zona del espacio y dicha zona continúa siendo la más prometedora (es decir, los ejemplos situados en la misma no han sido cubiertos aún a grado ϵ y tienen asociada una recompensa alta), el proceso de generación procede a obtener nuevas reglas en la misma zona, que interactuarán con las generadas anteriormente. El criterio de baja interacción entre nichos es el encargado de que estas reglas consigan un

buen nivel de cooperación. Esta segunda modificación puede ser considerada, como hemos comentado, como de bajo nivel ya que el algoritmo continúa trabajando en la misma zona del espacio, pero penaliza la excesiva proximidad entre los nichos existentes en ella. De este modo, fomenta una buena explotación del espacio.

3.3 El Proceso Genético de Multisimplificación

Como ya comentábamos en la sección 2.2.3 del capítulo 2, en [BBM93] se presenta un método evolutivo de optimización de funciones multimodales, que permite obtener el número deseado de óptimos haciendo uso de un esquema de formación de nichos. La forma de trabajo de la *Técnica de Nichos Secuenciales* se basa en iterar un proceso de optimización de funciones unimodales, en concreto, un AG, para obtener un óptimo en cada ejecución, y en considerar cada uno de estos óptimos generados como el centro de un nicho, al ser la mejor solución encontrada en una zona concreta del espacio. Con el propósito de alejar el foco de la búsqueda de las zonas en las que ya se ha encontrado un óptimo para buscar en otras distintas, la técnica hace uso de una función de penalización que modifica la recompensa asociada a los individuos según la distancia que exista entre ellos y los nichos previamente determinados en el espacio.

Como puede observarse, esta forma de trabajo es muy similar a la que llevan a cabo los procesos de generación de reglas en los SBRDEs en varias etapas basados en el enfoque AIR, descrita detenidamente en la sección anterior. Tomaremos estas ideas como base para diseñar nuestro proceso genético de multisimplificación.

Como también indicábamos en la sección 2.3.4 del capítulo anterior, la idea de emplear el esquema de formación de nichos para generar distintas definiciones posibles de la base de reglas difusas ya ha sido empleada por Satyadas y Krishnakumar en [KS95], donde los autores emplean un AG con un esquema de compartición fenotípico para generar distintas BRLs candidatas a formar parte de la BC final en un SBRDE para el diseño de SBRDs descriptivos de tipo Mamdani.

En los SBRDEs en varias etapas construidos a partir de la metodología propuesta, haremos uso de la Técnica de Nichos Secuenciales para fomentar la formación de nichos en el espacio de búsqueda y obtener varias definiciones de bases de reglas difusas de cualquier tipo formadas por las distintas combinaciones de las reglas difusas generadas en la fase anterior con mejor nivel de cooperación. Para ello, emplearemos el proceso genético de simplificación propuesto en [HLV97b] como algoritmo básico de optimización unimodal a ejecutar en cada iteración del proceso de multisimplificación. La forma de trabajo característica de la Técnica de Nichos Secuenciales, basada en compartir las recompensas existentes entre el individuo evaluado en ese momento y las soluciones encontradas en las ejecuciones anteriores del proceso (en lugar de entre el individuo evaluado y los restantes

que forman la población genética), permite obtener una combinación de reglas difusas distinta en cada iteración del algoritmo de multisimplificación.

En nuestros SBRDEs, hemos solucionado el problema de la selección de la mejor definición de la base de reglas difusas de una forma más sencilla que la considerada en [KS95], basada en el empleo de un método de Clustering. Puesto que la definición general de las funciones de pertenencia asociada a todas las bases obtenidas como salida del proceso genético de multisimplificación es refinada para cada una de ellas en particular, en la tercera etapa del SBRDE, esto es, el proceso evolutivo de ajuste, la base de reglas difusas obtenida como salida final del proceso evolutivo de aprendizaje será la que mejor comportamiento presente después del ajuste.

En las dos secciones siguientes presentamos el algoritmo básico de simplificación y los aspectos particulares del proceso genético de multisimplificación, respectivamente.

3.3.1 El Proceso Genético Básico de Simplificación

Este proceso está basado en un AG con codificación binaria, en el que la selección de los individuos se lleva a cabo haciendo uso del mecanismo de muestreo universal estocástico de Baker y el modelo elitista; y la generación de la nueva población se efectúa mediante los operadores clásicos de cruce multipunto en dos puntos y mutación aleatoria.

El esquema de codificación genera cromosomas de longitud fija. Si numeramos de 1 a m las reglas contenidas en el conjunto de reglas lingüísticas B^g , generado en la fase anterior, una cadena de m bits, $C = (c_1, \dots, c_m)$, representará un subconjunto de reglas candidatas a formar parte de la base de reglas difusas, B^s , obtenida finalmente como salida del proceso de tal modo que:

$$\text{Si } c_i = 1 \text{ entonces } R_i \in B^s, \text{ en caso contrario } R_i \notin B^s$$

El conocimiento disponible se incorpora a la población inicial mediante la inclusión de un cromosoma C_1 que codifica una base de reglas difusas, B^g , formada por la totalidad de las reglas obtenidas en el proceso anterior. Los demás cromosomas de la población son generados aleatoriamente.

En lo que respecta a la función de adaptación, $E(\cdot)$, está basada en el error cuadrático medio cometido por un SBRD que emplee la base de reglas difusas codificada en el individuo al actuar sobre las entradas contenidas en el conjunto de entrenamiento. Esta medida de error viene representada por la siguiente expresión:

$$E(C_j) = \frac{1}{2 \cdot |E_p|} \sum_{e_i \in E_p} (\epsilon y^l - S(\epsilon x^l))^2$$

donde $S(ex^l)$ es la salida proporcionada por el SBRD que emplea la base codificada en C_j cuando recibe ex^l como entrada y ey^l es el valor de salida deseado almacenado en el conjunto de entrenamiento.

Para conseguir que se siga verificando la propiedad de completitud considerada en la fase previa, obligaremos a que todos los ejemplos del conjunto de entrenamiento sean cubiertos en un grado mayor o igual que τ por la base de reglas difusas codificada en el cromosoma C_j , $R(C_j)$:

$$C_{R(C_j)}(e_l) = \bigcup_{j=1..T} R_j(e_l) \geq \tau, \quad \forall e_l \in E_p \text{ y } R_j \in R(C_j)$$

donde τ es el grado mínimo de completitud exigido a las bases generadas en el proceso de simplificación. Normalmente, el valor de τ es menor o igual que el del parámetro ω , que representa el grado de compatibilidad de los ejemplos positivos empleado en el proceso de generación.

De este modo, definimos el *grado de completitud sobre el conjunto de entrenamiento* de la base de reglas difusas $R(C_j)$ sobre el conjunto de ejemplos E_p de la forma:

$$GCC E(R(C_j), E_p) = \bigcap_{e_l \in E_p} C_{R(C_j)}(e_l)$$

y la función de adaptación empleada finalmente, la cual penaliza la no verificación de la propiedad de completitud, es la siguiente:

$$F(C_j) = \begin{cases} E(C_j), & \text{si } GCC E(R(C_j), E_p) \geq \tau \\ \frac{1}{2} \sum_{e_l \in E_p} (ey^l)^2, & \text{en otro caso} \end{cases}$$

3.3.2 Composición del Proceso Genético de Multisimplificación

Para provocar la aparición de nichos en el algoritmo de Nichos Secuenciales, es necesario definir algún tipo de métrica que, dados dos individuos, devuelva una medida de la distancia existente entre ambos [BBM93]. La naturaleza de esta métrica, es decir, el espacio solución sobre el que está definida, genotípico o fenotípico, determina el esquema de nichos empleado. En nuestro proceso de multisimplificación haremos uso de un *esquema de compartición genotípico*, ya que la métrica considerada será la distancia de Hamming medida en el espacio de codificación binario. Dados dos individuos $A = (a_1, \dots, a_m)$ y $B = (b_1, \dots, b_m)$, la métrica escogida se define de la forma:

$$H(A, B) = \sum_{i=1}^m |a_i - b_i|$$

La *función de adaptación modificada*, que guía la búsqueda por el espacio en el proceso genético de multisimplificación, hace uso de esta métrica para modificar el valor asociado al individuo por la función de adaptación $F(\cdot)$ empleada en el algoritmo básico de simplificación. Para ello, multiplica este valor por una *función de penalización* que reduce la recompensa en el caso en el que el cromosoma esté situado en un entorno cercano a una solución obtenida anteriormente. Así, la expresión de la función de adaptación modificada empleada por el proceso de multisimplificación es la siguiente:

$$F'(C_j) = F(C_j) \cdot G(C_j, S)$$

donde $S = \{s_1, \dots, s_k\}$ es el conjunto que contiene las distintas soluciones (bases de reglas difusas) generadas hasta el momento y G es un tipo de función de penalización. Puesto que el problema con el que trabajamos es un problema de minimización, la función de penalización que utilizaremos será la siguiente:

$$G(C_j, S) = \begin{cases} \infty, & \text{si } d = 0 \\ 2 - \left(\frac{d}{r}\right)^\beta, & \text{si } d < r \text{ y } d \neq 0 \\ 1, & \text{si } d \geq r \end{cases}$$

donde d es el valor de la distancia Hamming existente entre C_j y la solución $s_i \in S$ más cercana, es decir, $d = \text{Min}_i\{H(C_j, s_i)\}$, r es el *radio del nicho* y β es el *factor de potencia*, que determina cómo de cóncava ($\beta > 1$) o convexa ($\beta < 1$) es la curva de penalización. De este modo, la penalización asociada por la función $G(\cdot)$ es máxima cuando la definición de la base de reglas difusas codificada en el cromosoma C_j ya haya sido obtenida previamente. Por otro lado, no se penaliza en ningún grado al individuo cuando la solución que codifica está situada a una distancia mayor o igual que el radio de nicho r de las soluciones contenidas en S .

Para finalizar esta sección, presentamos el algoritmo del proceso genético de multisimplificación:

1. *Inicialización: Asignar la función de adaptación empleada en el proceso básico de simplificación a la función de adaptación modificada empleada en el proceso de multisimplificación: $F'(C_j) \leftarrow F(C_j)$.*
2. *Ejecutar el proceso básico de simplificación y hacer que emplee la función de adaptación modificada, almacenando el mejor individuo obtenido en esa ejecución.*
3. *Actualizar la estructura de la función de adaptación modificada de tal modo que penalice la zona del espacio solución cercana a la última solución encontrada, generando así una nueva función de adaptación modificada.*
4. *Si se han obtenido todas las definiciones de la base de reglas difusas deseadas, terminar la ejecución del algoritmo. En otro caso, volver al paso 2.*

Por ello, el número de ejecuciones del algoritmo secuencial a realizar es igual al número de definiciones distintas de la base de reglas difusas que deseemos generar. El diseñador del sistema es el encargado de determinar este valor, así como el de los parámetros r y β . Para facilitar la elección del valor de r , se especificará como un tanto por ciento del número de reglas que presenta el conjunto generado en la fase anterior, B^g .

3.4 El Proceso Evolutivo de Ajuste

Finalmente, en lo que respecta a la tercera etapa del aprendizaje, en la metodología evolutiva propuesta se introducen una serie de recomendaciones que se pueden enmarcar en dos grupos, el dedicado a la definición del modo en que se ajustan las funciones de pertenencia consideradas en las reglas difusas y los parámetros asociados a los consecuentes de las reglas difusas TSK; y el que especifica la forma de la función de adaptación a emplear. Analizaremos estos dos factores en las dos secciones siguientes.

3.4.1 Ajuste de Funciones de Pertenencia y de Consecuentes TSK

Como comentábamos en la sección 3.1.3, en la metodología evolutiva de aprendizaje propuesta se considera la posibilidad de efectuar dos tipos distintos de ajuste de las funciones de pertenencia sobre la definición inicial de una base de reglas difusas, dependiendo de la naturaleza de la misma. En el caso de SBRDs descriptivos de tipo Mamdani o de tipo TSK, el ajuste se efectúa a nivel global sobre las particiones difusas asociadas a cada variable lingüística, mientras que en el caso de los SBRDs aproximativos, dicho ajuste se produce a nivel local sobre las funciones de pertenencia individuales empleadas en cada regla difusa de la base.

En ambos casos el esquema de codificación a considerar es el mismo, un esquema de codificación real, y el esquema de representación es muy similar. En la metodología se considera el empleo de funciones de pertenencia lineales diferenciables por partes (por ejemplo, triangulares y trapezoidales) puesto que, como se indicaba en el primer capítulo de esta memoria, se ha demostrado que este tipo de funciones son capaces de aproximar adecuadamente a las de los demás tipos, con la ventaja de su mayor simplicidad. De este modo, cada función de pertenencia individual puede ser representada mediante una tupla formada por tres o cuatro valores reales, según sea triangular o trapezoidal. Los individuos se obtienen concatenando las representaciones individuales de los conjuntos difusos en cuestión, los que forman las particiones difusas iniciales en el caso de los sistemas descriptivos o TSK, o los empleados en las reglas en el caso de los aproximativos. En el caso de los SBRDs de tipo TSK, los individuos incluyen también una parte en la que se codifican los valores de los parámetros que definen los consecuentes de las reglas, para

el refinamiento de la definición inicial de éstos por parte del AE. Desarrollaremos más detenidamente estos conceptos en los capítulos 4, 5 y 6 de esta memoria, en los que presentaremos varios AEs concretos para el ajuste de las funciones de pertenencia de bases de reglas difusas de distintos tipos.

En esta sección, nos centraremos en la especificación del modo en el que se emplean las definiciones iniciales de las funciones de pertenencia consideradas para definir los intervalos de variación de los parámetros asociados, con objeto de llevar a cabo su adaptación en el proceso evolutivo. Es obvio que dichos intervalos han de estar bien definidos para que los cambios efectuados por los operadores evolutivos sobre la representación considerada no provoquen la aparición de conjuntos difusos incoherentes. Así, dada una definición inicial de una función de pertenencia triangular concreta representada por una tupla con tres valores reales, (a, b, c) , en la metodología evolutiva presentada se considerarán dos posibilidades para especificar los intervalos de variación asociados a cada uno de esos tres parámetros:

- En primer lugar, se puede asociar un *intervalo fijo de variación* a cada uno de ellos, que dependa de los valores iniciales de sus vecinos, del siguiente modo:

$$\begin{aligned} a &\in [a^i, a^d] = [a - \frac{b-a}{2}, a + \frac{b-a}{2}] \\ b &\in [b^i, b^d] = [b - \frac{b-a}{2}, b + \frac{c-b}{2}] \\ c &\in [c^i, c^d] = [c - \frac{c-b}{2}, c + \frac{c-b}{2}] \end{aligned}$$

La figura 3.4 muestra la representación gráfica de estos intervalos, los cuales permanecen constantes a lo largo de la ejecución del proceso evolutivo de ajuste.

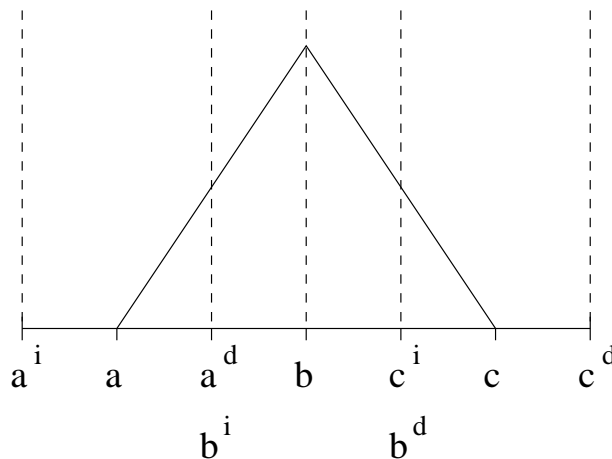


Figura 3.4. Definición de los intervalos de variación fijos

- La otra posibilidad consiste en considerar *intervalos de variación variables* asociados a cada uno de los tres parámetros. Para ello, se asocia primero un intervalo de

variación global a cada conjunto difuso, especificado a partir de la definición inicial del mismo considerada en las fases anteriores del aprendizaje, de la siguiente forma:

$$[I, D] = \left[a - \frac{b-a}{2}, c + \frac{c-b}{2} \right]$$

Este es el único intervalo de definición que permanece constante a lo largo de la ejecución del AE en la segunda forma de definición. Los intervalos asociados a los tres parámetros a , b y c varían en cada caso, dependiendo del individuo concreto y del valor de sus vecinos del modo que se muestra a continuación:

$$a \in [a^i, a^d] = [I, b], \quad b \in [b^i, b^d] = [a, c], \quad c \in [c^i, c^d] = [b, D]$$

La figura 3.5 muestra la representación gráfica de los valores iniciales de estos intervalos.

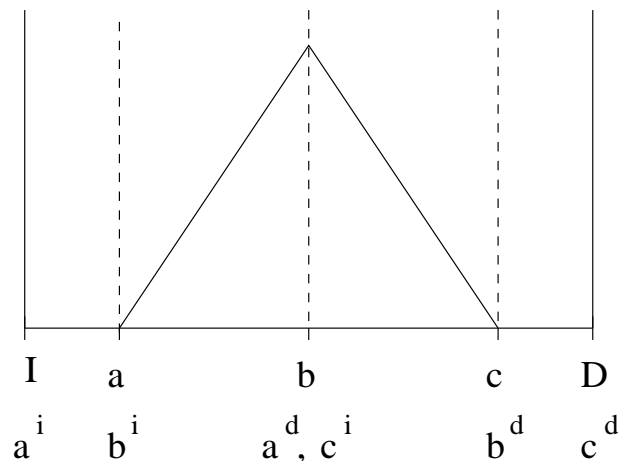


Figura 3.5. Definición inicial de los intervalos de variación variables

Como puede observarse claramente, la elección de la segunda posibilidad repercute en un grado de libertad mayor en el ajuste de las funciones de pertenencia, dando lugar a que éstas puedan presentar finalmente cualquier forma coherente dentro del intervalo global $[I, D]$.

Finalmente, en lo referente a la adaptación de las definiciones iniciales de los parámetros asociados a los consecuentes de las reglas difusas TSK por parte del proceso evolutivo de ajuste, no es necesario especificar ningún tipo de intervalo de variación, ya que la codificación angular permite llevar a cabo el refinamiento de los valores sin necesidad de definir éstos.

3.4.2 Definición de la Función de Adaptación

La función de adaptación a emplear en la tercera fase del aprendizaje dependerá de la composición global del SBRDE en varios niveles con el que se esté trabajando. Como hemos indicado a lo largo de este capítulo, según que el SBRD que se pretenda diseñar sea de tipo Mamdani o de tipo TSK, existen dos posibilidades.

En el primero de los casos, el SBRDE empleado estará formado por tres etapas: el proceso de generación de reglas difusas, el proceso genético de multisimplificación y el proceso evolutivo de ajuste. En este caso, las definiciones iniciales de las bases de reglas difusas sobre las que trabaja este último proceso para refinar la forma de las funciones de pertenencia provienen del proceso genético de multisimplificación, por lo que verifican la propiedad de ser completas a grado τ . De este modo, tiene sentido que la función de adaptación a emplear sea similar a la considerada en la segunda etapa del aprendizaje, es decir, que esté compuesta por un criterio que penalice la no satisfacción de la propiedad de completitud por parte de la base de reglas difusas que emplee la definición de las funciones de pertenencia codificadas en el individuo y por una medida de error que permita evaluar el comportamiento que presenta dicha base, en definitiva, la bondad de la cooperación existente entre las reglas que la componen. Con ello, la expresión genérica de dicha función será la siguiente:

$$F(C_j) = \begin{cases} ME(C_j), & \text{si } GCCE(R(C_j), E_p) \geq \tau \\ \infty, & \text{en otro caso} \end{cases}$$

donde $ME(C_j)$ es el valor obtenido por el SBRD que emplea la definición de la base de reglas difusas codificada en el individuo C_j en la medida de error concreta considerada; $GCCE(R(C_j), E_p)$ es el grado de completitud sobre el conjunto de entrenamiento de dicha definición, presentado en la sección 3.3.1.

Por otro lado, en el segundo caso, el SBRDE considerado está compuesto solamente por dos etapas: el proceso evolutivo de generación y el de refinamiento. Así, las definiciones iniciales con las que trabaja el segundo proceso no verifican la propiedad anterior, debido a la composición de las reglas difusas de tipo TSK. Por lo tanto, no es necesario considerar ningún tipo de criterio al respecto en la función de adaptación. En este tipo de SBRDEs construidos a partir de la metodología propuesta, dicha función estará compuesta únicamente por una medida de error global, que evalúe el comportamiento del SBRD que emplea la definición de las funciones de pertenencia y de los parámetros de los consecuentes, codificada en el individuo.

Las posibilidades para la elección de la medida de error a emplear en cualquiera de los dos casos son muy variadas. Todas las medidas incluidas en la familia de las medidas de error están basadas en el cálculo de la diferencia existente entre la salida que devuelve en la práctica el SBRD al recibir una entrada concreta y la salida teórica que debería proporcionar al recibir dicha entrada. Cuanto mayor sea el valor que devuelva la medida, peor será el rendimiento del SBRD, por lo que este tipo de medidas deben ser consideradas

como criterios a minimizar en el proceso de aprendizaje evolutivo.

La diferencia entre el valor teórico, y , y el devuelto por el sistema, y' , se obtiene haciendo uso de cualquier tipo de métrica. Así, todas las medidas de error son, en definitiva, modificaciones de las dos formas clásicas de calcular la diferencia entre dos valores, los *errores lineal* (EL) y *cuadrático* (EC), cuyas expresiones matemáticas se muestran a continuación:

$$EL = |y - y'| \quad ; \quad EC = \frac{1}{2}(y - y')^2$$

Estas dos medidas pueden ser generalizadas para calcular la distancia entre dos conjuntos de datos: uno formado por los valores teóricos y otro por las salidas devueltas por el SBRD ante los valores de las entradas contenidas en el primero, en lugar de entre un único par de datos simples. Esta tarea se puede llevar a cabo de dos formas distintas, ambas basadas en el cálculo de un valor de la medida para cada par de datos de ambos conjuntos y la caracterización de todos los valores intermedios así obtenidos por medio de un único valor asociado a la medida generalizada. La diferencia entre ambas formas de trabajo radica en el modo en el que se obtiene dicho valor: bien seleccionando uno de los valores intermedios por medio de algún criterio, o bien agregando todos ellos en un nuevo valor, que no tiene por qué pertenecer al conjunto inicial de valores intermedios, mediante la suma o la media de éstos.

El criterio empleado habitualmente para la primera forma de generalización se basa en la selección del máximo del conjunto de valores intermedios. Las medidas de error que generalizan a las dos anteriores procediendo de esta forma reciben, respectivamente, el nombre de *error lineal máximo* (ELMax) y *error cuadrático máximo* (ECMax):

$$ELMax = \max_l |y_l - y'_l|, \quad l = 1, \dots, p$$

$$ECMax = \max_l \frac{1}{2}(y_l - y'_l)^2, \quad l = 1, \dots, p$$

donde p es el cardinal del conjunto de datos.

Por otro lado, como ya hemos indicado, el segundo modo de generalización presenta las dos posibilidades siguientes:

- Calcular directamente la suma de los valores intermedios, con lo que se obtienen las medidas *suma de los errores lineales* (SEL) y *suma de los errores cuadráticos* (SEC):

$$SEL = \sum_{l=1}^p |y_l - y'_l|$$

$$SEC = \sum_{l=1}^p \frac{1}{2}(y_l - y'_l)^2$$

- Calcular la media de dichos valores, lo que provoca la creación de las medidas *error lineal medio* (ELM) y *error cuadrático medio* (ECM):

$$ELM = \frac{\sum_{i=1}^p |y_i - y'_i|}{p}$$

$$ECM = \frac{\frac{1}{2} \sum_{i=1}^p (y_i - y'_i)^2}{p}$$

Como puede observarse, el empleo de este tipo de medidas en cualquier proceso de aprendizaje da lugar a que dicho proceso esté condicionado en gran medida por la naturaleza del conjunto de entrenamiento empleado. El tamaño y la composición de este conjunto determinarán la bondad del sistema de aprendizaje obtenido finalmente, en un grado que dependerá de la complejidad del problema afrontado. Cuanto más complejo sea éste, más datos serán necesarios para que el proceso de aprendizaje se comporte de una forma adecuada.

En la metodología evolutiva de aprendizaje de bases de reglas difusas propuesta, se permite el empleo de cualquiera de las medidas de error presentadas para el diseño de las funciones de adaptación asociadas a la segunda y tercera fases del aprendizaje (el proceso genético de multisimplificación y el proceso evolutivo de ajuste, respectivamente), en el caso de SBRDs de tipo Mamdani, o de la segunda (el proceso evolutivo de refinamiento), en el caso de sistemas TSK.

En los distintos SBRDEs que presentaremos en los capítulos 4, 5 y 6, haremos uso de la medida de ECM para construir las funciones de adaptación empleadas en todos los procesos evolutivos de ajuste y refinamiento, por lo que las asociadas a los procesos evolutivos de ajuste de bases de reglas difusas de tipo Mamdani coincidirán con la considerada en el proceso genético de multisimplificación.

3.5 Análisis de los Parámetros Empleados

Finalizaremos el capítulo actual llevando a cabo un análisis del comportamiento que presentarán los SBRDEs basados en la metodología propuesta, dependiendo del valor que tomen los distintos parámetros que es necesario especificar para su funcionamiento. Además de los parámetros asociados a los AEs empleados en las distintas fases del proceso evolutivo de aprendizaje, los cuatro parámetros fundamentales cuyos valores condicionan la composición y el comportamiento del SBRD diseñado son los siguientes:

1. El parámetro k , asociado a la propiedad de k -consistencia. Este parámetro está definido en $[0, 1]$ y es considerado en el proceso de generación de reglas difusas.

2. El parámetro ω , considerado en el proceso de generación para el cálculo del conjunto de ejemplos positivos de una regla a un grado concreto, $E_{\omega}^{+}(R_i)$. Está definido en $(0, 1]$.
3. El parámetro $\epsilon \in \mathbf{R}^{+}$, que especifica el umbral de cubrimiento necesario para que un ejemplo sea eliminado del conjunto de entrenamiento en el proceso de generación de reglas.
4. El parámetro $\tau \in (0, 1]$, el cual especifica el mínimo grado de cubrimiento permitido para que una base de reglas difusas de tipo Mamdani se considere completa en los procesos evolutivos de multisimplificación y ajuste.

Como podremos comprobar en los experimentos recogidos en la sección 4.5.1 del capítulo siguiente, el valor de los dos primeros parámetros, k y ω , no afecta en exceso a la composición final de la base de reglas difusas aprendida, obteniéndose en algunos casos los mismos resultados aun empleando distintos valores para ellos. En cambio, los dos parámetros restantes, ϵ y τ , sí influyen de un modo significativo en la composición de ésta, por lo que es recomendable estudiar cuáles son los valores que han de tomar, para conseguir que la base aprendida presente el comportamiento deseado.

Ambos parámetros condicionan el número de reglas que tendrá dicha base de reglas difusas y, por tanto, el rendimiento del SBRD diseñado finalmente. Puesto que el parámetro ϵ determina el grado de cubrimiento que es necesario alcanzar para eliminar un ejemplo del conjunto de entrenamiento en el proceso de generación de reglas difusas, cuanto mayor sea el valor del mismo, más reglas será necesario generar para cubrir cada ejemplo y, por tanto, el conjunto de reglas difusas, B^g , obtenido como resultado de la ejecución de este proceso presentará un número mayor de reglas.

Esto lleva asociados una ventaja y un inconveniente. La ventaja proviene del hecho de que cuantas más reglas existan en B^g , mejor comportamiento podrán presentar los SBRDEs que manejen las distintas definiciones de bases de reglas difusas obtenidas como salida del proceso genético de multisimplificación. Esto se debe a que, al existir más reglas en el conjunto inicial, hay un número mucho mayor de combinaciones posibles de las mismas para seleccionar el subconjunto que presente mejor nivel de cooperación, por lo que el proceso de multisimplificación puede obtener mejores resultados. Es en este segundo proceso, así como en el tercero, en los que más influye el valor del parámetro τ . Dicho valor condicionará también el número de reglas asociado a las distintas bases generadas ya que, al determinar el grado de completitud exigido a las mismas, un valor alto hace que sean necesarias más reglas para satisfacer dicha condición.

En lo que respecta al inconveniente que destacábamos, éste es debido al número de reglas que formarán la base de reglas difusas final, cuando escojamos un valor alto para ϵ . Al ser alto este valor, el conjunto inicial de reglas difusas B^g tendrá un número elevado de reglas, lo que hará que la ejecución de los procesos de multisimplificación y ajuste conlleven un tiempo de máquina superior; y lo más importante, que la base aprendida

presente también, finalmente, un número de reglas alto, lo cual puede ser un inconveniente en problemas en los que se desee una respuesta rápida por parte del SBRD como, por ejemplo, en problemas de control en tiempo real.

De este modo, el control de los valores de los parámetros ϵ y τ permite al diseñador del SBRD el generar una base de reglas difusas que proporcione el comportamiento deseado en el ratio rendimiento-número de reglas. A la vista de lo comentado en los párrafos anteriores, parece adecuado el escoger siempre un valor bajo para el parámetro τ ya que, de ese modo, se consigue que el SBRDE simplifique al máximo posible la base de reglas difusas generada en la primera etapa del aprendizaje. Este hecho es positivo para ambos criterios ya que dota de una mayor libertad al proceso de multisimplificación, lo que repercute en la obtención de bases de reglas difusas con menos reglas (menor complejidad) y con mejor nivel de cooperación entre éstas (en definitiva, un SBRD con mejor rendimiento).

Por otro lado, puesto que el parámetro ϵ es el encargado de determinar el número de reglas que compondrán la base de reglas difusas inicial, su elección dependerá directamente del comportamiento que se desee que presente el sistema:

- Cuando el objetivo a conseguir sea el obtener el mayor nivel de precisión posible, el diseñador deberá fijar un valor alto para el parámetro ϵ . Esto hará que el SBRD diseñado presente un buen rendimiento y que su base de reglas difusas esté formada por un número elevado de reglas.
- En cambio, cuando el criterio principal a satisfacer sea el generar una base con un nivel de complejidad bajo, la elección debe ser inversa, un valor bajo para el parámetro ϵ . De este modo, la base aprendida contendrá menos reglas, aunque esto redundará en una pérdida de precisión por parte del sistema.

En la sección 4.5.1 del cuarto capítulo de esta memoria están recogidos una serie de experimentos con distintas combinaciones de valores para los parámetros ϵ y τ , que reafirmarán en la práctica las afirmaciones efectuadas en la teoría.

Capítulo 4

Un Sistema Basado en Reglas Difusas Evolutivo para el Aprendizaje de Bases de Conocimiento Descriptivas de tipo Mamdani

Como comentábamos en el primer capítulo, la principal aplicación de los SBRDs descriptivos de tipo Mamdani es el modelado cualitativo de sistemas. Los SBRDs dedicados a esta tarea tienen como característica predominante la interpretabilidad existente en el modelo obtenido pero, desafortunadamente, la estructura de regla difusa que manejan, basada en el empleo de variables lingüísticas, provoca que la precisión de éste quede, en algunas ocasiones, por debajo de lo deseado. En estos casos, una posible solución consiste en cambiar la estructura del sistema y decidirse por un modelo difuso aproximativo, con lo que se gana en precisión en la aproximación pero se pierde en poder de descripción, característica fundamental que se desea mantener.

A la vista del problema, sería deseable el poder obtener un modelo cualitativo que, manteniendo al máximo posible su nivel de interpretabilidad, presente una mayor precisión.

En [NIT97], los autores tratan de poner esta idea en práctica diseñando en primer lugar un modelo difuso aproximativo basado en reglas de tipo TSK simplificadas y transformándolo posteriormente en un modelo cualitativo que presente el mismo rendimiento. Para ello, introducen una BC secundaria, además de la BC habitual, y presentan un Sistema de Inferencia capaz de obtener un resultado combinado de la actuación de ambas bases de reglas difusas. Lo que realmente hace el sistema es permitir que una combinación de antecedentes determinada tenga asociados dos consecuentes distintos, el primero y el segundo en importancia, con lo que se evitan algunos de los problemas asociados a la estructura de las reglas lingüísticas. Por un lado, se logra que el número de éstas no venga delimitado por las particiones difusas asociadas a las variables de entrada y, por

otro, se evita la falta de flexibilidad derivada del uso de este tipo de particiones (véase la sección 1.1.3 en el primer capítulo de esta memoria). El hecho de que el resultado de la inferencia sobre las dos reglas resultantes sea una interpolación de la salida individual de ambas permite obtener un rendimiento más preciso, sin perder la interpretabilidad del sistema.

En este cuarto capítulo vamos a presentar un SBRDE basado en la metodología evolutiva de aprendizaje para obtener SBRDs descriptivos de tipo Mamdani que extienden la forma de trabajo comentada en el párrafo anterior. Dicho SBRDE posibilitará el aprendizaje automático de BCs en las que podrán coexistir tantas reglas lingüísticas con la misma combinación de antecedentes como fuera necesario para que el comportamiento global del sistema sea el deseado. Además, no sólo será posible que dos reglas que presenten la misma combinación de antecedentes tengan asociado un valor de consecuente distinto, sino también que el valor de éste sea el mismo (es decir, que la misma regla lingüística esté contenida dos o más veces en la base), lo que equivaldrá a dar una mayor importancia a dicha regla.

Una BC del tipo comentado puede ser considerada como un conjunto de reglas lingüísticas con pesos enteros asociados, aprendidos de un modo automático por el SBRDE, que determinan la importancia de cada una de ellas. El empleo de una BC formada por reglas lingüísticas con múltiples consecuentes da lugar a que el modelo sea mucho más preciso, más exacto en la aproximación del sistema real, al mismo tiempo que puede mantener un grado de interpretabilidad aceptable.

La organización de este capítulo es la que indicamos a continuación. En primer lugar, analizaremos la estructura que deben presentar los SBRDs descriptivos de tipo Mamdani para poder manejar una BC del tipo comentado. Posteriormente, introduciremos la estructura concreta en tres niveles del SBRDE propuesto, basada en las recomendaciones analizadas en el capítulo anterior, y dedicaremos dos secciones a describir los procesos individuales que componen la primera y la tercera etapa del mismo, los procesos de generación de reglas lingüísticas y de ajuste, respectivamente. A continuación, estudiaremos el comportamiento del proceso evolutivo de aprendizaje en el modelado cualitativo de las cuatro funciones tridimensionales presentadas en el Apéndice A, realizando un breve estudio práctico de la influencia de los distintos parámetros considerados. Finalmente, compararemos el comportamiento de los modelos obtenidos con el de otros generados a partir de otros dos SBRDEs con distintas características.

4.1 Estructura del Sistema Basado en Reglas Difusas Considerado

Tal y como hemos indicado, el SBRDE propuesto trabaja con SBRDs descriptivos de tipo Mamdani con múltiples entradas y una única salida. La estructura general de la BC de un sistema de este tipo se modifica únicamente para permitir que contenga distintas reglas lingüísticas con la misma combinación de antecedentes y un valor igual o distinto de consecuente. Por lo demás, la estructura general de regla descriptiva es la habitual, introducida en el capítulo 1:

$$R_i : \text{Si } X_1 \text{ es } A_{i1} \text{ y } \dots \text{ y } X_n \text{ es } A_{in} \text{ entonces } Y \text{ es } B_i,$$

donde X_i e Y son variables lingüísticas de entrada y salida respectivamente, y los A_{ij} y B_i son etiquetas lingüísticas asociadas a conjuntos difusos que determinan su semántica. Trabajaremos con funciones de pertenencia triangulares, aunque podríamos emplear cualquier otro tipo.

Para poder aprovechar la potencialidad de una BC con esta estructura, es necesario emplear un Interfaz de Defuzzificación que trabaje en modo B, es decir, que primero defuzzifique los conjuntos difusos individuales obtenidos a partir del proceso de inferencia y después agregue los valores característicos así generados mediante algún tipo de operador de agregación que potencie la presencia del mismo valor varias veces. Para llevar a cabo esta tarea, puede emplearse cualquier operador basado en una media, ponderada o no. En nuestro caso, haremos uso de la *media ponderada por el grado de emparejamiento*, el cual combinaremos con el defuzzificador *Centro de Gravedad* [CHP97, HT93, SY93].

En lo que respecta a los restantes parámetros de diseño del mecanismo de inferencia del SBRD, emplearemos la t-norma del Mínimo como operador de conjunción e implicación. De este modo, el mecanismo de inferencia diseñado procederá del siguiente modo ante una entrada $x_0 = (x_1, \dots, x_n)$:

1. Para cada regla R_i , $i = 1, \dots, T$, de la BC:

- (a) Calcula el grado de emparejamiento, h_i , de la regla:

$$h_i = \text{Min}(\mu_{A_{i1}}(x_1), \dots, \mu_{A_{in}}(x_n))$$

- (b) Aplica la t-norma del Mínimo actuando como operador de implicación para obtener el conjunto difuso resultante del proceso de inferencia sobre la regla en cuestión, B'_i :

$$\mu_{B'_i}(y) = \text{Min}(h_i, \mu_{B_i}(y))$$

2. Obtiene el Centro de Gravedad de cada conjunto difuso individual B'_i :

$$y_i = \frac{\int_{\mathbf{V}} y \cdot \mu_{B'_i}(y) \cdot dy}{\int_{\mathbf{V}} \mu_{B'_i}(y) \cdot dy}$$

3. Calcula la salida final devuelta por el sistema, y_0 , agregando las acciones parciales obtenidas, mediante la media ponderada por el grado de emparejamiento:

$$y_0 = \frac{\sum_{i=1}^T h_i \cdot y_i}{\sum_{i=1}^T h_i}$$

4.2 Estructura del Sistema Basado en Reglas Difusas Evolutivo en Varias Etapas

El SBRDE para el aprendizaje automático de BCs descriptivas de tipo Mamdani presentado en el capítulo actual plasma los conceptos generales de la metodología evolutiva de aprendizaje propuesta, mediante una estructura compuesta por los siguientes tres procesos:

1. *Un proceso iterativo de generación de reglas lingüísticas*, compuesto por un *método de generación de reglas lingüísticas*, basado en un algoritmo inductivo y en una medida frecuentista de la bondad de las reglas, y el *método iterativo de cubrimiento* presentado en el capítulo anterior. Este primer proceso permite obtener un conjunto de reglas lingüísticas que cubra el conjunto de entrenamiento de un modo adecuado.
2. *El proceso genético de multisimplificación* presentado en el capítulo 3 para llevar a cabo la selección de reglas. Como vimos en dicho capítulo, este proceso se implementa por medio de un AG con codificación binaria que emplea una función de compartición genotípica y una medida del rendimiento del SBRD codificado en el cromosoma. Además, permite obtener varias definiciones de la BRL formadas por las distintas combinaciones de las reglas lingüísticas generadas en la fase anterior que mejor nivel de cooperación presentan.
3. *Un proceso genético de ajuste* basado en un AG con codificación real y en una función de adaptación que considera dos criterios: una medida del rendimiento del SBRD codificado en el cromosoma y una función que penaliza la no satisfacción de la propiedad de completitud. Este último proceso proporciona la BC final, ajustando la definición previa de la forma de las funciones de pertenencia que especifican la semántica de los términos lingüísticos existentes en la BD. La BC que presente el mejor comportamiento tras el ajuste, de entre todas las generadas en el segundo proceso, será la que constituirá la salida global del SBRDE.

4.3 El Proceso Iterativo de Generación de Reglas Lingüísticas

Como hemos indicado, el proceso iterativo de generación de reglas lingüísticas está basado en el empleo del método iterativo de cubrimiento presentado en la sección 3.2.2 del capítulo anterior y de un método inductivo de generación de reglas lingüísticas, encargado de obtener la mejor regla en cada momento, con respecto al estado actual del conjunto de entrenamiento. Para ello, este último método hace uso de los ejemplos no cubiertos existentes en dicho conjunto en un instante determinado, para generar un conjunto de reglas candidatas, y escoge la que presente mejor valor en una función de selección de reglas basada en una serie de criterios frecuentistas.

El método de generación de reglas lingüísticas se basa en la existencia de una definición previa de la BD que contenga los términos lingüísticos asociados a las variables existentes en el problema, así como de las funciones de pertenencia de los conjuntos difusos que especifican la semántica de éstos. Esta definición puede ser obtenida a partir del conocimiento experto disponible, o bien mediante una equipartición difusa de los espacios de entrada y salida considerando un número fijo de términos.

Cada vez que se ejecuta, el método de generación construye un conjunto de reglas lingüísticas candidatas a partir de la regla que mejor cubre a cada ejemplo contenido en el conjunto de entrenamiento. La estructura de estas reglas se obtiene tomando un ejemplo concreto, un vector de valores reales con dimensión $n + 1$ (n valores de entrada y uno de salida), y asociando a cada variable lingüística la etiqueta que mejor se empareje con la componente correspondiente del ejemplo. Esta etiqueta será aquella de entre todas las existentes en el conjunto de términos de la variable, que tenga asociado el conjunto difuso para el que la componente del vector presente un mayor grado de pertenencia.

Una vez obtenido el conjunto de reglas candidatas, se evalúa el rendimiento de las distintas reglas lingüísticas que lo componen, empleando una función multicriterio de selección de reglas. Los criterios que componen dicha función dan lugar a que las reglas con mejores valores en la misma sean prometedoras para formar parte de una BC con buen comportamiento, además de verificar las propiedades de completitud, cubrimiento y consistencia. Los criterios considerados son los que se muestran a continuación. Sus expresiones matemáticas pueden ser consultadas en la sección 3.2.1.1 del capítulo 3.

- Alto valor de frecuencia: $\Psi_{E_p}(R_i)$.
- Alto grado de cubrimiento medio sobre los ejemplos positivos: $G_\omega(R_i)$.
- Penalización debida a la no satisfacción de la propiedad de k -consistencia: $g_n^-(R_i)$.

Estos criterios se combinan en una función de adaptación, mediante el uso de una función de agregación creciente en las tres variables. En este caso, usaremos el producto, por lo que la expresión final de la función de selección de reglas es la siguiente:

$$F(R_i) = \Psi_{E_p}(R_i) \cdot G_\omega(R_i) \cdot g_n^-(R_i)$$

Cuanto mayor sea el valor obtenido por una regla lingüística en esta función, mejor considerada estará dicha regla en el proceso de selección.

4.4 El Proceso Genético de Ajuste de Bases de Conocimiento Descriptivas

El proceso genético de ajuste empleado como tercera y última fase de nuestro SBRDE es una variante del presentado en [HLV95]. Dicho proceso, introducido brevemente en la sección 2.4.1.2 del capítulo 2 de esta memoria, está basado en la existencia de una definición previa de una BC o una BRD y emplea un AG para trabajar sobre una codificación de las funciones de pertenencia utilizadas en cada regla difusa individual, ajustando los parámetros de las mismas de un modo local para cada conjunto difuso. De este modo, se obtiene un SBRD aproximativo, aun en el caso de que el sistema inicial fuese descriptivo.

Una vez que se han ejecutado las dos fases iniciales del SBRDE propuesto, ya se poseen varias definiciones distintas de la BC, formadas por la definición inicial de la BD y las BRLs aprendidas tras la aplicación de los procesos de generación y multisimplificación. En ese momento, podemos emplear un método de ajuste similar al comentado, para refinar la definición de las funciones de pertenencia empleadas en cada BC y obtener SBRDs con mejor rendimiento. Como pretendemos que el SBRDE se aplique en tareas de modelado cualitativo, es necesario que este proceso genético de ajuste mantenga la naturaleza descriptiva de la BC, así como que genere una definición interpretable de los conjuntos difusos que especifican la semántica de los términos lingüísticos empleados.

Tomando como base estos objetivos, mantendremos la estructura general del proceso genético de ajuste comentado, pero cambiaremos el esquema de representación empleado para que el ajuste de la definición de las funciones de pertenencia efectuado por el AG se realice a nivel global de BC y no a nivel individual de regla. Con ello, lo que codificaremos en cada cromosoma será una definición distinta de los conjuntos difusos primarios que componen las particiones difusas asociadas a cada variable lingüística existente en la base.

Emplearemos la representación paramétrica de las funciones de pertenencia triangulares consideradas, en forma de tupla con tres valores reales y representaremos cada partición difusa primaria por medio de un vector de valores reales de dimensión $3 \cdot N$, donde N es el número de conjuntos difusos existentes en la partición difusa en cuestión, es decir, el número de términos lingüísticos que puede tomar como valor la variable lingüística. Así, la definición de todas las funciones de pertenencia existentes en la BD en un problema que

maneje m variables lingüísticas puede codificarse en un cromosoma C_r con codificación real y de longitud fija $\sum_{i=1}^m 3 \cdot N_i$, formado por la concatenación de las representaciones parciales de las m particiones difusas:

$$\begin{aligned} C_{ri} &= (a_{i1}, b_{i1}, c_{i1}, \dots, a_{iN_i}, b_{iN_i}, c_{iN_i}) \\ C_r &= C_{r1} C_{r2} \dots C_{rm} \end{aligned}$$

La generación de la población inicial de cromosomas que manejará el algoritmo se realiza haciendo uso de la definición inicial de la que se dispone, tal y como recomienda la metodología evolutiva de aprendizaje presentada. De este modo, se codifica la definición del conjunto de funciones de pertenencia empleada en las dos etapas anteriores del SBRDE en un cromosoma C_1 y se generan los individuos restantes inicializando aleatoriamente cada uno de los genes que las componen, c_h , $h = 1, \dots, \sum_{i=1}^m 3 \cdot N_i$, en su intervalo de variación asociado, $[c_h^i, c_h^d]$. Estos intervalos de variación se calculan considerando la primera posibilidad presentada en la sección 3.4.1 del capítulo anterior, es decir, serán intervalos fijos que dependerán de la forma de los distintos conjuntos difusos existentes en las particiones difusas iniciales. De este modo, se asegura que las definiciones de los distintos conjuntos difusos obtenidas después del ajuste mantengan su interpretabilidad, ya que se imposibilitan acciones tales como que el punto modal del conjunto difuso asociado a una etiqueta de menor orden supere al de otra de orden superior.

De este modo, si tomamos como base el cromosoma que codifica la definición original de las funciones de pertenencia, C_1 , y $(t \bmod 3) = 1$, entonces c_t es el valor situado más a la izquierda del soporte del número difuso. Dicho número se define mediante tres parámetros (c_t, c_{t+1}, c_{t+2}) y los intervalos de variación asociados a cada uno de ellos son los siguientes:

$$\begin{aligned} c_t \in [c_t^i, c_t^d] &= [c_t - \frac{c_{t+1} - c_t}{2}, c_t + \frac{c_{t+1} - c_t}{2}] \\ c_{t+1} \in [c_{t+1}^i, c_{t+1}^d] &= [c_{t+1} - \frac{c_{t+1} - c_t}{2}, c_{t+1} + \frac{c_{t+2} - c_{t+1}}{2}] \\ c_{t+2} \in [c_{t+2}^i, c_{t+2}^d] &= [c_{t+2} - \frac{c_{t+2} - c_{t+1}}{2}, c_{t+2} + \frac{c_{t+2} - c_{t+1}}{2}] \end{aligned}$$

El último cambio a efectuar se refiere a la función de adaptación. Siguiendo las pautas marcadas por la metodología evolutiva de aprendizaje propuesta, emplearemos una función constituida por dos criterios, la medida de error cuadrático medio (ECM) y un criterio que penaliza la no verificación de la propiedad de completitud, en lugar de la empleada en [HLV95], que únicamente consideraba la medida de ECM para la definición de ésta. La expresión de la función de adaptación empleada es la siguiente:

$$F(C_j) = \begin{cases} \frac{1}{2 \cdot |E_p|} \sum_{e_l \in E_p} (ey^l - S(ex^l))^2, & \text{si } GCC E(R(C_j), E_p) \geq \tau \\ \frac{1}{2} \sum_{e_l \in E_p} (ey^l)^2, & \text{en otro caso} \end{cases}$$

donde $GCC E(R(C_j), E_p) = \bigcap_{e_l \in E_p} C_{R(C_j)}(e_l)$ es el grado de completitud de la BC $R(C_j)$ sobre el conjunto de ejemplos E_p , como ya vimos en el capítulo anterior.

Por último, los aspectos restantes del AG presentado en [HLV95] se mantienen: la selección se efectúa mediante el mecanismo de muestreo universal estocástico considerando el modelo elitista, y los operadores de cruce y mutación empleados son respectivamente el cruce max-min-aritmético y la mutación no uniforme. Esta combinación de operadores genéticos permite llevar a cabo un proceso de búsqueda adecuado sobre un espacio en el que las soluciones posibles están definidas mediante parámetros reales.

4.5 Aplicación del Sistema Basado en Reglas Difusas Evolutivo Propuesto en el Modelado Cualitativo de Algunas Funciones Tridimensionales

En esta sección vamos a analizar el comportamiento del SBRDE presentado, aplicándolo al problema de la obtención de un modelado cualitativo de las cuatro funciones tridimensionales presentadas en el Apéndice A. En dicho Apéndice se recoge, además de las expresiones matemáticas y las representaciones gráficas de estas funciones, una descripción del modo en el que se han obtenido los distintos conjuntos de datos empleados como conjuntos de entrenamiento y prueba, así como la forma de las particiones difusas iniciales consideradas en los distintos experimentos realizados.

En primer lugar, presentaremos un pequeño estudio preliminar en el que aplicaremos el método evolutivo de aprendizaje propuesto al modelado cualitativo de dos de estas funciones, considerando distintas combinaciones de valores para los cuatro parámetros, ϵ , ω , k y τ , comentados en la sección 3.5 del capítulo anterior. Este estudio previo nos servirá para reafirmarnos en las afirmaciones efectuadas en dicha sección y para escoger los valores deseados para la experimentación completa a realizar sobre las cuatro funciones, que se mostrará en la sección siguiente. Finalmente, compararemos los resultados obtenidos por nuestro SBRDE con los presentados por otros dos procesos evolutivos de aprendizaje pertenecientes a la misma familia.

4.5.1 Experimentación Preliminar

Como ya hemos indicado, en esta sección mostraremos una serie de experimentos preliminares realizados para comprobar la influencia de los valores de los distintos parámetros asociados al SBRDE propuesto.

En este primer estudio trabajaremos solamente con una función sencilla y otra compleja, las notadas respectivamente como F_1 y F_4 en el Apéndice A. Hemos efectuado 18 ejecuciones del SBRDE propuesto con cada una de ellas, combinando los siguientes valo-

Tabla 4.1. Resultados obtenidos en los experimentos preliminares realizados para el modelado de la función F_1

| Parámetros | | | Generación | | | Simplificación | | | Ajuste | |
|------------|------------|--------|------------|-------------|-------------|----------------|-------------|-------------|-------------|-------------|
| ω | ϵ | τ | $\#R$ | EC_{entr} | EC_{prue} | $\#R$ | EC_{entr} | EC_{prue} | EC_{entr} | EC_{prue} |
| 0.05 | 0.5 | 0.1 | 73 | 4.80570 | 5.18393 | 57 | 2.55084 | 2.76559 | 0.36945 | 0.36680 |
| 0.05 | 0.5 | 0.25 | | | | 57 | 2.87154 | 2.92044 | 0.48768 | 0.36236 |
| 0.05 | 0.5 | 0.5 | | | | 63 | 4.20974 | 4.72377 | 0.44703 | 0.38259 |
| 0.05 | 1 | 0.1 | 118 | 4.39681 | 4.71096 | 62 | 2.06517 | 1.97865 | 0.41382 | 0.28957 |
| 0.05 | 1 | 0.25 | | | | 65 | 2.19964 | 2.06674 | 0.41420 | 0.33581 |
| 0.05 | 1 | 0.5 | | | | 74 | 2.56748 | 2.40234 | 0.49003 | 0.43852 |
| 0.05 | 1.5 | 0.1 | 167 | 3.51876 | 3.49868 | 90 | 1.45306 | 1.50833 | 0.28843 | 0.31545 |
| 0.05 | 1.5 | 0.25 | | | | 91 | 1.67093 | 1.62251 | 0.33891 | 0.43852 |
| 0.05 | 1.5 | 0.5 | | | | 97 | 1.70188 | 1.73554 | 0.33356 | 0.31328 |
| 0.5 | 0.5 | 0.1 | 73 | 4.80570 | 5.18393 | 57 | 2.55084 | 2.76559 | 0.36945 | 0.36680 |
| 0.5 | 0.5 | 0.25 | | | | 57 | 2.87154 | 2.92044 | 0.45987 | 0.36521 |
| 0.5 | 0.5 | 0.5 | | | | 63 | 4.20974 | 4.72377 | 0.37773 | 0.32123 |
| 0.5 | 1 | 0.1 | 122 | 4.27251 | 4.58297 | 62 | 2.02195 | 2.01614 | 0.44413 | 0.36273 |
| 0.5 | 1 | 0.25 | | | | 63 | 2.13095 | 2.12120 | 0.40254 | 0.34652 |
| 0.5 | 1 | 0.5 | | | | 76 | 2.46612 | 2.43469 | 0.55359 | 0.41878 |
| 0.5 | 1.5 | 0.1 | 167 | 3.51876 | 3.49869 | 88 | 1.42442 | 1.42804 | 0.31554 | 0.32427 |
| 0.5 | 1.5 | 0.25 | | | | 91 | 1.67093 | 1.62251 | 0.33891 | 0.43852 |
| 0.5 | 1.5 | 0.5 | | | | 97 | 1.70188 | 1.73554 | 0.33356 | 0.31328 |

res de parámetros: $\epsilon \in \{0.5, 1, 1.5\}$, $\omega \in \{0.05, 0.5\}$ y $\tau \in \{0.1, 0.25, 0.5\}$. En todos los experimentos, el valor de k se ha mantenido constante e igual a 0.1.

Los valores de los restantes parámetros del SBRDE, asociados a los AGs que componen la segunda y la tercera fase, son los siguientes:

- Tamaño de la población: $N = 61$, en ambos casos.
- Valores de las probabilidades de cruce y mutación: $P_c = 0.6$ y $P_m = 0.1$ (por individuo), en ambos casos.
- Número de generaciones: 500 en cada ejecución del proceso genético básico de simplificación y 1000 en el de ajuste.
- Número de soluciones a generar en el proceso genético de multisimplificación: 1.
- Valores asociados al operador de cruce max-min-aritmético y a la mutación no uniforme (véase la sección 2.1.1.3 del capítulo 2) en el proceso genético de ajuste: $a = 0.35$ y $b = 5$.

Tabla 4.2. Resultados obtenidos en los experimentos preliminares realizados para el modelado de la función F_4

| Parámetros | | | Generación | | | Simplificación | | | Ajuste | |
|------------|------------|--------|------------|-------------|-------------|----------------|-------------|-------------|-------------|-------------|
| ω | ϵ | τ | $\#R$ | EC_{entr} | EC_{prue} | $\#R$ | EC_{entr} | EC_{prue} | EC_{entr} | EC_{prue} |
| 0.05 | 0.5 | 0.1 | 182 | 0.57602 | 0.63563 | 145 | 0.48761 | 0.52880 | 0.39404 | 0.42853 |
| 0.05 | 0.5 | 0.25 | | | | 156 | 0.53153 | 0.57128 | 0.41467 | 0.43312 |
| 0.05 | 0.5 | 0.5 | | | | 177 | 0.56355 | 0.61959 | 0.42101 | 0.45404 |
| 0.05 | 1 | 0.1 | 281 | 0.55630 | 0.63904 | 195 | 0.47585 | 0.51951 | 0.37665 | 0.42368 |
| 0.05 | 1 | 0.25 | | | | 198 | 0.48828 | 0.54110 | 0.39916 | 0.45008 |
| 0.05 | 1 | 0.5 | | | | 214 | 0.49672 | 0.55397 | 0.41238 | 0.45742 |
| 0.05 | 1.5 | 0.1 | 373 | 0.58052 | 0.67216 | 249 | 0.46998 | 0.49975 | 0.37217 | 0.38878 |
| 0.05 | 1.5 | 0.25 | | | | 243 | 0.48413 | 0.51895 | 0.39188 | 0.40937 |
| 0.05 | 1.5 | 0.5 | | | | 263 | 0.48271 | 0.51845 | 0.40191 | 0.42646 |
| 0.5 | 0.5 | 0.1 | 175 | 0.58595 | 0.65220 | 138 | 0.49883 | 0.54147 | 0.39859 | 0.42375 |
| 0.5 | 0.5 | 0.25 | | | | 147 | 0.53112 | 0.58508 | 0.39710 | 0.42180 |
| 0.5 | 0.5 | 0.5 | | | | 171 | 0.57548 | 0.64431 | 0.42431 | 0.46188 |
| 0.5 | 1 | 0.1 | 272 | 0.56849 | 0.66579 | 189 | 0.48143 | 0.53047 | 0.39565 | 0.44527 |
| 0.5 | 1 | 0.25 | | | | 191 | 0.48728 | 0.54043 | 0.39978 | 0.44576 |
| 0.5 | 1 | 0.5 | | | | 216 | 0.50472 | 0.56334 | 0.40029 | 0.44161 |
| 0.5 | 1.5 | 0.1 | 367 | 0.58561 | 0.67286 | 238 | 0.47400 | 0.50698 | 0.37627 | 0.38409 |
| 0.5 | 1.5 | 0.25 | | | | 242 | 0.48180 | 0.50536 | 0.39522 | 0.41835 |
| 0.5 | 1.5 | 0.5 | | | | 251 | 0.48821 | 0.53281 | 0.40958 | 0.44248 |

Los distintos resultados obtenidos se muestran en las tablas 4.1 y 4.2, en las que $\#R$ representa el número de reglas existente en la BC correspondiente, y EC_{entr} y EC_{prue} los valores obtenidos en la medida de ECM sobre los conjuntos de entrenamiento y prueba, respectivamente, por el SBRD que emplea la BC en cuestión.

A la vista de estos resultados, podemos constatar la corrección de las conclusiones a las que llegábamos en la sección 3.5 del capítulo anterior. En la gran mayoría de los casos, se obtienen mejores resultados cuanto menor es el valor del parámetro τ . Por otro lado, el empleo de un valor alto para el parámetro ϵ provoca que las BCs aprendidas presenten un número de reglas elevado, pero da lugar a que el SBRD diseñado presente un mejor rendimiento. En cambio, al hacer uso de un valor bajo para ϵ , obtenemos el comportamiento contrario.

Respecto al parámetro ω , se puede comprobar que, en efecto, su valor condiciona en menor medida el comportamiento de los SBRDs diseñados, obteniéndose en muchos casos las mismas definiciones finales de la BC, aun considerando valores distintos para el mismo. Analizando más detenidamente los resultados obtenidos en el modelado de la segunda función, F_4 , se puede observar que el valor 0.5 provoca un comportamiento similar a la combinación de valores bajos para ϵ y τ ; es decir, da lugar a que la BC generada presente menos reglas y un peor rendimiento, aunque las diferencias son mucho

menos significativas que las que se producen al variar los valores de estos dos últimos parámetros.

Puesto que la aplicación en la que estamos empleando nuestro SBRD es una aplicación de modelado cualitativo, cuyo objetivo principal es el conseguir la mayor precisión posible, manteniendo el poder de descripción del modelo obtenido, y no el reducir el tamaño de las BCs generadas, en los experimentos posteriores trabajaremos con la combinación de valores que nos permita satisfacer en lo posible dicho objetivo. Así, los experimentos restantes que presentamos en la sección siguiente han sido efectuados con los valores que se muestran en la tabla 4.3.

Tabla 4.3. Valores considerados para los parámetros del SBRDE

| Parámetro | Valor |
|------------|-------|
| ϵ | 1.5 |
| ω | 0.05 |
| k | 0.1 |
| τ | 0.1 |

4.5.2 Experimentos Realizados

A continuación, mostramos los resultados obtenidos en la aplicación del SBRDE propuesto al modelado cualitativo de las cuatro funciones presentadas en el Apéndice A. Los experimentos han sido realizados teniendo en cuenta los valores de los parámetros comentados en la sección anterior. El único cambio aparece en la segunda fase del proceso evolutivo de aprendizaje, el proceso genético de multisimplificación, en el que generaremos tres definiciones distintas de la BRL empleando el 2.5 por ciento del número de reglas de la base generada en la primera fase, como valor del radio de nicho r , y 0.5, como valor del factor de potencia β (véase la sección 3.3.2 del capítulo 3 para la descripción de ambos parámetros). La tabla 4.4 recoge estos resultados.

Obsérvese que los resultados obtenidos por el proceso de generación en las funciones F_1 y F_4 , coinciden con los mostrados en las tablas 4.1 y 4.2 para la misma combinación de valores de parámetros. El mismo hecho se da en la primera de las definiciones de la BC generada tras la ejecución de los procesos de multisimplificación y ajuste. Esto se debe a que, en todos los casos, se ha empleado la misma semilla aleatoria en la ejecución de los tres procesos que componen el SBRDE en varias etapas.

Analizando estos datos podemos comprobar la efectividad del proceso genético de multisimplificación en algunos de los experimentos. En los experimentos realizados para obtener un modelo cualitativo de las funciones F_1 y F_3 , la mejor BC obtenida finalmente

Tabla 4.4. Resultados obtenidos por el SBRDE propuesto en el modelado cualitativo de las funciones F_1 , F_2 , F_3 y F_4

| Función | Generación | | | Multisimplificación | | | Ajuste | |
|---------|------------|-------------|-------------|---------------------|-------------|-------------|-------------|-------------|
| | #R | EC_{entr} | EC_{prue} | #R | EC_{entr} | EC_{prue} | EC_{entr} | EC_{prue} |
| F_1 | 167 | 3.518764 | 3.498680 | 90 | 1.453064 | 1.508335 | 0.288431 | 0.315454 |
| | | | | 83 | 1.329703 | 1.291596 | 0.301532 | 0.327240 |
| | | | | 85 | 1.358346 | 1.377591 | 0.300626 | 0.324036 |
| F_2 | 156 | 0.220794 | 0.104332 | 94 | 0.037078 | 0.038341 | 0.020508 | 0.019334 |
| | | | | 98 | 0.039022 | 0.045395 | 0.020150 | 0.014971 |
| | | | | 96 | 0.028952 | 0.032070 | 0.011865 | 0.014518 |
| F_3 | 178 | 129389.2 | 161301.1 | 104 | 70079.70 | 74665.57 | 45557.76 | 30962.97 |
| | | | | 100 | 70046.96 | 77168.03 | 46189.13 | 33259.15 |
| | | | | 103 | 68235.86 | 77187.35 | 43856.60 | 35033.07 |
| F_4 | 373 | 0.580528 | 0.672165 | 249 | 0.469988 | 0.499758 | 0.372177 | 0.388787 |
| | | | | 253 | 0.471293 | 0.492110 | 0.370366 | 0.375440 |
| | | | | 251 | 0.471148 | 0.513990 | 0.378052 | 0.412863 |

corresponde a la primera definición de la BRL determinada por el proceso de multisimplificación. En el caso de F_3 , el comportamiento es un poco mejor, ya que la tercera definición da lugar a un modelo con mayor precisión en la aproximación del conjunto de entrenamiento, pero dicho modelo generaliza peor que el original.

En cambio, en las dos funciones restantes el proceso responde de un modo óptimo. En ambas experimentaciones, las definiciones obtenidas a posteriori (la tercera en el caso de F_2 y la segunda en el de F_4) dan lugar a mejores modelos que la primera después de la fase de ajuste, tanto en la aproximación del conjunto de entrenamiento original como en el nivel de generalización que presentan sobre el conjunto de prueba. En el modelado de la función F_2 , además, las dos definiciones de la BRL generadas tras la primera dan lugar a la obtención de modelos cualitativos con el comportamiento comentado.

Al final de este capítulo presentaremos una serie de representaciones gráficas que recogen la forma de las funciones de pertenencia asociadas a las distintas etiquetas lingüísticas existentes después de la actuación del proceso de ajuste. Las particiones difusas representadas en las figuras 4.1 a 4.4 son las asociadas al mejor modelo cualitativo obtenido para cada función, es decir, aquel que presenta un mejor valor en la medida de ECM sobre el conjunto de prueba. Como puede observarse en estas figuras, los términos lingüísticos siguen manteniendo una semántica coherente después del ajuste realizado.

4.5.3 Estudio Comparativo y Análisis de los Resultados Obtenidos

Con objeto de analizar el comportamiento del proceso evolutivo de aprendizaje propuesto, vamos a comparar los resultados obtenidos por los modelos cualitativos generados mediante su uso, con otros obtenidos a partir de dos SBRDEs pertenecientes a la misma familia y que también permiten diseñar SBRDs descriptivos de tipo Mamdani. Los dos SBRDEs considerados posibilitan la generación de una definición completa de una BC en dos fases, en las que, respectivamente, generan la BRL y ajustan la semántica de los términos lingüísticos. Ambos emplean el proceso genético de ajuste introducido en la sección 4.4 como segunda fase y se diferencian en la composición del proceso de generación de reglas lingüísticas que constituye la primera. El primero de ellos emplea el método de generación de reglas de Wang y Mendel (WM) [WM92] (presentado brevemente en el Apéndice B); mientras que el segundo se basa en el uso del proceso evolutivo de aprendizaje de BRLs propuesto por Thrift en [Thr91], el cual fue introducido brevemente en la sección 2.4.2.1 del capítulo 2 de esta memoria. En el estudio comparativo posterior, notaremos respectivamente por **D1** y **D2** a estos dos procesos evolutivos de aprendizaje y por **D3** al SBRDE propuesto en el capítulo actual.

Las ejecuciones del proceso genético de ajuste que constituye la segunda fase de los SBRDEs en varias etapas **D1** y **D2**, han sido realizadas considerando los mismos valores de parámetros empleados en la tercera fase del SBRDE **D3**. En lo que respecta a los procesos de generación, ambos han tomado como base la misma definición inicial de la BD considerada para nuestro proceso y, mientras que el método de WM no usa ningún tipo de parámetro, los valores de los considerados para el AG empleado por el método de Thrift han sido los siguientes: $N = 61$, $P_c = 0.6$, $P_m = 0.1$ (por individuo) y *Número de generaciones* = 1000.

Las tablas 4.5 a 4.8 presentan los resultados obtenidos por los SBRDs generados a partir de estos dos SBRDEs en el modelado de las cuatro funciones, así como los mejores resultados obtenidos en la ejecución de nuestro proceso. Consideraremos mejores aquellos modelos que presenten un menor valor en la medida de ECM sobre el conjunto de prueba, es decir, aquellos que generalicen mejor.

A la vista de las tablas de resultados presentadas, el comportamiento de nuestro SBRDE mejora ampliamente el de los otros dos procesos presentados en todos los casos excepto en el del modelado de la función F_3 en el que, a pesar de obtener un modelo que aproxima mejor el conjunto de entrenamiento que el generado empleando el SBRDE **D2** basado en el proceso de generación de Thrift, su nivel de generalización es algo peor que el obtenido empleando este último proceso evolutivo de aprendizaje.

Con objeto de ilustrar de un modo más claro el comportamiento de los tres SBRDEs comparados, en las figuras 4.5 a 4.8, recogidas al final del capítulo, presentamos las representaciones gráficas de los distintos modelos obtenidos mediante su aplicación para cada una de las funciones seleccionadas. Estas figuras han sido generadas representando

Tabla 4.5. Resultados obtenidos en el modelado cualitativo de la función F_1

| SBRDE | #R | Generación | | Multisimplificación | | | Ajuste | |
|-----------|-----|-------------|-------------|---------------------|-------------|-------------|-------------|-------------|
| | | EC_{entr} | EC_{prue} | #R | EC_{entr} | EC_{prue} | EC_{entr} | EC_{prue} |
| D1 | 49 | 2.048137 | 2.255928 | | | | 0.358522 | 0.377134 |
| D2 | 49 | 1.609885 | 1.221451 | | | | 0.672429 | 0.551139 |
| D3 | 167 | 3.518764 | 3.498680 | 90 | 1.453064 | 1.508335 | 0.288431 | 0.315454 |

Tabla 4.6. Resultados obtenidos en el modelado cualitativo de la función F_2

| SBRDE | #R | Generación | | Multisimplificación | | | Ajuste | |
|-----------|-----|-------------|-------------|---------------------|-------------|-------------|-------------|-------------|
| | | EC_{entr} | EC_{prue} | #R | EC_{entr} | EC_{prue} | EC_{entr} | EC_{prue} |
| D1 | 49 | 0.194386 | 0.044466 | | | | 0.060296 | 0.028621 |
| D2 | 48 | 0.097398 | 0.043984 | | | | 0.051477 | 0.028996 |
| D3 | 156 | 0.220794 | 0.104332 | 96 | 0.028952 | 0.032070 | 0.011865 | 0.014518 |

Tabla 4.7. Resultados obtenidos en el modelado cualitativo de la función F_3

| SBRDE | #R | Generación | | Multisimplificación | | | Ajuste | |
|-----------|-----|-------------|-------------|---------------------|-------------|-------------|-------------|-------------|
| | | EC_{entr} | EC_{prue} | #R | EC_{entr} | EC_{prue} | EC_{entr} | EC_{prue} |
| D1 | 49 | 172530.0 | 206244.0 | | | | 52705.00 | 40999.98 |
| D2 | 47 | 68675.28 | 60022.20 | | | | 49910.34 | 29614.56 |
| D3 | 178 | 129389.2 | 161301.1 | 104 | 70079.70 | 74665.57 | 45557.76 | 30962.97 |

Tabla 4.8. Resultados obtenidos en el modelado cualitativo de la función F_4

| SBRDE | #R | Generación | | Multisimplificación | | | Ajuste | |
|-----------|-----|-------------|-------------|---------------------|-------------|-------------|-------------|-------------|
| | | EC_{entr} | EC_{prue} | #R | EC_{entr} | EC_{prue} | EC_{entr} | EC_{prue} |
| D1 | 49 | 1.778353 | 2.049079 | | | | 0.814157 | 0.964132 |
| D2 | 48 | 0.503970 | 0.547687 | | | | 0.403650 | 0.432996 |
| D3 | 373 | 0.580528 | 0.672165 | 253 | 0.471293 | 0.492110 | 0.370366 | 0.375440 |

gráficamente las salidas que devuelven dichos modelos ante las entradas contenidas en los dos ficheros de datos, el de entrenamiento y el de prueba.

La razón de que los modelos cualitativos obtenidos mediante nuestro SBRDE presenten mejor comportamiento, se debe a la estructura del SBRD descriptivo de tipo Mamdani considerado, el cual, como hemos indicado previamente, permite emplear reglas lingüísticas con el mismo antecedente y más de un consecuente. En las tablas 4.9 a 4.13, presentadas también al final del capítulo, mostraremos las BRLs que componen las BCs que constituyen los modelos cualitativos aprendidos empleando los tres SBRDEs con objeto de comprobar las ideas comentadas.

Como puede observarse, en la gran mayoría de los casos, las BRLs aprendidas mediante nuestro SBRDE incluyen a las generadas a partir de los otros dos procesos evolutivos de aprendizaje considerados, **D1** y **D2**. Los únicos casos en los cuales una regla generada

mediante dichos procesos no aparece en la BRL obtenida a partir de nuestro proceso evolutivo de aprendizaje son aquellos en los que dicha regla ha sido desestimada por el proceso genético de multisimplificación, al no presentar un buen comportamiento en su cooperación con las demás existentes en la base. De este modo, podemos concluir que el SBRDE propuesto extiende de un modo adecuado los otros dos estudiados.

Otra conclusión a destacar es que ninguno de los modelos cualitativos obtenidos presenta un comportamiento aceptable en el caso de la función F_4 . Las BCs de los modelos generados a partir de los SBRDEs **D1** y **D2** contienen un número de reglas pequeño que permite que puedan ser interpretados adecuadamente, pero su precisión es bastante pobre. Nuestro SBRDE permite generar un modelo con mayor precisión, pero la complejidad de la BC hace que no sea interpretable en la práctica y, aun así, su nivel de precisión es demasiado limitado, por lo que no se verifican adecuadamente ninguno de los dos requerimientos exigidos.

Esto deja patente las limitaciones que presentan los SBRDEs descriptivos de tipo Mamdani en el modelado cualitativo. Como ya comentábamos en la sección 1.1.3 del primer capítulo, estas limitaciones se deben a la rigidez de la estructura de regla difusa que manejan. El hecho de emplear particiones difusas para definir de una forma unívoca la semántica de los términos lingüísticos existentes en la BC, independientemente de la regla en la que se empleen, hace que estos SBRDs presenten mal comportamiento en problemas en los cuales pequeñas diferencias en las entradas tienen asociadas grandes diferencias en la salida, como por ejemplo el modelado de la función F_4 .

Esta dificultad no puede ser subsanada por el SBRDE propuesto ya que, al tratar de solventarla, el sistema genera una BRL formada por un número muy alto de reglas lingüísticas, buscando el conseguir un nivel de precisión aceptable. Esto provoca a su vez que cada combinación concreta de valores de las variables de entrada tenga asociada un gran número de etiquetas distintas en la variable de salida (como puede observarse en la tabla 4.13), lo cual no es muy coherente. *Este tipo de problemas se aproximarían mejor mediante un modelo difuso obtenido a partir de un SBRD aproximativo*, como veremos en el siguiente capítulo.

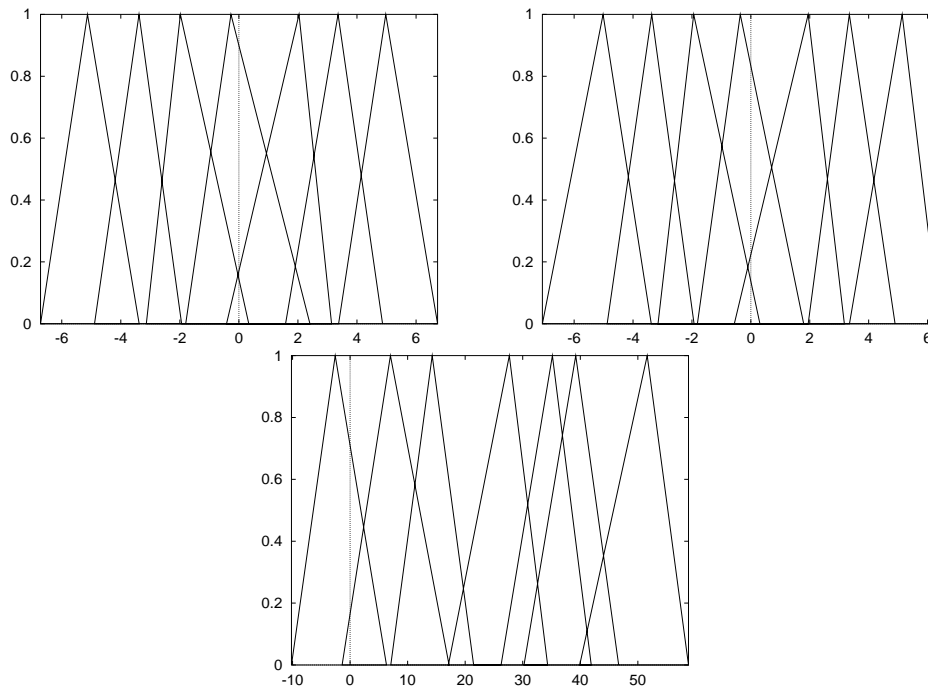


Figura 4.1. Particiones difusas ajustadas asociadas a las variables x_1 (arriba a la izquierda), x_2 (arriba a la derecha) e y (abajo) en el mejor modelo cualitativo obtenido para la función F_1

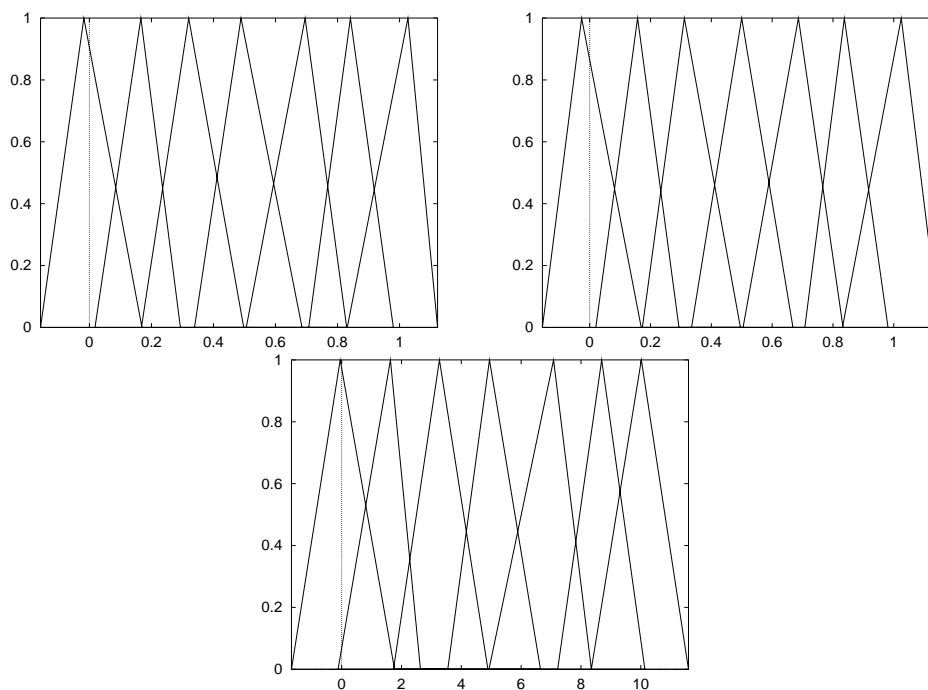


Figura 4.2. Particiones difusas ajustadas asociadas a las variables x_1 (arriba a la izquierda), x_2 (arriba a la derecha) e y (abajo) en el mejor modelo cualitativo obtenido para la función F_2

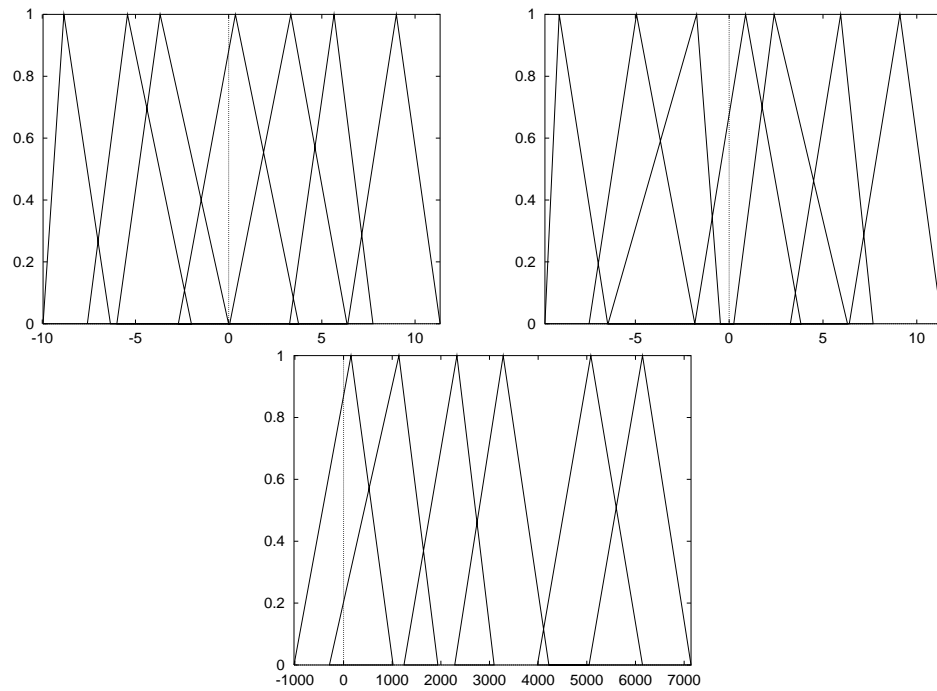


Figura 4.3. Particiones difusas ajustadas asociadas a las variables x_1 (arriba a la izquierda), x_2 (arriba a la derecha) e y (abajo) en el mejor modelo cualitativo obtenido para la función F_3

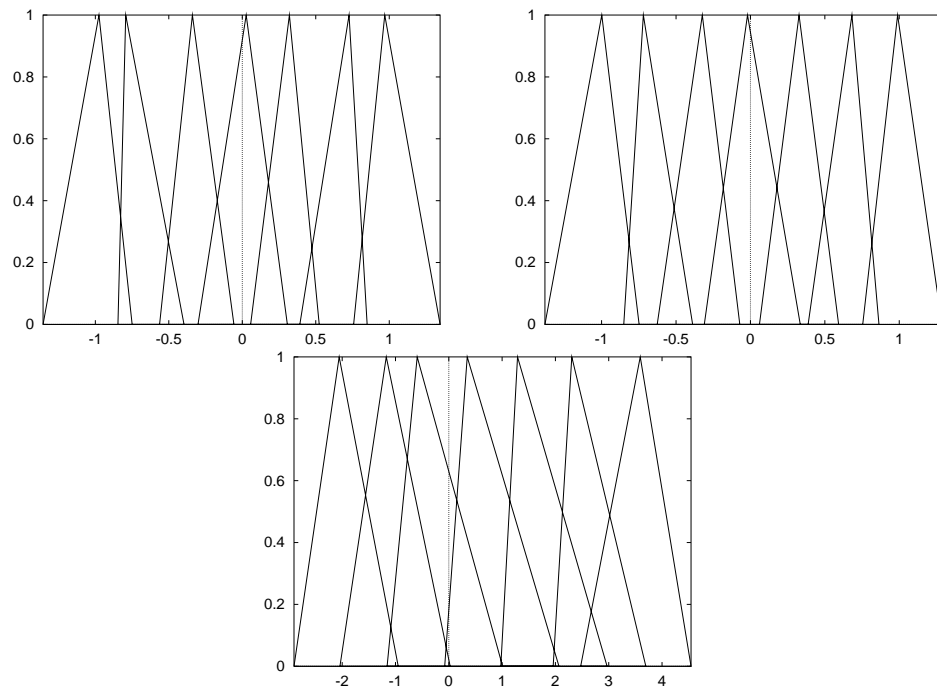


Figura 4.4. Particiones difusas ajustadas asociadas a las variables x_1 (arriba a la izquierda), x_2 (arriba a la derecha) e y (abajo) en el mejor modelo cualitativo obtenido para la función F_4

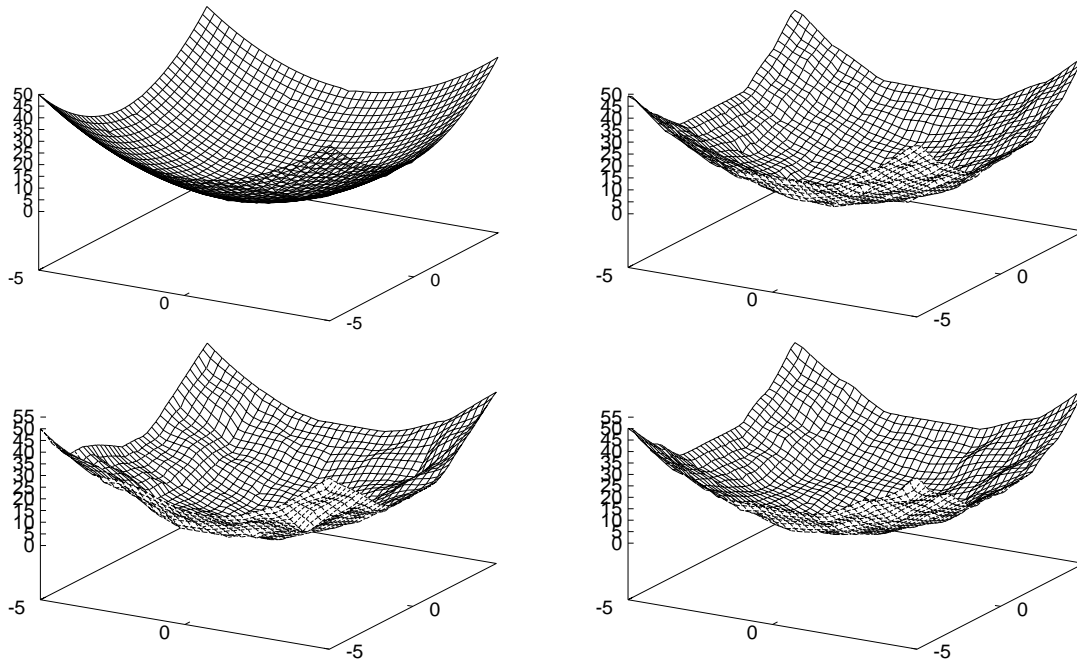


Figura 4.5. Representaciones gráficas de la función F_1 (arriba a la izquierda) y de los tres modelados obtenidos para ella empleando los procesos **D1** (arriba a la derecha), **D2** (abajo a la izquierda) y **D3** (abajo a la derecha)

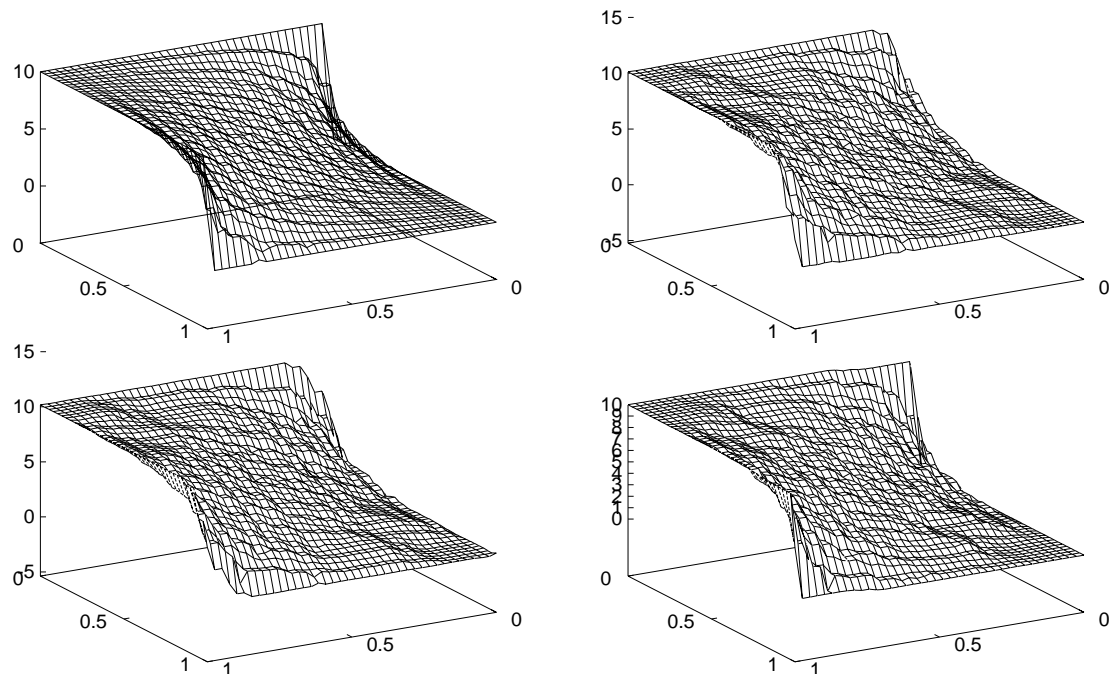


Figura 4.6. Representaciones gráficas de la función F_2 (arriba a la izquierda) y de los tres modelados obtenidos para ella empleando los procesos **D1** (arriba a la derecha), **D2** (abajo a la izquierda) y **D3** (abajo a la derecha)

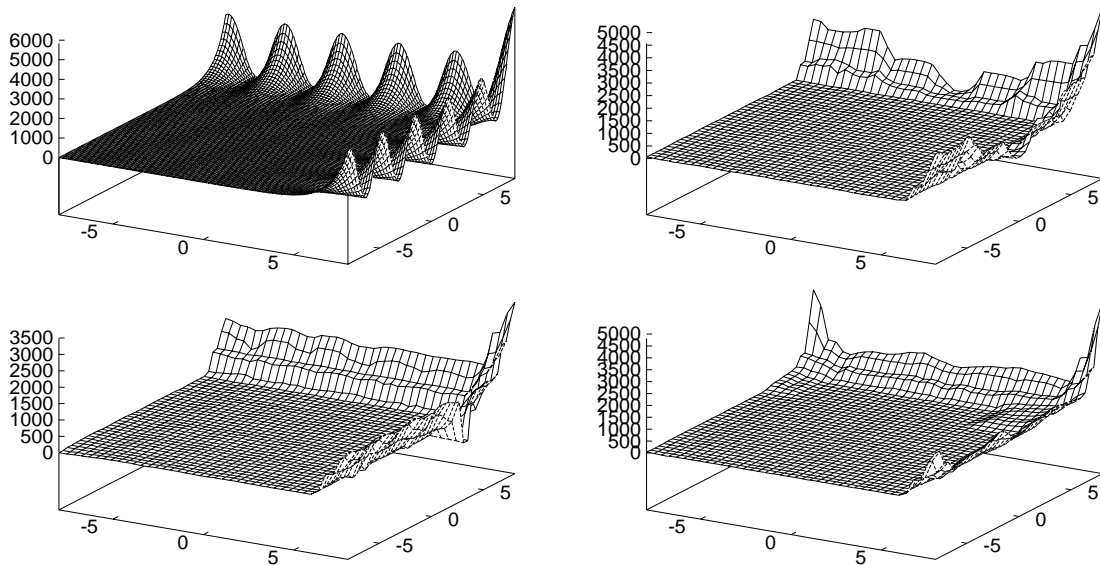


Figura 4.7. Representaciones gráficas de la función F_3 (arriba a la izquierda) y de los tres modelados obtenidos para ella empleando los procesos **D1** (arriba a la derecha), **D2** (abajo a la izquierda) y **D3** (abajo a la derecha)

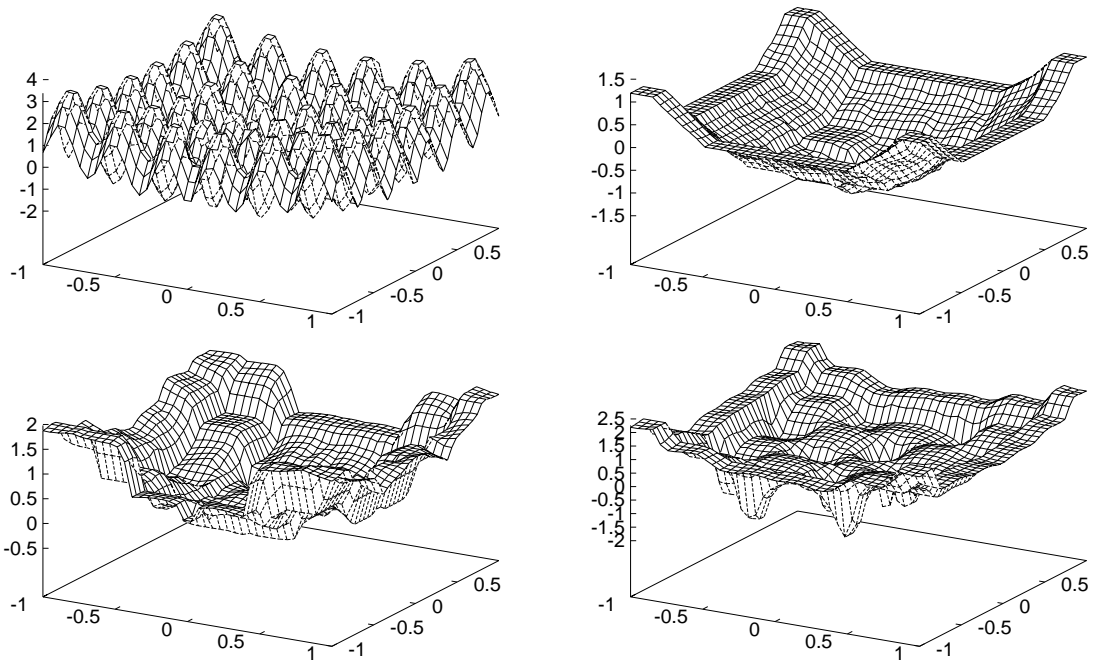


Figura 4.8. Representaciones gráficas de la función F_4 (arriba a la izquierda) y de los tres modelados obtenidos para ella empleando los procesos **D1** (arriba a la derecha), **D2** (abajo a la izquierda) y **D3** (abajo a la derecha)

Tabla 4.9. Tablas de decisión de los modelos cualitativos obtenidos para la función F_1 mediante los SBRDEs **D1** (arriba), **D2** (en medio) y **D3** (abajo)

| | | | | | | | | |
|-------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | | x_2 | | | | | | |
| | | <i>NG</i> | <i>NM</i> | <i>NP</i> | <i>CE</i> | <i>PP</i> | <i>PM</i> | <i>PG</i> |
| x_1 | <i>NG</i> | <i>PG</i> | <i>PP</i> | <i>CE</i> | <i>CE</i> | <i>CE</i> | <i>PP</i> | <i>PG</i> |
| | <i>NM</i> | <i>PP</i> | <i>CE</i> | <i>NP</i> | <i>NM</i> | <i>NP</i> | <i>CE</i> | <i>PP</i> |
| | <i>NP</i> | <i>CE</i> | <i>NP</i> | <i>NM</i> | <i>NG</i> | <i>NM</i> | <i>NP</i> | <i>CE</i> |
| | <i>CE</i> | <i>CE</i> | <i>NM</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NM</i> | <i>CE</i> |
| | <i>PP</i> | <i>CE</i> | <i>NP</i> | <i>NM</i> | <i>NG</i> | <i>NM</i> | <i>NP</i> | <i>CE</i> |
| | <i>PM</i> | <i>PP</i> | <i>CE</i> | <i>NP</i> | <i>NM</i> | <i>NP</i> | <i>CE</i> | <i>PP</i> |
| | <i>PG</i> | <i>PG</i> | <i>PP</i> | <i>CE</i> | <i>CE</i> | <i>CE</i> | <i>PP</i> | <i>PG</i> |

| | | | | | | | | |
|-------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | | x_2 | | | | | | |
| | | <i>NG</i> | <i>NM</i> | <i>NP</i> | <i>CE</i> | <i>PP</i> | <i>PM</i> | <i>PG</i> |
| x_1 | <i>NG</i> | <i>PG</i> | <i>PP</i> | <i>PP</i> | <i>CE</i> | <i>CE</i> | <i>PM</i> | <i>PG</i> |
| | <i>NM</i> | <i>PP</i> | <i>CE</i> | <i>NM</i> | <i>NM</i> | <i>NP</i> | <i>NP</i> | <i>PP</i> |
| | <i>NP</i> | <i>PP</i> | <i>NM</i> | <i>NM</i> | <i>NG</i> | <i>NG</i> | <i>NP</i> | <i>CE</i> |
| | <i>CE</i> | <i>CE</i> | <i>NM</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NM</i> | <i>CE</i> |
| | <i>PP</i> | <i>CE</i> | <i>NP</i> | <i>CE</i> | <i>CE</i> | <i>CE</i> | <i>NP</i> | <i>CE</i> |
| | <i>PM</i> | <i>PP</i> | <i>NP</i> | <i>NP</i> | <i>NM</i> | <i>NP</i> | <i>NP</i> | <i>PP</i> |
| | <i>PG</i> | <i>PG</i> | <i>PM</i> | <i>CE</i> | <i>CE</i> | <i>CE</i> | <i>PM</i> | <i>PG</i> |

| | | | | | | | | |
|-------|-----------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | | x_2 | | | | | | |
| | | <i>NG</i> | <i>NM</i> | <i>NP</i> | <i>CE</i> | <i>PP</i> | <i>PM</i> | <i>PG</i> |
| x_1 | <i>NG</i> | $2 \cdot PG$ | PP PG | $2 \cdot CE$ | $2 \cdot CE$ | <i>CE</i> | $2 \cdot PP$ | $2 \cdot PG$ |
| | <i>NM</i> | $2 \cdot PP$ | NP CE | $2 \cdot NP$ | $2 \cdot NM$ | <i>NP</i> | NP CE | PP PM |
| | <i>NP</i> | <i>CE</i> | <i>NP</i> | $3 \cdot NM$ | <i>NG</i> | $3 \cdot NM$ | $2 \cdot NP$ | $2 \cdot CE$ |
| | <i>CE</i> | $2 \cdot CE$ | $2 \cdot NM$ | <i>NG</i> | $3 \cdot NG$ | <i>NG</i> | $2 \cdot NM$ | $2 \cdot CE$ |
| | <i>PP</i> | $2 \cdot CE$ | $2 \cdot NP$ | $3 \cdot NM$ | <i>NG</i> | $3 \cdot NM$ | <i>NP</i> | <i>CE</i> |
| | <i>PM</i> | PP PM | NP CE | $2 \cdot NP$ | $2 \cdot NM$ | <i>NP</i> | NP CE | $2 \cdot PP$ |
| | <i>PG</i> | <i>PG</i> | PP PM | $2 \cdot CE$ | $2 \cdot CE$ | <i>CE</i> | $2 \cdot PP$ | $2 \cdot PG$ |

Tabla 4.10. Tablas de decisión de los modelos cualitativos obtenidos para la función F_2 mediante los SBRDEs **D1** (arriba), **D2** (en medio) y **D3** (abajo)

| | | | | | | | |
|-------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | | x_2 | | | | | |
| | | <i>NG</i> | <i>NM</i> | <i>NP</i> | <i>CE</i> | <i>PP</i> | <i>PM</i> |
| x_1 | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> |
| | <i>NM</i> | <i>NG</i> | <i>CE</i> | <i>NP</i> | <i>NM</i> | <i>NM</i> | <i>NG</i> |
| | <i>NP</i> | <i>PG</i> | <i>PP</i> | <i>CE</i> | <i>NP</i> | <i>NM</i> | <i>NM</i> |
| | <i>CE</i> | <i>PG</i> | <i>PM</i> | <i>PP</i> | <i>CE</i> | <i>NP</i> | <i>NM</i> |
| | <i>PP</i> | <i>PG</i> | <i>PM</i> | <i>PM</i> | <i>PP</i> | <i>CE</i> | <i>NP</i> |
| | <i>PM</i> | <i>PG</i> | <i>PG</i> | <i>PM</i> | <i>PM</i> | <i>PP</i> | <i>CE</i> |
| | <i>PG</i> | <i>PG</i> | <i>PG</i> | <i>PG</i> | <i>PG</i> | <i>PG</i> | <i>PG</i> |

| | | | | | | | |
|-------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | | x_2 | | | | | |
| | | <i>NG</i> | <i>NM</i> | <i>NP</i> | <i>CE</i> | <i>PP</i> | <i>PM</i> |
| x_1 | <i>NG</i> | <i>CE</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> |
| | <i>NM</i> | <i>PG</i> | <i>CE</i> | <i>NP</i> | <i>NM</i> | <i>NM</i> | <i>NG</i> |
| | <i>NP</i> | <i>PG</i> | <i>PP</i> | <i>CE</i> | <i>NP</i> | <i>NM</i> | <i>NM</i> |
| | <i>CE</i> | <i>PG</i> | <i>PM</i> | <i>PP</i> | <i>CE</i> | <i>NP</i> | <i>NM</i> |
| | <i>PP</i> | <i>PG</i> | <i>PM</i> | <i>PM</i> | <i>PP</i> | <i>CE</i> | <i>NP</i> |
| | <i>PM</i> | <i>PG</i> | <i>PG</i> | <i>PM</i> | <i>PM</i> | <i>PP</i> | <i>CE</i> |
| | <i>PG</i> | <i>PG</i> | <i>PG</i> | <i>PG</i> | <i>PG</i> | <i>PG</i> | <i>PG</i> |

| | | | | | | | |
|-------|-----------|---------------------------|-------------------------------------|--------------|--------------|--------------|--|
| | | x_2 | | | | | |
| | | <i>NG</i> | <i>NM</i> | <i>NP</i> | <i>CE</i> | <i>PP</i> | <i>PM</i> |
| x_1 | <i>NG</i> | | $2 \cdot NG$ <i>NP</i> | $2 \cdot NG$ | $2 \cdot NG$ | $2 \cdot NG$ | $2 \cdot NG$ |
| | <i>NM</i> | <i>PM</i> $2 \cdot PG$ | <i>NP</i> <i>CE</i> <i>PP</i> | $2 \cdot NP$ | $2 \cdot NM$ | <i>NM</i> | $3 \cdot NG$ $2 \cdot NG$ |
| | <i>NP</i> | $2 \cdot PG$ | $2 \cdot PP$ | <i>CE</i> | $2 \cdot NP$ | $2 \cdot NM$ | <i>NM</i> $2 \cdot NG$ |
| | <i>CE</i> | $2 \cdot PG$ | $2 \cdot PM$ | $2 \cdot PP$ | $2 \cdot CE$ | $2 \cdot NP$ | $2 \cdot NM$ $2 \cdot NG$ |
| | <i>PP</i> | $2 \cdot PG$ | <i>PM</i> | $2 \cdot PM$ | $2 \cdot PP$ | <i>CE</i> | $2 \cdot NP$ $2 \cdot NG$ |
| | <i>PM</i> | $2 \cdot PG$ | $3 \cdot PG$ | <i>PM</i> | $2 \cdot PM$ | $2 \cdot PP$ | <i>NP</i> <i>CE</i> <i>PP</i> $2 \cdot NG$ <i>NM</i> |
| | <i>PG</i> | $2 \cdot PG$ | $2 \cdot PG$ | $2 \cdot PG$ | $2 \cdot PG$ | $2 \cdot PG$ | <i>PM</i> $2 \cdot PG$ |

Tabla 4.11. Tablas de decisión de los modelos cualitativos obtenidos para la función F_3 mediante los SBRDEs **D1** (arriba), **D2** (en medio) y **D3** (abajo)

| | | | | | | | | |
|-------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | | x_2 | | | | | | |
| | | <i>NG</i> | <i>NM</i> | <i>NP</i> | <i>CE</i> | <i>PP</i> | <i>PM</i> | <i>PG</i> |
| x_1 | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>CE</i> |
| | <i>NM</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>CE</i> |
| | <i>NP</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NM</i> |
| | <i>CE</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> |
| | <i>PP</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NM</i> |
| | <i>PM</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>CE</i> |
| | <i>PG</i> | <i>CE</i> | <i>CE</i> | <i>NM</i> | <i>NG</i> | <i>NM</i> | <i>CE</i> | <i>PG</i> |

| | | | | | | | | |
|-------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | | x_2 | | | | | | |
| | | <i>NG</i> | <i>NM</i> | <i>NP</i> | <i>CE</i> | <i>PP</i> | <i>PM</i> | <i>PG</i> |
| x_1 | <i>NG</i> | | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NM</i> |
| | <i>NM</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NM</i> |
| | <i>NP</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NM</i> |
| | <i>CE</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NM</i> |
| | <i>PP</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NM</i> |
| | <i>PM</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NM</i> |
| | <i>PG</i> | <i>NM</i> | <i>NM</i> | <i>NM</i> | <i>NM</i> | <i>NM</i> | | <i>NM</i> |

| | | | | | | | | |
|-------|-----------|------------------------|-------------------------------------|------------------------|-------------------------------------|------------------------|------------------------------|--|
| | | x_2 | | | | | | |
| | | <i>NG</i> | <i>NM</i> | <i>NP</i> | <i>CE</i> | <i>PP</i> | <i>PM</i> | <i>PG</i> |
| x_1 | <i>NG</i> | <i>NG</i> | $2 \cdot NG$ | <i>NG</i> | <i>NG</i> | $2 \cdot NG$ | $2 \cdot NG$ <i>NP</i> | <i>CE</i> |
| | <i>NM</i> | <i>NG</i> | $2 \cdot NG$ | <i>NG</i> | <i>NG</i> | $2 \cdot NG$ | $2 \cdot NG$ | $2 \cdot NG$ <i>NM</i> <i>NP</i> |
| | <i>NP</i> | <i>NG</i> | $2 \cdot NG$ | $2 \cdot NG$ | $2 \cdot NG$ | $2 \cdot NG$ | $2 \cdot NG$ | $2 \cdot NG$ <i>CE</i> |
| | <i>CE</i> | $2 \cdot NG$ | $2 \cdot NG$ | $2 \cdot NG$ | $2 \cdot NG$ | $2 \cdot NG$ | $2 \cdot NG$ | <i>NG</i> <i>NM</i> <i>NP</i> |
| | <i>PP</i> | $2 \cdot NG$ | $2 \cdot NG$ | $2 \cdot NG$ | <i>NG</i> | $3 \cdot NG$ | $2 \cdot NG$ | <i>NG</i> <i>CE</i> |
| | <i>PM</i> | $2 \cdot NG$ | $2 \cdot NG$ | $2 \cdot NG$ | $2 \cdot NG$ | $2 \cdot NG$ | $2 \cdot NG$ $3 \cdot NM$ | $3 \cdot NM$ <i>NP</i> |
| | <i>PG</i> | <i>NM</i> <i>CE</i> | <i>NG</i> <i>NM</i> <i>NP</i> | <i>NG</i> <i>CE</i> | <i>NG</i> <i>NM</i> <i>NP</i> | <i>NG</i> <i>CE</i> | $3 \cdot NM$ <i>NP</i> | <i>CE</i> <i>PM</i> <i>PG</i> |

Tabla 4.12. Tablas de decisión de los modelos cualitativos obtenidos para la función F_4 mediante los SBRDEs **D1** (arriba) y **D2** (abajo)

| | | x_2 | | | | | | |
|-------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | | <i>NG</i> | <i>NM</i> | <i>NP</i> | <i>CE</i> | <i>PP</i> | <i>PM</i> | <i>PG</i> |
| x_1 | <i>NG</i> | <i>CE</i> | <i>NP</i> | <i>NP</i> | <i>NP</i> | <i>NP</i> | <i>NP</i> | <i>CE</i> |
| | <i>NM</i> | <i>NP</i> | <i>NM</i> | <i>NM</i> | <i>NM</i> | <i>NM</i> | <i>NM</i> | <i>NP</i> |
| | <i>NP</i> | <i>NP</i> | <i>NM</i> | <i>NM</i> | <i>NG</i> | <i>NM</i> | <i>NM</i> | <i>NP</i> |
| | <i>CE</i> | <i>NP</i> | <i>NM</i> | <i>NG</i> | <i>NG</i> | <i>NG</i> | <i>NM</i> | <i>NP</i> |
| | <i>PP</i> | <i>NP</i> | <i>NM</i> | <i>NM</i> | <i>NG</i> | <i>NM</i> | <i>NM</i> | <i>NP</i> |
| | <i>PM</i> | <i>NP</i> | <i>NM</i> | <i>NM</i> | <i>NM</i> | <i>NM</i> | <i>NM</i> | <i>NP</i> |
| | <i>PG</i> | <i>CE</i> | <i>NP</i> | <i>NP</i> | <i>NP</i> | <i>NP</i> | <i>NP</i> | <i>CE</i> |

| | | x_2 | | | | | | |
|-------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | | <i>NG</i> | <i>NM</i> | <i>NP</i> | <i>CE</i> | <i>PP</i> | <i>PM</i> | <i>PG</i> |
| x_1 | <i>NG</i> | <i>PP</i> | <i>PP</i> | <i>PP</i> | <i>CE</i> | <i>PP</i> | <i>PP</i> | <i>PP</i> |
| | <i>NM</i> | <i>PP</i> | <i>CE</i> | <i>CE</i> | <i>NP</i> | <i>CE</i> | <i>CE</i> | <i>PP</i> |
| | <i>NP</i> | <i>CE</i> | <i>CE</i> | <i>NP</i> | <i>CE</i> | <i>NP</i> | <i>CE</i> | <i>CE</i> |
| | <i>CE</i> | | <i>CE</i> | <i>NP</i> | <i>NP</i> | <i>NP</i> | <i>CE</i> | <i>CE</i> |
| | <i>PP</i> | <i>CE</i> | <i>CE</i> | <i>NP</i> | <i>CE</i> | <i>NP</i> | <i>CE</i> | <i>CE</i> |
| | <i>PM</i> | <i>PP</i> | <i>CE</i> | <i>CE</i> | <i>NP</i> | <i>CE</i> | <i>CE</i> | <i>PP</i> |
| | <i>PG</i> | <i>PP</i> | <i>PP</i> | <i>CE</i> | <i>PP</i> | <i>CE</i> | <i>PP</i> | <i>PP</i> |

Tabla 4.13. Tabla de decisión del modelo cualitativo obtenido para la función F_4 mediante el SBRDE **D3**

| | | x_2 | | | | | | |
|-------|-----------|---|--|--|---|--|--|--|
| | | <i>NG</i> | <i>NM</i> | <i>NP</i> | <i>CE</i> | <i>PP</i> | <i>PM</i> | <i>PG</i> |
| x_1 | <i>NG</i> | <i>NG</i> 3 · <i>NM</i> 3 · <i>NP</i> <i>CE</i> | 2 · <i>NM</i> <i>NP</i> <i>CE</i> <i>PP</i> | 2 · <i>NP</i> <i>CE</i> <i>PP</i> <i>PG</i> | <i>NM</i> <i>NP</i> 2 · <i>CE</i> <i>PP</i> 3 · <i>PG</i> | <i>NP</i> <i>CE</i> 2 · <i>PP</i> <i>PM</i> | <i>NP</i> <i>CE</i> 2 · <i>PP</i> 2 · <i>PM</i> | <i>CE</i> 3 · <i>PP</i> 3 · <i>PM</i> <i>PG</i> |
| | <i>NM</i> | <i>NP</i> <i>CE</i> 2 · <i>PP</i> 2 · <i>PM</i> | 2 · <i>NP</i> 2 · <i>CE</i> <i>PP</i> <i>PM</i> | <i>NM</i> | <i>NM</i> <i>NP</i> 2 · <i>CE</i> <i>PP</i> | 2 · <i>NM</i> <i>PP</i> | 2 · <i>NP</i> 2 · <i>CE</i> <i>PP</i> <i>PM</i> | <i>NP</i> <i>CE</i> <i>PP</i> 2 · <i>PM</i> |
| | <i>NP</i> | <i>NP</i> 2 · <i>CE</i> <i>PP</i> <i>PM</i> | <i>NM</i> | <i>NM</i> 2 · <i>NP</i> 3 · <i>CE</i> 2 · <i>PP</i> | <i>NM</i> | <i>NG</i> <i>NP</i> 3 · <i>CE</i> 2 · <i>PP</i> | <i>NM</i> <i>NP</i> <i>PP</i> | 2 · <i>NP</i> <i>CE</i> 2 · <i>PP</i> <i>PM</i> |
| | <i>CE</i> | <i>NM</i> <i>NP</i> 2 · <i>CE</i> 2 · <i>PP</i> 2 · <i>PM</i> | <i>NM</i> 2 · <i>CE</i> <i>PP</i> | <i>NM</i> | <i>NG</i> 2 · <i>NP</i> 2 · <i>PP</i> | <i>NG</i> | <i>NM</i> <i>NP</i> 2 · <i>CE</i> <i>PP</i> | 2 · <i>NP</i> 2 · <i>CE</i> 2 · <i>PP</i> 2 · <i>PM</i> |
| | <i>PP</i> | 2 · <i>NP</i> 2 · <i>CE</i> 2 · <i>PP</i> <i>PM</i> | <i>NM</i> | <i>NM</i> 2 · <i>NP</i> 3 · <i>CE</i> 2 · <i>PP</i> | <i>NM</i> | <i>NM</i> 2 · <i>NP</i> 2 · <i>CE</i> 2 · <i>PP</i> | 2 · <i>NM</i> <i>PP</i> | <i>NP</i> 2 · <i>CE</i> 2 · <i>PP</i> <i>PM</i> |
| | <i>PM</i> | <i>NM</i> <i>CE</i> 2 · <i>PP</i> 2 · <i>PM</i> | 2 · <i>NP</i> 2 · <i>CE</i> <i>PP</i> <i>PM</i> | <i>NM</i> <i>NP</i> <i>PP</i> | <i>NM</i> 2 · <i>CE</i> | <i>NM</i> <i>NP</i> <i>CE</i> <i>PP</i> | 2 · <i>NP</i> 2 · <i>CE</i> <i>PP</i> <i>PM</i> | <i>NP</i> <i>CE</i> 2 · <i>PP</i> 2 · <i>PM</i> |
| | <i>PG</i> | <i>CE</i> 3 · <i>PP</i> 3 · <i>PM</i> <i>PG</i> | 2 · <i>NP</i> <i>PP</i> 2 · <i>PM</i> | <i>NP</i> 2 · <i>CE</i> 2 · <i>PP</i> <i>PM</i> | <i>NM</i> 2 · <i>NP</i> <i>CE</i> <i>PP</i> 3 · <i>PM</i> | <i>NP</i> 2 · <i>CE</i> 2 · <i>PP</i> <i>PM</i> | <i>NP</i> <i>CE</i> <i>PP</i> 2 · <i>PM</i> | <i>CE</i> 3 · <i>PP</i> 3 · <i>PM</i> <i>PG</i> |

Capítulo 5

Dos Sistemas Basados en Reglas Difusas Evolutivos para el Aprendizaje de Bases de Reglas Difusas Aproximativas de tipo Mamdani

Los experimentos realizados en el capítulo anterior han puesto de manifiesto las limitaciones de los modelos cualitativos cuando se trabaja con problemas complejos en los que pequeños cambios en las entradas están asociados con grandes cambios en las salidas. En estos casos, se precisa del empleo de los SBRDs aproximativos, los cuales no están sujetos a las limitaciones debidas al manejo de variables lingüísticas en la estructura de regla difusa considerada.

En este quinto capítulo, nos planteamos la construcción de métodos evolutivos de aprendizaje que nos permitan diseñar SBRDs aproximativos de tipo Mamdani para su aplicación en problemas de modelado difuso. Puesto que en los procesos de aprendizaje de BRDs aproximativas es necesario llevar a cabo la generación de la forma de las funciones de pertenencia consideradas en cada regla individual, podemos distinguir entre dos posibilidades según el modo en el que se afronte la generación de éstas:

- *Semántica libre restringida*: Las funciones de pertenencia de las reglas contenidas en la BRD se obtienen tomando como base unas particiones difusas iniciales asociadas a cada variable que determinan los intervalos en los que puede variar cada parámetro.
- *Semántica libre no restringida*: En este caso, las únicas restricciones que se imponen a las funciones de pertenencia es que sean coherentes y que estén definidas dentro de un intervalo global concreto. El caso más extremo se daría cuando este intervalo coincidiera con el universo de discurso de la variable en cuestión.

Ambos enfoques dan lugar a que los SBRDs aproximativos de tipo Mamdani generados

se comporten mejor que los descriptivos en el modelado de sistemas con un nivel de complejidad alto pero existe una cierta diferencia que determina el área de aplicación concreta de cada uno de ellos:

- los SBRDs aproximativos obtenidos a partir de un proceso basado en semántica restringida consiguen mejores resultados en el caso en el que dicho nivel de complejidad sea intermedio.
- En cambio, los SBRDs aproximativos obtenidos a partir de un proceso de aprendizaje basado en semántica no restringida presentan mejor rendimiento en el modelado difuso de sistemas muy complejos.

Dedicaremos las dos grandes secciones que forman este capítulo a presentar los dos SBRDEs comentados, incluyendo en cada una de ellas la descripción de las etapas de las que están compuestos y la experimentación realizada sobre las funciones introducidas en el Apéndice A. Finalmente, compararemos los resultados obtenidos por los modelos difusos generados a partir de ambos SBRDEs con los de otros dos pertenecientes a la misma familia.

5.1 Un Sistema Basado en Reglas Difusas Evolutivo Según el Enfoque de Semántica Libre No Restringida

Esta primera sección está dedicada a introducir el SBRDE para el aprendizaje de BRDs aproximativas de tipo Mamdani con semántica libre no restringida. A continuación, presentaremos la estructura en tres etapas de dicho sistema, para introducir posteriormente la composición concreta de la primera fase del mismo y matizar brevemente la composición de la tercera etapa. Finalmente, aplicaremos el SBRDE propuesto al modelado difuso de las cuatro funciones tridimensionales presentadas en el Apéndice A.

5.1.1 Estructura del Sistema Basado en Reglas Difusas Evolutivo en Varias Etapas

El SBRDE para el aprendizaje automático de BRDs aproximativas de tipo Mamdani ha sido construido siguiendo la metodología evolutiva de aprendizaje introducida en el capítulo 3 de esta memoria. De este modo, está formado por una estructura en tres etapas basada en los tres procesos siguientes:

1. *Un proceso evolutivo de generación de reglas difusas aproximativas de tipo Mamdani*, formado por un *método de generación de reglas difusas aproximativas*, compuesto por una $EE - (1 + 1)$ guiada por una función de adaptación basada en el criterio de baja interacción entre nichos y en una serie de criterios frecuentistas; y el *método iterativo de cubrimiento* presentado en el capítulo 3. Este primer proceso permite obtener una BRD aproximativa de tipo Mamdani, que incluya el conocimiento contenido en el conjunto de entrenamiento de un modo adecuado.
2. *El proceso genético de multisimplificación*, presentado en el capítulo 3, que lleva a cabo la tarea de selección de reglas. Como vimos en dicho capítulo, este proceso está implementado por medio de un AG con codificación binaria que emplea una función de compartición genotípica y una medida del rendimiento del SBRD codificado en el cromosoma; y permite obtener varias definiciones de la BRD formadas por las distintas combinaciones de las reglas difusas aproximativas generadas en la fase anterior que mejor nivel de cooperación presentan.
3. *Un proceso genético de ajuste*, basado en un AG con codificación real y en una función de adaptación que considera dos criterios: una medida del rendimiento del SBRD codificado en el cromosoma y una función que penaliza la no satisfacción de la propiedad de completitud. Este último proceso proporciona la BRD final, ajustando la definición previa de la forma de las funciones de pertenencia empleadas en las reglas difusas aproximativas. La BRD que presente el mejor comportamiento tras el ajuste de entre todas las generadas en el segundo proceso será la que constituirá la salida global del SBRDE.

5.1.2 El Proceso Evolutivo de Generación de Reglas Difusas Aproximativas de tipo Mamdani con Semántica Libre No Restringida

Como hemos indicado, el primer SBRDE para el aprendizaje de BRDs presentado en este capítulo considera el enfoque de semántica libre no restringida para la generación de las funciones de pertenencia consideradas en las distintas reglas difusas aproximativas que componen la base. De este modo, la única restricción que se impone sobre dichas funciones de pertenencia para llevar a cabo su aprendizaje mediante el proceso evolutivo, es que mantengan su coherencia y que estén definidas dentro de un intervalo concreto, el cual se mantiene constante a lo largo de este proceso.

El método de generación de reglas difusas aproximativas constituye una extensión al modo de trabajo del método de generación de reglas lingüísticas presentado en el capítulo anterior. Considera la existencia de una partición difusa preliminar asociada a cada variable del problema y hace uso de ésta tanto para generar un conjunto inicial de reglas descriptivas en cada ejecución como para determinar los intervalos globales de definición asociados a cada conjunto difuso en cada regla difusa aproximativa.

De este modo, el método de generación de reglas difusas aproximativas trabaja ejecutando, en primer lugar, el método generador de reglas lingüísticas para obtener la mejor regla descriptiva con respecto al estado actual del conjunto de entrenamiento. Una vez que dispone de esta regla, adapta individualmente la forma de las funciones de pertenencia que maneja la misma, dentro de un intervalo global asociado a cada conjunto difuso a partir de las particiones iniciales. Para ello, emplea una EE-(1 + 1) guiada por una función de adaptación multicriterio. En las tres secciones siguientes, presentaremos los aspectos principales del AE considerado: el *esquema de representación*, el *proceso de mutación* y la *función de adaptación*.

5.1.2.1 El esquema de representación

Puesto que el individuo empleado en la EE codifica una regla individual y, más concretamente, la información codificada está compuesta por las definiciones de las funciones de pertenencia triangulares de los conjuntos difusos empleados en la misma, haremos uso de la representación en forma de tupla con tres valores introducida en capítulos anteriores. De este modo, el esquema de representación estará basado en vectores reales de dimensión $3 \cdot (n + 1)$, donde n es el número de variables de entrada del sistema, y cada tres valores consecutivos especificarán la forma del conjunto difuso que toma como valor la variable en cuestión en la regla difusa aproximativa.

5.1.2.2 El proceso de mutación

Será necesario llevar a cabo dos cambios sobre el proceso de mutación que habitualmente ponen en práctica las EEs-(1 + 1), presentado en la sección 2.1.2.1 del capítulo 2, para poder aplicar un AE de este tipo en el aprendizaje de las funciones de pertenencia de las reglas difusas aproximativas en el método de generación. Estos cambios son los siguientes:

1. *Definición de distintos tamaños de paso:* Puesto que en la EE-(1 + 1), la fuerza de la mutación efectuada depende directamente del valor del parámetro σ que determina la desviación típica de una variable z distribuida según una normal, nos encontramos con que en nuestro problema concreto el valor del tamaño de paso no puede ser único. Este hecho se debe a que las diferentes funciones de pertenencia codificadas en el individuo están definidas sobre universos de discurso distintos y requieren, por tanto, mutaciones de distinto orden, ya que el empleo de una mutación del mismo rango en todas las componentes del vector de valores reales redundaría en un peor comportamiento del AE.

Para solucionar este problema, basta con considerar un tamaño de paso $\sigma_i = \sigma \cdot s_i$ distinto para cada componente, tal y como se indica en [BS95, Bäck96]. De este modo, los valores σ_i se obtienen a partir de una serie de modificadores fijos s_i asociados a cada componente del individuo y el único factor que se adapta durante el proceso de mutación sigue siendo el valor del parámetro global σ .

2. *Adaptación incremental de los parámetros individuales:* Normalmente, los componentes del vector de valores reales asociado al individuo padre no están relacionados entre sí y la EE puede adaptar todos ellos al mismo tiempo. Desafortunadamente, en nuestro caso sí existe una relación entre ellos. Cada tres parámetros consecutivos (x_0, x_1, x_2) definen una función de pertenencia concreta y han de verificar la propiedad $x_0 \leq x_1 \leq x_2$ para que el conjunto difuso codificado sea coherente. Puesto que únicamente el intervalo global de definición de cada conjunto difuso es conocido a priori, no es posible adaptar todos los parámetros al mismo tiempo, ya que el intervalo de variación de cada parámetro depende del valor actual de sus vecinos. La solución en este caso es llevar a cabo una adaptación incremental de los parámetros individuales, como describimos a continuación.

Sea $C_k = (x_0, x_1, x_2)$, $k = 1, \dots, n + 1$, el conjunto difuso que está siendo adaptado en el momento actual y sea $[C_k^i, C_k^d]$ su intervalo global de definición. El proceso de adaptación incremental se basa en generar el conjunto difuso $C'_k = (x'_0, x'_1, x'_2)$, adaptando en primer lugar el valor asociado al punto modal x_1 , obteniendo así el valor mutado x'_1 definido en el intervalo $[x_0, x_2]$, y mutando posteriormente los valores asociados a los puntos izquierdo y derecho x_0 y x_2 , obteniendo los valores x'_0 y x'_2 definidos respectivamente en los intervalos $[C_k^i, x'_1]$ y $[x'_1, C_k^d]$. Puede observarse claramente que la aplicación progresiva de este proceso permite obtener conjuntos difusos definidos libremente en el intervalo global especificado, como corresponde a una generación basada en una semántica libre no restringida.

Como ya hemos comentado anteriormente, vamos a hacer uso de una serie de particiones difusas preliminares de los espacios de entrada y salida para especificar los intervalos globales de definición asociados a cada conjunto difuso individual. Para ello, emplearemos la misma filosofía que en el caso de la definición de los intervalos de tamaño variable en los procesos evolutivos de ajuste presentada en la sección 3.4.1 del capítulo 3 (de hecho, como puede observarse, la forma de trabajo comentada es idéntica a aquella). Así, si el conjunto difuso existente en la partición, C_k , está definido inicialmente por la tripleta de valores (a, b, c) , el intervalo global de variación considerado en la EE será el siguiente:

$$[C_k^i, C_k^d] = \left[a - \frac{b - a}{2}, c + \frac{c - b}{2} \right]$$

Por otro lado, el valor del parámetro $s(x_i)$ que determina los tamaños de paso particulares, $\sigma_i = \sigma \cdot s(x_i)$, se calcula cada vez que la componente x_i va a ser mutada. Sea C_k el conjunto difuso cuyos valores están siendo mutados, cuando $i = 1$, es decir, en el momento en el que estamos adaptando el valor del punto modal, $s(x_1)$ toma el valor $\frac{\text{Min}(x_1 - x_0, x_2 - x_1)}{2}$. En los dos casos restantes, $i = 0$ e $i = 2$, los valores respectivos son $s(x_0) = \frac{\text{Min}(x_0 - C_k^i, x'_1 - x_0)}{2}$ y $s(x_2) = \frac{\text{Min}(x_2 - x'_1, C_k^d - x_2)}{2}$.

Con los valores anteriores de $s(\cdot)$ se asegura la obtención de un número elevado de valores normales z_i que produzcan una mutación con éxito (es decir, aquellos en los que el

valor $x'_i = x_i + z_i$ correspondiente, con $z_i \sim N(0, \sigma_i^2)$, pertenece al intervalo de definición asociado a x_i en la ejecución actual) en la primera ejecución de la EE en la que $\sigma = 1$. Obviamente, en el caso en que el valor mutado se salga del intervalo correspondiente, la componente en cuestión toma el valor del extremo del intervalo por el que se ha salido.

El algoritmo que presentamos a continuación resume la aplicación del proceso de adaptación introducido sobre una función de pertenencia codificada en el individuo padre. Sea $C_k = (x_0, x_1, x_2)$ el conjunto difuso que está siendo adaptado, los pasos a llevar a cabo son los siguientes:

1. Calcular el tamaño de paso asociado al punto modal, $s(x_1) \leftarrow \frac{\text{Min}\{x_1-x_0, x_2-x_1\}}{2}$.
2. Generar un valor $z_1 \sim N(0, \sigma_1^2)$ y calcular el valor mutado x'_1 del modo siguiente:

$$x'_1 \leftarrow \begin{cases} x_1 + z_1, & \text{si } x_1 + z_1 \in [x_0, x_2] \\ x_0, & \text{si } x_1 + z_1 < x_0 \\ x_2, & \text{si } x_1 + z_1 > x_2 \end{cases}$$

3. Mutar los puntos restantes:

(a) $s(x_0) \leftarrow \frac{\text{Min}\{x_0-C_k^i, x'_1-x_0\}}{2}$
 Generar un valor $z_0 \sim N(0, \sigma_0^2)$ y calcular x'_0 :

$$x'_0 \leftarrow \begin{cases} x_0 + z_0, & \text{si } x_0 + z_0 \in [C_k^i, x'_1] \\ C_k^i, & \text{si } x_0 + z_0 < C_k^i \\ x'_1, & \text{si } x_0 + z_0 > x'_1 \end{cases}$$

(b) $s(x_2) \leftarrow \frac{\text{Min}\{x_2-x'_1, C_k^d-x_2\}}{2}$
 Generar un valor $z_2 \sim N(0, \sigma_2^2)$ y calcular x'_2 :

$$x'_2 \leftarrow \begin{cases} x_2 + z_2, & \text{si } x_2 + z_2 \in [x'_1, C_k^d] \\ x'_1, & \text{si } x_2 + z_2 < x'_1 \\ C_k^d, & \text{si } x_2 + z_2 > C_k^d \end{cases}$$

5.1.2.3 La función de adaptación

La función de adaptación empleada por la EE está basada en la combinación de cuatro criterios distintos, los tres criterios frecuentistas considerados en la función de selección de reglas lingüísticas y el criterio de baja interacción entre nichos, todos ellos presentados en el capítulo 3 de esta memoria. Como comentábamos en dicho capítulo, la combinación de un conjunto de criterios del primer tipo con este último permite afrontar de un modo adecuado la resolución de parte del problema de la cooperación entre reglas en la primera etapa del aprendizaje además de generar un BRD preliminar con buen comportamiento que verifique las propiedades de completitud, cubrimiento y consistencia.

De este modo, los cuatro criterios considerados son los siguientes:

- Alto valor de frecuencia: $\Psi_{E_p}(R_i)$.
- Alto grado de cubrimiento medio sobre los ejemplos positivos: $G_\omega(R_i)$.
- Penalización debida a la no satisfacción de la propiedad de k -consistencia: $g_n^-(R_i)$.
- Baja interacción entre nichos: $BIEN(R_i)$

y se combinan para formar la función de adaptación del siguiente modo:

$$F(R_i) = \Psi_{E_p}(R_i) \cdot G_\omega(R_i) \cdot g_n^-(R_i) \cdot BIEN(R_i)$$

De nuevo, cuanto mayor sea el valor obtenido por la regla difusa aproximativa codificada en el individuo, mejor adaptada estará dicha regla.

Finalmente, realizaremos un pequeño comentario con respecto al empleo de un criterio de nichos en la EE considerada. Como indicábamos en el capítulo 2 de esta memoria, los esquemas de nichos han sido pensados para ser aplicados en los AGs de modo que provoquen la compartición de las recompensas asociadas a los individuos de la población que estén situados dentro del mismo nicho. De este modo, el hacer uso de un esquema de este tipo en un AE basado en el manejo de dos únicos individuos por generación, parece una tarea complicada. El problema está resuelto directamente por la propia filosofía del criterio considerado, la cual está basada, como vimos en el capítulo 3, en compartir la recompensa entre el individuo actual y los obtenidos anteriormente en las distintas ejecuciones del método de generación, por lo que el manejo del criterio de baja interacción entre nichos en la función de adaptación de una EE-(1 + 1) puede ser llevado a cabo sin ningún tipo de inconveniente.

5.1.3 El Proceso Genético de Ajuste de Bases de Reglas Difusas Aproximativas

En este caso, la tercera etapa del SBRDE propuesto estará compuesta directamente por el proceso genético de ajuste presentado en [HLV95], ya que dicho proceso fue diseñado específicamente para trabajar con BRDs aproximativas de tipo Mamdani. Para una descripción de dicho proceso véase la sección 2.4.1.2 del capítulo 2.

El proceso comentado responde directamente a todos los aspectos exigidos por la metodología evolutiva para esta tercera etapa del aprendizaje, salvo en el caso de la función de adaptación empleada, la cual no preserva la completitud de la BRD generada en las dos fases anteriores. Para solventar este problema, haremos uso de la misma función que consideramos en el proceso genético básico de simplificación y que empleábamos en el proceso genético de ajuste de BCs descriptivas presentado en el capítulo anterior. Dicha función es la siguiente:

$$F(C_j) = \begin{cases} \frac{1}{2 \cdot |E_p|} \sum_{e_l \in E_p} (ey^l - S(ex^l))^2, & \text{si } GCCE(R(C_j), E_p) \geq \tau \\ \frac{1}{2} \sum_{e_l \in E_p} (ey^l)^2, & \text{en otro caso} \end{cases}$$

donde $GCCE(R(C_j), E_p)$ es el grado de completitud de la BRD codificada en el cromosoma, $R(C_j)$, sobre el conjunto de ejemplos E_p .

5.1.4 Aplicación del Sistema Basado en Reglas Difusas Evolutivo Propuesto en el Modelado Difuso de Algunas Funciones Tridimensionales

En esta sección, consideramos de nuevo las cuatro funciones tridimensionales del Apéndice A para afrontar, en este caso, el problema de su modelado difuso. En la experimentación realizada, hemos considerado los mismos valores que empleábamos en el capítulo anterior para los distintos parámetros del SBRDE:

1. *Proceso Evolutivo de Generación:*

- Valor de los parámetros considerados en la función de adaptación: $\epsilon = 1.5$, $\omega = 0.05$ y $k = 0.1$.
- Número de iteraciones sin obtener mejora antes de finalizar la ejecución de la EE-(1 + 1): 100.
- Valor del parámetro c de la regla de éxito $\frac{1}{5}$ de Rechenberg (véase la sección 2.1.2.1 del capítulo 2): 0.9.

2. *Proceso Genético de Multisimplificación:*

- Número de soluciones a generar en el proceso genético de multisimplificación: 3.
- Valor del parámetro τ : 0.1.
- Valor de los parámetros de radio de nicho y de potencia: $r = 2.5$ por ciento del número de reglas de la base generada en la primera fase y $\beta = 0.5$.
- Tamaño de la población: $N = 61$.
- Valores de las probabilidades de cruce y mutación: $P_c = 0.6$ y $P_m = 0.1$ (por individuo).
- Número de generaciones: 500.

3. *Proceso Genético de Ajuste:*

- Valor del parámetro τ : 0.1.

- Tamaño de la población: $N = 61$.
- Valores de las probabilidades de cruce y mutación: $P_c = 0.6$ y $P_m = 0.1$ (por individuo).
- Número de generaciones: 1000.
- Valores asociados al operador de cruce max-min-aritmético y a la mutación no uniforme (véase la sección 2.1.1.3 del capítulo 2): $a = 0.35$ y $b = 5$.

Los resultados obtenidos en la aplicación del SBRDE propuesto al modelado difuso de las cuatro funciones se muestran en la tabla 5.1.

Tabla 5.1. Resultados obtenidos por el SBRDE propuesto en el modelado difuso de las funciones F_1 , F_2 , F_3 y F_4

| Función | Generación | | | Multisimplificación | | | Ajuste | |
|---------|------------|-------------|-------------|---------------------|-------------|-------------|-------------|-------------|
| | #R | EC_{entr} | EC_{prue} | #R | EC_{entr} | EC_{prue} | EC_{entr} | EC_{prue} |
| F_1 | 128 | 4.104345 | 3.552771 | 74 | 2.549414 | 2.062016 | 1.451833 | 1.134147 |
| | | | | 76 | 2.608678 | 2.181019 | 1.394737 | 1.039955 |
| | | | | 79 | 2.564189 | 2.097277 | 1.429367 | 0.911785 |
| F_2 | 132 | 0.191232 | 0.105817 | 83 | 0.097098 | 0.129513 | 0.061211 | 0.103495 |
| | | | | 86 | 0.094095 | 0.092310 | 0.056008 | 0.055712 |
| | | | | 88 | 0.095124 | 0.099825 | 0.062324 | 0.060518 |
| F_3 | 116 | 85249.51 | 69674.67 | 63 | 43776.73 | 37187.83 | 29858.72 | 30607.89 |
| | | | | 65 | 47023.69 | 34197.37 | 31470.95 | 25310.40 |
| | | | | 64 | 45987.86 | 36358.50 | 31600.67 | 27324.93 |
| F_4 | 368 | 0.287001 | 0.342191 | 256 | 0.190038 | 0.222112 | 0.177740 | 0.211421 |
| | | | | 253 | 0.183342 | 0.211726 | 0.169827 | 0.196127 |
| | | | | 246 | 0.185364 | 0.229173 | 0.172509 | 0.212415 |

De nuevo, vamos a realizar un pequeño análisis del comportamiento del proceso genético de multisimplificación. En este caso, podemos observar que los resultados son aún mejores que los obtenidos en los experimentos realizados en el capítulo anterior. En las cuatro funciones, el proceso trabaja de la mejor forma posible ya que, en todos los casos, las definiciones de BRDs obtenidas a posteriori mejoran en su nivel de generalización a la original. Además, en tres de ellas, F_1 , F_2 y F_3 , los dos modelos difusos obtenidos a partir de la segunda y tercera definición de la BRD presentan mejores resultados que el obtenido a partir de la primera en la aproximación del conjunto de prueba.

Puesto que hemos empleado la misma semilla aleatoria considerada en los experimentos realizados en el capítulo 4, podemos comparar aquellos resultados con los presentados en esta sección. De este modo, cabe destacar el hecho de que los modelos difusos obtenidos para las funciones simples F_1 y F_2 , presentan un comportamiento mucho peor que los

modelos cualitativos generados a partir del SBRDE para el aprendizaje de BCs descriptivas presentado en el capítulo anterior. Por el contrario, el rendimiento de los modelos difusos obtenidos para las funciones complejas F_3 y F_4 es muy superior al de los modelos cualitativos generados en dicho capítulo. Analizaremos más detenidamente estos hechos en la sección 5.3 de este mismo capítulo.

5.2 Un Sistema Basado en Reglas Difusas Evolutivo Según el Enfoque de Semántica Libre Restringida

En la segunda sección de este capítulo presentamos el segundo SBRDE diseñado para el aprendizaje de BRDs aproximativas de tipo Mamdani, que en este caso sigue el enfoque de semántica libre restringida. La estructura es muy similar a la empleada en la sección anterior, es decir, introduciremos en primer lugar las tres etapas que componen el sistema, presentaremos después la composición concreta de la primera fase y finalizaremos mostrando los resultados obtenidos en la aplicación de éste al modelado difuso de las cuatro funciones consideradas.

5.2.1 Estructura del Sistema Basado en Reglas Difusas Evolutivo en Varias Etapas

La estructura en tres etapas de este segundo SBRDE para el diseño de SBRDs aproximativos de tipo Mamdani es la que se muestra a continuación. Como puede observarse, la única diferencia entre ambos la constituye la composición de la primera etapa y, más concretamente, la del método de generación de reglas difusas aproximativas.

1. *Un proceso evolutivo de generación de reglas difusas aproximativas de tipo Mamdani*, formado por un *método de generación de reglas difusas aproximativas*, compuesto por un AE resultante de hibridizar un AG con una $EE - (1 + 1)$ y guiado por una función de adaptación basada en el criterio de baja interacción entre nichos y en una serie de criterios frecuentistas; y el *método iterativo de cubrimiento* presentado en el capítulo 3.
2. *El proceso genético de multisimplificación* presentado en el capítulo 3 y empleado en el SBRDE en varias etapas presentado en la sección anterior.
3. *El proceso genético de ajuste* empleado también como tercera etapa del SBRDE con semántica libre no restringida introducido en la sección anterior.

5.2.2 El Proceso Evolutivo de Generación de Reglas Difusas Aproximativas de tipo Mamdani con Semántica Libre Restringida

En esta primera fase del segundo proceso evolutivo de aprendizaje presentado en este capítulo, el método de generación de reglas difusas aproximativas está basado en el enfoque de semántica libre restringida. Como hemos comentado, dicho método se implementa por medio de un AG con codificación real, en el que cada cromosoma representa una regla difusa individual, con la característica especial de incorporar una EE como un operador genético más, para ajustar localmente las mejores reglas difusas aproximativas obtenidas en cada generación. Dedicaremos las secciones siguientes a presentar los distintos componentes del AE considerado.

5.2.2.1 El esquema de representación

Siguiendo la filosofía del modelo evolutivo presentado en esta memoria, el esquema de representación empleado en este AE trata de incorporar cuanta información sea posible, con el propósito de obtener mejores soluciones. Debido a esta razón, cada cromosoma contendrá dos tipos de información distintos: uno referente a los conjuntos difusos de las particiones difusas iniciales que toman como valor cada una de las variables que componen la regla difusa codificada en el mismo (recordemos que el enfoque de generación de semántica libre restringida exige la existencia de unas particiones difusas preliminares) y otro relacionado con la forma concreta que toman dichos conjuntos difusos en la regla actual. Podemos considerar así que cada cromosoma C está formado por dos partes, C_1 y C_2 , relacionadas, respectivamente, con la estructura general de las reglas y las restricciones asociadas a la semántica de las mismas (C_1); y con la forma concreta de los conjuntos difusos considerados en cada una de ellas (C_2).

Para representar el primer tipo de información se empleará un esquema de codificación k -ario puesto que, a fin de cuentas, lo que se codifica es el conjunto difuso concreto de la partición difusa asociada que la variable en cuestión toma como valor en la regla. Así, al conjunto de valores posibles para la variable X_i , $T(X_i) = \{L_1(X_i), \dots, L_{N_i}(X_i)\}$, donde N_i representa el número de conjuntos difusos existentes en la partición difusa asociada a X_i , se le hace corresponder un conjunto ordenado $T'(X_i) = \{1, \dots, N_i\}$ con objeto de definir el esquema de representación asociado a la primera parte del cromosoma.

En cambio, puesto que el segundo tipo de información está relacionado con la forma de los conjuntos difusos manejados en la regla, para la segunda parte del cromosoma consideraremos una codificación real. En concreto, haremos uso de la codificación de los conjuntos difusos triangulares en forma de tupla con tres valores reales, ya empleada en distintos procesos evolutivos desarrollados en esta memoria, con lo que la forma que presenta un conjunto difuso obtenido a partir del conjunto difuso primario $L_i(X_j)$ en una regla concreta se codifica mediante la tupla $(a_{L_i(X_j)}, b_{L_i(X_j)}, c_{L_i(X_j)})$.

Por ejemplo, una regla difusa aproximativa que toma como base la siguiente

$$\text{Si } X_1 \text{ es } L_{i_1}(X_1) \text{ y } \dots \text{ y } X_n \text{ es } L_{i_n}(X_n) \text{ entonces } Y \text{ es } L_{i_{n+1}}(Y)$$

se codifica en un cromosoma C con la forma:

$$\begin{aligned} C_1 &= (i_1, \dots, i_n, i_{n+1}) \\ C_2 &= (a_{L_{i_1}(X_1)}, b_{L_{i_1}(X_1)}, c_{L_{i_1}(X_1)}, \dots, a_{L_{i_n}(X_n)}, b_{L_{i_n}(X_n)}, c_{L_{i_n}(X_n)}, \\ &\quad a_{L_{i_{n+1}}(Y)}, b_{L_{i_{n+1}}(Y)}, c_{L_{i_{n+1}}(Y)}) \\ C &= C_1 C_2 \end{aligned}$$

Teniendo en cuenta el esquema de representación introducido y el hecho de que el SBRDE presentado en esta sección está basado en el enfoque de generación de semántica libre restringida, cada uno de los genes que componen la segunda parte del cromosoma, C_2 , tendrá asociado un intervalo de variación, que dependerá de los conjuntos difusos primarios que tomen como valor las variables de la regla en la primera parte del mismo, C_1 .

La tripleta de valores (a, b, c) que define la forma de cualquiera de los conjuntos difusos existentes en las particiones difusas iniciales, se empleará para especificar los intervalos de variación en los que pueden ser ajustados cada uno de esos parámetros a lo largo del proceso evolutivo del siguiente modo:

$$\begin{aligned} [a^i, a^d] &= [a - \frac{b-a}{2}, a + \frac{b-a}{2}] \\ [b^i, b^d] &= [b - \frac{b-a}{2}, b + \frac{c-b}{2}] \\ [c^i, c^d] &= [c - \frac{c-b}{2}, c + \frac{c-b}{2}] \end{aligned}$$

La figura 5.1 representa gráficamente estos intervalos.

5.2.2.2 La generación de la población inicial

Para llevar a cabo esta tarea nos plantearemos también las recomendaciones propuestas por la metodología evolutiva de aprendizaje y trataremos de aprovechar al máximo el conocimiento disponible sobre el problema para mejorar la búsqueda que desarrolla el proceso evolutivo. En este caso, aparte de las particiones difusas iniciales, podemos hacer uso de la información numérica contenida en el conjunto de entrenamiento para inicializar de forma adecuada los individuos que compondrán la primera población que manejará el AE.

El mecanismo de generación de esta primera población inicializará un tercio de los individuos a partir de los ejemplos contenidos en el conjunto de entrenamiento y otro tercio de un modo totalmente aleatorio. El tercio restante será generado de un modo intermedio,

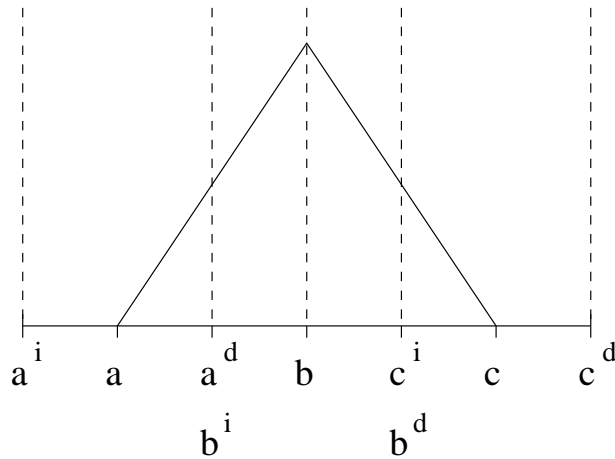


Figura 5.1. Ejemplo de los intervalos de variación considerados en el proceso de generación de reglas difusas aproximativas con semántica restringida

inicializando la primera parte del cromosoma a partir de los ejemplos disponibles y la segunda de un modo aleatorio.

Sea M el tamaño de la población que maneja el AE y sea $t = \min\{|E_p|, \frac{M}{3}\}$. Este mecanismo procede seleccionando aleatoriamente t ejemplos del conjunto de entrenamiento y llevando a cabo los tres pasos siguientes:

1. Haciendo uso de las particiones difusas iniciales, genera t individuos a partir de los t ejemplos seleccionados. Para ello, construye la primera parte del cromosoma, C_1 , asociando a cada variable el conjunto difuso primario con el que la componente del ejemplo correspondiente presenta mayor valor de pertenencia, y codifica la forma inicial de estos conjuntos difusos en la segunda parte del mismo, C_2 .
2. Genera otros t individuos inicializando C_1 de igual modo que en el paso anterior y calculando aleatoriamente los valores de C_2 dentro de los intervalos determinados por los conjuntos difusos primarios correspondientes.
3. Finalmente, genera los $M - (2 \cdot t)$ individuos restantes, calculando aleatoriamente los valores de C_1 y haciendo uso de éstos para generar de nuevo aleatoriamente los valores de C_2 en los intervalos específicos.

5.2.2.3 La función de adaptación

La función de adaptación debe llevar a cabo la misma tarea que la empleada en el SBRDE en varias etapas presentado en la sección anterior, es decir, potenciar la generación de reglas con buen comportamiento y que cooperen de un modo adecuado con las ya obtenidas. Debido a esta razón, consideraremos la misma función empleada en dicho proceso.

5.2.2.4 Los operadores genéticos

Debido a la naturaleza especial de los cromosomas con los que se trabaja en este proceso de generación, el diseño de operadores genéticos capaces de manejar de un modo adecuado la representación considerada se convierte en una tarea de gran importancia. Puesto que existe un nivel de relación muy alto entre las dos partes del cromosoma, será necesario emplear operadores que trabajen cooperativamente en C_1 y en C_2 , para hacer el mejor uso posible del esquema de representación escogido.

Puede observarse claramente que esta relación puede dar lugar a distintos problemas si no se maneja correctamente. Por ejemplo, las modificaciones efectuadas en la primera parte del cromosoma han de ser reflejadas automáticamente en la segunda, ya que no tiene sentido modificar el conjunto difuso primario y continuar trabajando con la función de pertenencia asociada al anterior. Por otro lado, es necesario llevar a cabo la recombinación de un modo correcto para obtener descendientes que representen reglas coherentes.

Teniendo en cuenta estos aspectos, emplearemos los siguientes operadores:

Mutación: Haremos uso de dos operadores de mutación distintos y cada uno de ellos actuará sobre una parte distinta del cromosoma. A continuación describimos brevemente estos operadores:

- Puesto que C_2 emplea una codificación real para codificar la forma de los conjuntos difusos existentes en la regla y ésta varía dentro de unos intervalos especificados por los conjuntos difusos primarios, parece coherente emplear el operador de mutación considerado en el proceso genético de ajuste, el cual lleva a cabo la misma tarea. De este modo, para mutar la segunda parte del cromosoma, haremos uso del operador de mutación no uniforme de Michalewicz, presentado en la sección 2.1.1.3 del capítulo 2.
- El operador de mutación seleccionado para C_1 está basado en el empleado por Thrift en el SBRDE presentado en [Thr91] e introducido en la sección 2.4.2.1 del capítulo 2. Cuando se vaya a mutar un gen de la primera parte del cromosoma, se llevará a cabo una modificación local que cambiará aleatoriamente el valor del conjunto difuso primario existente en el mismo por el inmediatamente anterior o posterior en la partición. Cuando el conjunto difuso primario a alterar sea el primero o el último de la partición, se efectuará directamente el único cambio posible.

Como hemos indicado, una mutación en C_1 provoca un cambio en C_2 . Cuando una variable difusa cambia su valor de un conjunto difuso primario a otro, la función de pertenencia asociada al mismo es automáticamente actualizada en la segunda parte del cromosoma para pasar a codificar los valores iniciales definidos en la partición primaria.

Cruce: En lo que respecta al proceso de recombinación, emplearemos también dos operadores distintos, pero su aplicación dependerá en este caso de la relación existente entre los dos padres que vayan a ser cruzados:

- Si la regla codificada en ambos individuos es la misma, es decir, toma como base los mismos conjuntos difusos primarios en todas las variables, entonces el AE ha localizado una zona del espacio de búsqueda que puede ser prometedora y que, por tanto, tiene que ser explotada adecuadamente. Esta tarea se efectúa por medio del operador de cruce max-min-aritmético que se aplica en C_2 , manteniendo los valores de C_1 en los dos descendientes obtenidos. La descripción de este operador fue presentada en la sección 2.1.1.3 del capítulo 2.
- En cambio, en el caso en el que los dos padres codifiquen una regla distinta, no tiene ningún sentido el aplicar el operador anterior, ya que provocaría la obtención de descendientes incoherentes. Este hecho se debe a que la combinación de dos funciones de pertenencia asociadas a dos conjuntos difusos primarios distintos redundaría en la obtención de un conjunto difuso que no está definido en los intervalos especificados por la partición difusa inicial. En este segundo caso está muy recomendado el uso de la información codificada en los padres para explorar el espacio de búsqueda tratando de descubrir nuevas zonas prometedoras.

De este modo, aplicaremos un operador de cruce simple sobre ambas partes del cromosoma del siguiente modo: generaremos un punto de cruce aleatorio pc en C_1 y cruzaremos ambos padres a partir de las posiciones pc (perteneciente a C_1) y $n + 1 + (3 \cdot pc)$ (contenida en C_2). El operador de cruce simple se aplica de este modo en ambas partes del cromosoma, C_1 y C_2 , dando lugar a la obtención de dos descendientes coherentes.

Mostraremos un ejemplo para clarificar la aplicación del operador de cruce simple. Sean

$$C_t = (c_1, \dots, c_{pc}, c_{pc+1}, \dots, c_{n+1}, a_{c_1}, b_{c_1}, c_{c_1}, \dots, a_{c_{pc}}, b_{c_{pc}}, c_{c_{pc}}, a_{c_{pc+1}}, b_{c_{pc+1}}, c_{c_{pc+1}}, \dots, a_{c_{n+1}}, b_{c_{n+1}}, c_{c_{n+1}})$$

$$C'_t = (c'_1, \dots, c'_{pc}, c'_{pc+1}, \dots, c'_{n+1}, a'_{c_1}, b'_{c_1}, c'_{c_1}, \dots, a'_{c_{pc}}, b'_{c_{pc}}, c'_{c_{pc}}, a'_{c_{pc+1}}, b'_{c_{pc+1}}, c'_{c_{pc+1}}, \dots, a'_{c_{n+1}}, b'_{c_{n+1}}, c'_{c_{n+1}})$$

los individuos que van a ser cruzados a partir del punto pc . Los dos descendientes obtenidos a partir del operador de cruce simple son los siguientes:

$$C_{t+1} = (c_1, \dots, c_{pc}, c'_{pc+1}, \dots, c'_{n+1}, a_{c_1}, b_{c_1}, c_{c_1}, \dots, a_{c_{pc}}, b_{c_{pc}}, c_{c_{pc}}, a'_{c_{pc+1}}, b'_{c_{pc+1}}, c'_{c_{pc+1}}, \dots, a'_{c_{n+1}}, b'_{c_{n+1}}, c'_{c_{n+1}})$$

$$C'_{t+1} = (c'_1, \dots, c'_{pc}, c_{pc+1}, \dots, c_{n+1}, a'_{c_1}, b'_{c_1}, c'_{c_1}, \dots, a'_{c_{pc}}, b'_{c_{pc}}, c'_{c_{pc}}, a_{c_{pc}}, b_{c_{pc}}, c_{c_{pc}}, \dots, a_{c_{n+1}}, b_{c_{n+1}}, c_{c_{n+1}})$$

De esta forma, el proceso global de recombinación permite al AE alcanzar un equilibrio adecuado entre la exploración y la explotación en la búsqueda que lleva a cabo sobre el espacio de soluciones. El comportamiento esperado consiste en una fase inicial, en la que se efectúan un número elevado de cruces simples y uno muy bajo de cruces max-min-aritméticos (igual a cero en la gran mayoría de los casos). En esta primera fase, el AE desarrolla una fuerte exploración, localizando las zonas prometedoras y distribuyendo individuos en ellas a lo largo de algunas generaciones. A partir de este momento comienza una nueva fase caracterizada por el incremento de la explotación de estas zonas y por el decremento de la exploración del espacio. Con ello, el número de cruces de tipo max-min-aritmético crece enormemente y la aplicación del cruce simple disminuye proporcionalmente. En la sección siguiente, mostraremos un ejemplo de este comportamiento en la práctica.

Estrategia de Evolución: El último operador genético considerado es una EE-(1 + 1). Esta técnica evolutiva de optimización ha sido escogida e integrada en el proceso genético, para llevar a cabo un ajuste local de los mejores individuos de la población (es decir, de las mejores reglas) obtenidos en cada generación. Cada vez que se efectúa una iteración del AG, la EE se aplica sobre un porcentaje $\alpha \in [0, 1]$ de los M individuos contenidos en la población actual. Concretamente, se consideran los $\alpha \cdot M$ mejores individuos de la misma, sin tener en cuenta los repetidos. De este modo, este último operador permite desarrollar de nuevo una explotación fuerte sobre las zonas prometedoras del espacio halladas en cada generación, ajustando la segunda parte, C_2 , de los cromosomas localizados en ellas.

Las bases de la EE empleada fueron presentadas en la sección 2.1.2.1 del capítulo 2. En este caso, los cambios que es necesario llevar a cabo sobre el esquema general para poder emplear este AE son más sencillos que los efectuados en el algoritmo presentado en la sección 5.1.2 de este mismo capítulo. Puesto que el SBRDE actual se basa en una semántica libre restringida, los parámetros reales que constituyen el individuo tienen asociados intervalos de variación concretos cuya intersección es vacía. Este hecho da lugar a que no sea necesario efectuar una mutación incremental de dichos parámetros, sino que baste con trabajar de la forma habitual.

Así, el único cambio a realizar será el definir distintos tamaños de paso asociados a cada variable. De nuevo, es necesario considerar esta característica ya que, con el esquema de representación en forma de tuplas con tres valores reales empleado en la segunda parte del cromosoma, cada tres valores consecutivos estarán definidos en universos de discurso distintos, por lo que el empleo de un único tamaño de paso σ es insuficiente para llevar a cabo un proceso de mutación con éxito.

En este caso, la definición es más sencilla, ya que no es necesario alterar, en cada paso de la EE, el valor de la componente s_k , encargada de adecuar el tamaño de paso global σ al empleado por cada variable concreta. Esto se debe a que, al considerar una semántica libre restringida, los intervalos de variación son fijos a lo largo de toda la ejecución.

Con ello, cada componente c_k de C_2 que pueda variar en el intervalo $[c_k^i, c_k^d]$ tendrá aso-

ciado su tamaño de paso individual σ_k con $s_k = \frac{c_k^d - c_k^i}{4}$. Así, cuando σ tome valor 1 en la primera iteración de la EE, nos aseguramos la obtención de un gran número de valores normales z_k contenidos en el intervalo $[-\frac{c_k^d - c_k^i}{4}, \frac{c_k^d - c_k^i}{4}]$. Todos estos valores, junto con los que caigan en los intervalos $[-\frac{c_k^d - c_k^i}{2}, -\frac{c_k^d - c_k^i}{4}]$ y $[\frac{c_k^d - c_k^i}{4}, \frac{c_k^d - c_k^i}{2}]$ dan como resultado una mutación con éxito (es decir, provocan que el valor $c_k + z_k$ correspondiente pertenezca al intervalo de variación asociado a c_k , $[c_k^i, c_k^d]$). En el caso en que este valor no pertenezca al intervalo en el que ha de estar definido, se asociará al valor mutado c'_k el valor del extremo del intervalo, c_k^i o c_k^d , más cercano a $c_k + z_k$.

Como puede observarse, la EE diseñada mantiene el enfoque de semántica libre restringida seguido por el proceso de generación de reglas.

A modo de resumen de los conceptos introducidos en esta sección, presentamos a continuación un algoritmo que recoge el modo de funcionamiento global del proceso de aplicación de los distintos operadores genéticos. La figura 5.2 muestra gráficamente el ámbito de aplicación de dichos operadores.

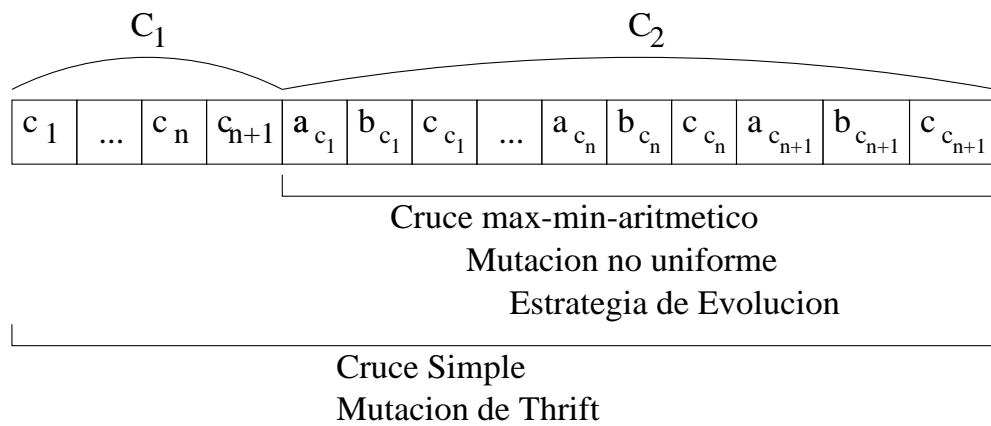


Figura 5.2. Esquema de representación y ámbito de aplicación de los operadores empleados

1. Calcular la cantidad n_c de parejas de individuos pertenecientes a $P(t)$ que van a ser cruzados a partir del valor de la probabilidad de cruce P_c .
2. Mientras ($n_c > 0$)
 - (a) Seleccionar aleatoriamente la pareja de padres que van a ser recombinados.
 - (b) Si ($C_1(\text{padre}) = C_1(\text{madre})$)
entonces mantener los valores de C_1 y efectuar un cruce max-min-aritmético en C_2 para obtener dos descendientes;
en otro caso, efectuar un cruce simple en C_1 y C_2 .
 - (c) $n_c \leftarrow n_c - 1$.

3. Calcular el número n_m de genes que van a ser mutados a partir del valor de la probabilidad de mutación P_m .
4. Mientras ($n_m > 0$)
 - (a) Seleccionar aleatoriamente el padre y el gen concreto que van a ser mutados.
 - (b) Si (el gen pertenece a C_1)
entonces llevar a cabo la mutación de Thrift en dicho gen de C_1 y actualizar los valores de la parte de cromosoma correspondiente en C_2 , para reflejar la forma del nuevo conjunto difuso que toma como valor la variable mutada en la partición inicial;
en otro caso efectuar una mutación no uniforme sobre el gen de C_2 .
 - (c) $n_m \leftarrow n_m - 1$.
5. Calcular el número $n_{ee} = \alpha \cdot M$ de individuos sobre los que se aplicará la EE y ordenar en orden decreciente la población actual, considerando únicamente los cromosomas no repetidos.
6. Mientras ($n_{ee} > 0$)
 - (a) Seleccionar el siguiente padre que va a ser ajustado localmente.
 - (b) Ejecutar la EE-(1 + 1) sobre la parte C_2 de éste.
 - (c) $n_{ee} \leftarrow n_{ee} - 1$.

Para finalizar, indicaremos que el *mecanismo de selección* considerado es el muestreo universal estocástico de Baker, introducido en el capítulo 2, junto con el modelo de selección elitista.

5.2.3 Aplicación del Sistema Basado en Reglas Difusas Evolutivo Propuesto en el Modelado Difuso de Algunas Funciones Tridimensionales

Para la experimentación realizada en la sección actual, hemos trabajado con los mismos valores de parámetros mostrados en la sección 5.1.4 de este mismo capítulo. El único cambio se produce en la primera fase del aprendizaje en la que, en esta ocasión, trabajamos con un AE híbrido entre un AG y una EE, cuyos parámetros toman los valores siguientes:

- Tamaño de la población: $N = 61$.
- Valores de las probabilidades de cruce y mutación: $P_c = 0.6$ y $P_m = 0.1$ (por individuo).

- Número de generaciones: 100.
- Valores asociados al operador de cruce max-min-aritmético y a la mutación no uniforme (véase la sección 2.1.1.3 del capítulo 2): $a = 0.35$ y $b = 5$.
- Porcentaje de individuos sobre los que se aplica la EE-(1 + 1) en cada generación: $\alpha = 0.2$
- Número de iteraciones sin obtener mejora antes de finalizar la ejecución de la EE-(1 + 1): 25.
- Valor del parámetro c de la regla de éxito $\frac{1}{5}$ de Rechenberg empleada en la EE (véase la sección 2.1.2.1 del capítulo 2): 0.9.

Los resultados obtenidos en la aplicación del segundo SBRDE presentado en el capítulo actual en el modelado difuso de las cuatro funciones son los recogidos en la tabla 5.2.

Tabla 5.2. Resultados obtenidos por el SBRDE propuesto en el modelado difuso de las funciones F_1 , F_2 , F_3 y F_4

| Función | Generación | | | Multisimplificación | | | Ajuste | |
|---------|------------|-------------|-------------|---------------------|-------------|-------------|-------------|-------------|
| | #R | EC_{entr} | EC_{prue} | #R | EC_{entr} | EC_{prue} | EC_{entr} | EC_{prue} |
| F_1 | 86 | 4.100004 | 2.273936 | 56 | 2.585311 | 1.798171 | 1.517401 | 1.100314 |
| | | | | 57 | 2.545355 | 1.710012 | 1.244670 | 0.846199 |
| | | | | 59 | 2.710044 | 1.838420 | 1.261368 | 0.953937 |
| F_2 | 92 | 0.275656 | 0.161006 | 55 | 0.160458 | 0.111845 | 0.092998 | 0.066488 |
| | | | | 58 | 0.165595 | 0.103664 | 0.094580 | 0.033741 |
| | | | | 58 | 0.167173 | 0.112521 | 0.096423 | 0.065204 |
| F_3 | 84 | 62129.07 | 48970.94 | 39 | 32928.77 | 22817.41 | 17561.98 | 15272.32 |
| | | | | 40 | 32879.96 | 22778.24 | 20047.66 | 12137.11 |
| | | | | 40 | 33962.44 | 22893.66 | 18728.33 | 10875.79 |
| F_4 | 251 | 0.289239 | 0.346819 | 170 | 0.222864 | 0.256048 | 0.202673 | 0.214119 |
| | | | | 166 | 0.223572 | 0.266738 | 0.203139 | 0.229625 |
| | | | | 167 | 0.222669 | 0.268000 | 0.203584 | 0.236712 |

En lo que respecta al comportamiento del proceso genético de multisimplificación en esta ocasión, es de nuevo muy adecuado. En las funciones F_1 , F_2 y F_3 se obtienen mejores modelos difusos a partir de las definiciones posteriores. En concreto, en la primera de las funciones, los dos modelos obtenidos a partir de la segunda y la tercera definición generadas para la BRD aproximan mejor el conjunto de entrenamiento y generalizan mejor el conjunto de prueba que el modelo obtenido a partir de la primera. En la segunda función se presenta una situación muy similar, pero sólo con respecto al nivel de generalización,

ya que los dos modelos posteriores generalizan mejor que el primero. La segunda definición consigue reducir a la mitad el error de la primera sobre el conjunto de prueba. El modelado difuso de la función F_3 presenta un comportamiento idéntico, ya que las dos definiciones posteriores generalizan mejor que la original, aunque en esta ocasión el mejor modelo global se obtiene a partir de la tercera. Finalmente, en la cuarta función, el proceso genético de multisimplificación no consigue mejorar el error asociado a la primera definición obtenida sobre ninguno de los dos conjuntos.

Por otro lado, los experimentos realizados han servido para cerciorarnos de que el comportamiento de los operadores de cruce empleados en el método de generación es el que introducíamos en la sección 5.2.2.4. Recordemos que dicho comportamiento consistía en una fase inicial, en la que se efectuaban un número muy elevado de cruces simples y uno muy bajo de cruces max-min-aritméticos, que gradualmente iba cambiando hasta llegar a una segunda fase, en la que la proporción era totalmente la inversa. La figura 5.3 muestra una representación gráfica del comportamiento de estos operadores, que corrobora las ideas preconcebidas. Para dibujar dicha figura, hemos hecho uso de los datos recogidos en la primera generación de uno de los experimentos realizados con la función F_2 , aunque el comportamiento mostrado se alcanza en casi todos ellos.

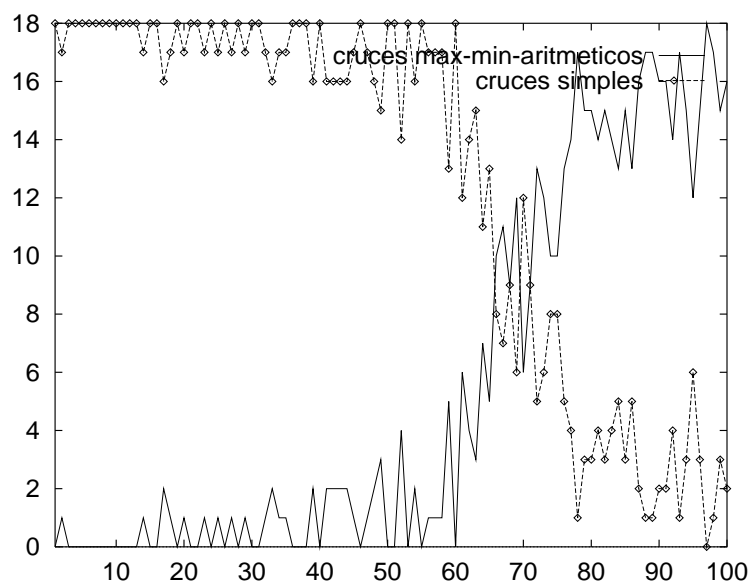


Figura 5.3. Número de cruces por generación en una ejecución de un experimento realizado

Finalmente, podemos observar que, al igual que ocurría en los experimentos presentados en la sección 5.1.4, los modelos difusos obtenidos para las funciones simples F_1 y F_2 se comportan mucho peor que los modelos cualitativos generados a partir del SBRDE para el aprendizaje de BCs descriptivas presentado en el capítulo anterior, mientras que el rendimiento de los modelos difusos obtenidos para las funciones complejas F_3 y F_4 es muy superior al de los modelos cualitativos generados en dicho capítulo. Trataremos con más profundidad este tema en la sección siguiente.

5.3 Estudio Comparativo del Comportamiento de los dos Sistemas Basados en Reglas Difusas Evolutivos Propuestos y Análisis de los Resultados Obtenidos

En esta sección, vamos a comparar los resultados obtenidos empleando los dos SBRDEs para la generación automática de BRDs aproximativas de tipo Mamdani presentados en este capítulo con los obtenidos a partir de otros dos métodos en el modelado difuso de las cuatro funciones consideradas. Los dos nuevos SBRDEs considerados posibilitan la generación de una BRD aproximativa de tipo Mamdani en dos fases, en las que, respectivamente, se hace uso de un algoritmo inductivo para generar una definición preliminar de ésta y se ajusta la semántica de las reglas existentes en la misma de un modo individual, mediante el proceso genético de ajuste empleado como tercera etapa en nuestros SBRDEs. La diferencia entre ambos radica en el algoritmo inductivo de generación de reglas empleado en la primera etapa, que en el primer caso es el de Wang y Mendel (WM) [WM92], presentado en el Apéndice B de esta memoria, por lo que la primera definición de la BC obtenida es descriptiva y no aproximativa (se convertirá en aproximativa tras el proceso de ajuste); y en el segundo caso es el *Algoritmo de Conteo Ponderado* (ACP) [BD95], presentado en el Apéndice C, el cual sí genera directamente una BRD aproximativa de tipo Mamdani.

En resumen, los cuatro SBRDEs considerados para llevar a cabo el estudio comparativo serán los siguientes:

- A1.** Un SBRDE en dos etapas compuesto por el algoritmo de WM como primera fase y el proceso genético de ajuste como segunda.
- A2.** Un SBRDE en dos etapas compuesto por el ACP y el proceso genético de ajuste comentado.
- A3.** El SBRDE en tres etapas basado en semántica libre no restringida, propuesto en la sección 5.1 de este capítulo.
- A4.** El SBRDE en tres etapas basado en semántica libre restringida, propuesto en la sección 5.2 de este capítulo.

Las ejecuciones del proceso genético de ajuste que constituye la segunda fase de los SBRDEs en dos etapas **A1** y **A2** han sido realizadas considerando los mismos valores de parámetros empleados en la tercera fase de los SBRDEs **A3** y **A4**. En lo que respecta a los dos procesos de generación inductivos, en ambos se ha hecho uso de la misma definición inicial de las particiones difusas presentada en el Apéndice A. En el primero de ellos, el de WM, se han considerado estas particiones como una definición inicial de la BD, mientras que en el segundo, se han empleado los soportes de los distintos conjuntos difusos incluidos

Tabla 5.3. Resultados obtenidos en el modelado difuso de la función F_1

| SBRDE | Generación | | | Multisimplificación | | | Ajuste | |
|-----------|------------|-------------|-------------|---------------------|-------------|-------------|-------------|-------------|
| | #R | EC_{entr} | EC_{prue} | #R | EC_{entr} | EC_{prue} | EC_{entr} | EC_{prue} |
| A1 | 49 | 2.048137 | 2.255928 | | | | 0.453033 | 0.454395 |
| A2 | 49 | 3.156302 | 2.403974 | | | | 1.133428 | 0.802255 |
| A3 | 128 | 4.104345 | 3.552771 | 79 | 2.564189 | 2.097277 | 1.429367 | 0.911785 |
| A4 | 86 | 4.100004 | 2.273936 | 57 | 2.545355 | 1.710012 | 1.244670 | 0.846199 |

Tabla 5.4. Resultados obtenidos en el modelado difuso de la función F_2

| SBRDE | Generación | | | Multisimplificación | | | Ajuste | |
|-----------|------------|-------------|-------------|---------------------|-------------|-------------|-------------|-------------|
| | #R | EC_{entr} | EC_{prue} | #R | EC_{entr} | EC_{prue} | EC_{entr} | EC_{prue} |
| A1 | 49 | 0.194386 | 0.044466 | | | | 0.056292 | 0.021056 |
| A2 | 49 | 0.288179 | 0.138665 | | | | 0.174549 | 0.079568 |
| A3 | 132 | 0.191232 | 0.105817 | 86 | 0.094095 | 0.092310 | 0.056008 | 0.055712 |
| A4 | 92 | 0.275656 | 0.161006 | 58 | 0.165595 | 0.103664 | 0.094580 | 0.033741 |

en ellas para definir los intervalos de actuación necesarios para ejecutar el ACP (véase el Apéndice C). El valor del parámetro ϕ considerado para este último proceso ha sido 0.5 en todos los casos.

Las tablas 5.3 a 5.6 recogen los resultados obtenidos a partir de los cuatro SBRDEs en el modelado difuso. En este caso, dichos resultados requieren un análisis más en profundidad, debido a que las características presentadas por los distintos SBRDEs comparados hacen que cada uno de ellos sea más adecuado para un tipo de problema.

Podemos observar, en primer lugar, como los cuatro procesos evolutivos de aprendizaje de BRDs aproximativas de tipo Mamdani obtienen peores resultados que los que permitían diseñar SBRDs descriptivos del mismo tipo, presentados en el capítulo anterior, en el modelado de las funciones F_1 y F_2 . Este comportamiento se debe a la sencillez de dichas funciones, aspecto que hace que sea más adecuado afrontar su modelado mediante un SBRD descriptivo. Como ya comentábamos en el capítulo 1, estos últimos sistemas se adecuan mejor al modelado de funciones muy suaves debido a que, al considerar unas particiones difusas globales que definen la semántica de todos los términos empleados en las reglas lingüísticas de la BC, el nivel de interacción entre los conjuntos difusos existentes en dichas particiones se mantiene constante para todos ellos. Este hecho beneficia el modelado de superficies suaves con pocos cambios. De hecho, puede observarse que el SBRDE aproximativo que mejor comportamiento presenta en las dos primeras funciones es el notado como **A1**, el cual estaba compuesto por un proceso de generación de reglas descriptivas de tipo Mamdani.

En cambio, el nivel de interacción existente entre los conjuntos difusos incluidos en reglas vecinas que forman las BRDs aproximativas es variable y depende de cada pareja de reglas concreta. Aunque esta característica es deseable para funciones muy complejas,

Tabla 5.5. Resultados obtenidos en el modelado difuso de la función F_3

| SBRDE | #R | Generación | | Multisimplificación | | | Ajuste | |
|-----------|-----|-------------|-------------|---------------------|-------------|-------------|-------------|-------------|
| | | EC_{entr} | EC_{prue} | #R | EC_{entr} | EC_{prue} | EC_{entr} | EC_{prue} |
| A1 | 49 | 172530.0 | 206244.0 | | | | 56709.97 | 47528.77 |
| A2 | 49 | 86748.00 | 81319.68 | | | | 56626.56 | 52188.98 |
| A3 | 116 | 85249.51 | 69674.67 | 65 | 47023.69 | 34197.37 | 31470.95 | 25310.40 |
| A4 | 84 | 62129.07 | 48970.94 | 40 | 33962.44 | 22893.66 | 18728.33 | 10875.79 |

Tabla 5.6. Resultados obtenidos en el modelado difuso de la función F_4

| SBRDE | #R | Generación | | Multisimplificación | | | Ajuste | |
|-----------|-----|-------------|-------------|---------------------|-------------|-------------|-------------|-------------|
| | | EC_{entr} | EC_{prue} | #R | EC_{entr} | EC_{prue} | EC_{entr} | EC_{prue} |
| A1 | 49 | 1.778353 | 2.049079 | | | | 1.018829 | 1.203226 |
| A2 | 49 | 0.494001 | 0.536226 | | | | 0.456672 | 0.481369 |
| A3 | 368 | 0.287001 | 0.342191 | 253 | 0.183342 | 0.211726 | 0.169827 | 0.196127 |
| A4 | 251 | 0.289239 | 0.346819 | 170 | 0.222864 | 0.256048 | 0.202673 | 0.214119 |

en las que pequeños cambios en las entradas tienen asociados grandes cambios en la salida, resulta perjudicial para el modelado de funciones sencillas, en las que un nivel de interacción fijo y constante para todas las reglas da lugar a la obtención de mejores resultados. Este efecto será tanto más perjudicial cuantas más reglas contenga la BRD ya que, en ese caso, los niveles de interacción serán más variados y numerosos, como puede observarse en los resultados obtenidos por los modelados difusos en las funciones comentadas.

Centrándonos ya en las funciones más complejas, como la F_3 , la cual es fuertemente multimodal en una zona del espacio y presenta una complejidad muy baja en otra zona muy amplia del mismo, y la F_4 que, como se puede observar en su representación gráfica, es una función fuertemente multimodal con cambios muy bruscos en todo su espacio de definición, los resultados obtenidos por los SBRDEs propuestos en este capítulo mejoran ampliamente los asociados a los otros dos SBRDEs basados en generaciones inductivas. Dentro del buen rendimiento de ambos, constatamos el hecho esperado de que cada uno de estos sistemas presenta un comportamiento mejor, dependiendo de la complejidad de la función modelada. Mientras que el SBRDE **A3** que, recordemos, estaba basado en el enfoque de semántica libre no restringida, consigue un modelado más exacto para la función altamente compleja F_4 , el segundo SBRDE propuesto, **A4**, basado en una semántica libre restringida, es el más adecuado para funciones con un nivel de complejidad intermedio como la F_3 .

En lo que respecta al número de reglas contenidas en las BRDs, los experimentos realizados permiten constatar también el hecho de que los SBRDs descriptivos presentan siempre un número de reglas superior al de los SBRDs aproximativos trabajando con los mismos problemas. Este hecho puede constatarse si se compara el número de reglas existentes en las BRDs mostradas en las tablas 5.3 a 5.6 presentadas en esta sección con

las contenidas en las BCs recogidas en las tablas 4.5 a 4.8 presentadas en la sección 4.5.2 del capítulo anterior.

Finalmente, las figuras 5.4 a 5.7 muestran las representaciones gráficas de los mejores modelos obtenidos para cada función, empleando los cuatro procesos evolutivos de aprendizaje considerados.

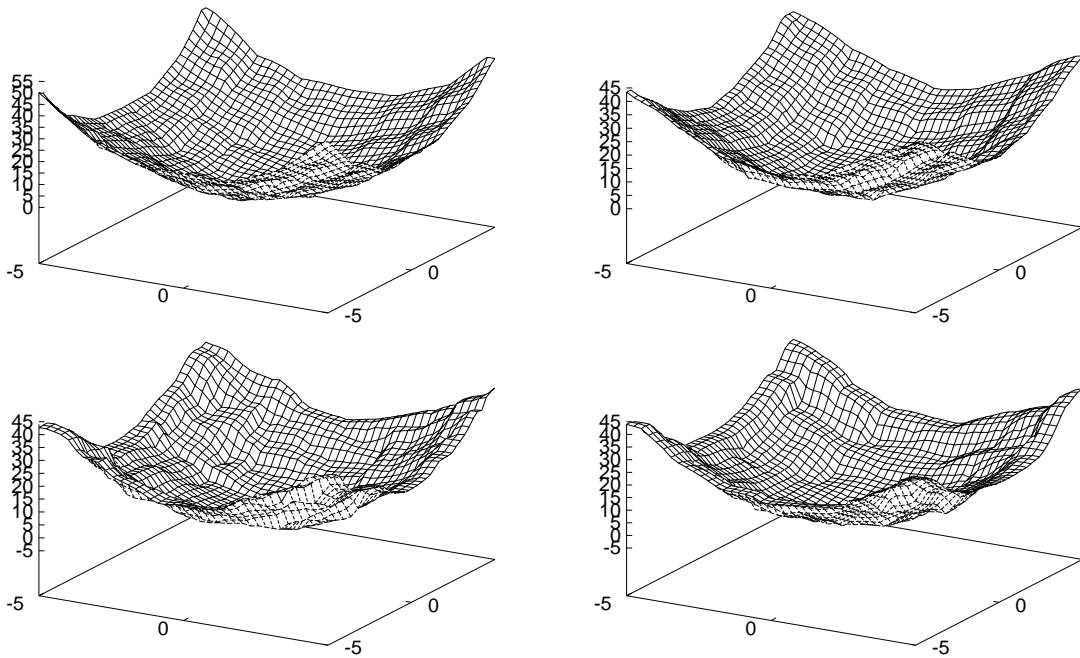


Figura 5.4. Representaciones gráficas de los cuatro modelados obtenidos para la función F_1 empleando los procesos **A1** (arriba a la izquierda), **A2** (arriba a la derecha), **A3** (abajo a la izquierda) y **A4** (abajo a la derecha)

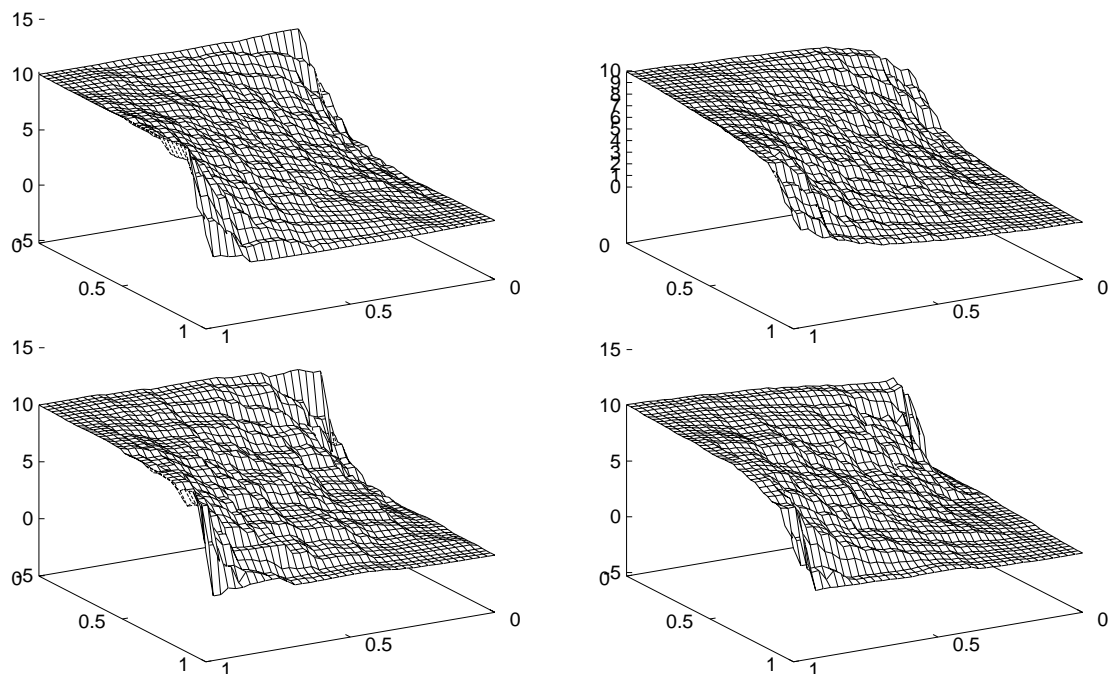


Figura 5.5. Representaciones gráficas de los cuatro modelados obtenidos para la función F_2 empleando los procesos **A1** (arriba a la izquierda), **A2** (arriba a la derecha), **A3** (abajo a la izquierda) y **A4** (abajo a la derecha)

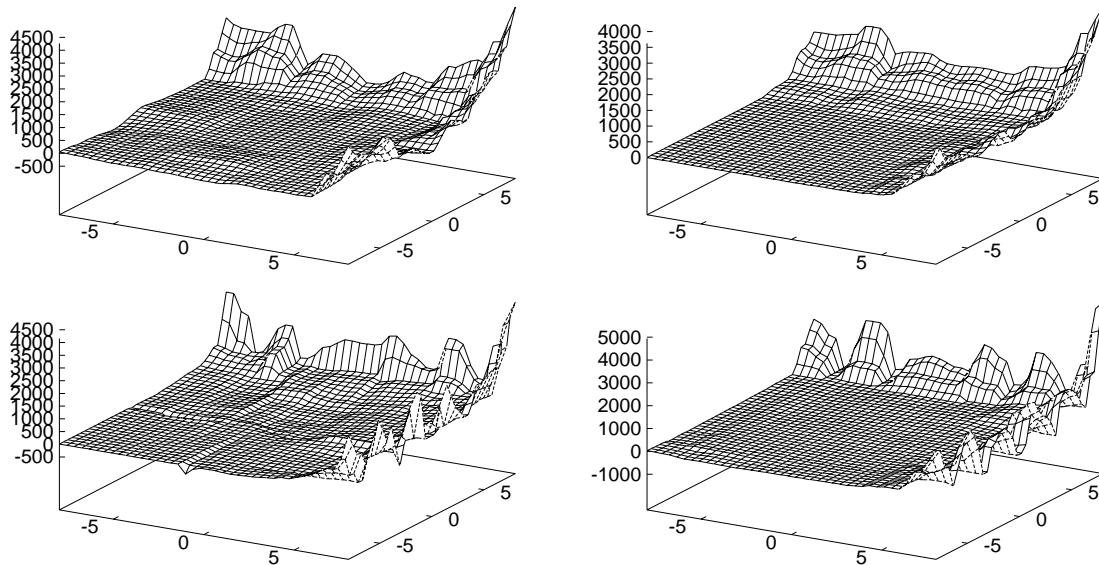


Figura 5.6. Representaciones gráficas de los cuatro modelados obtenidos para la función F_3 empleando los procesos **A1** (arriba a la izquierda), **A2** (arriba a la derecha), **A3** (abajo a la izquierda) y **A4** (abajo a la derecha)

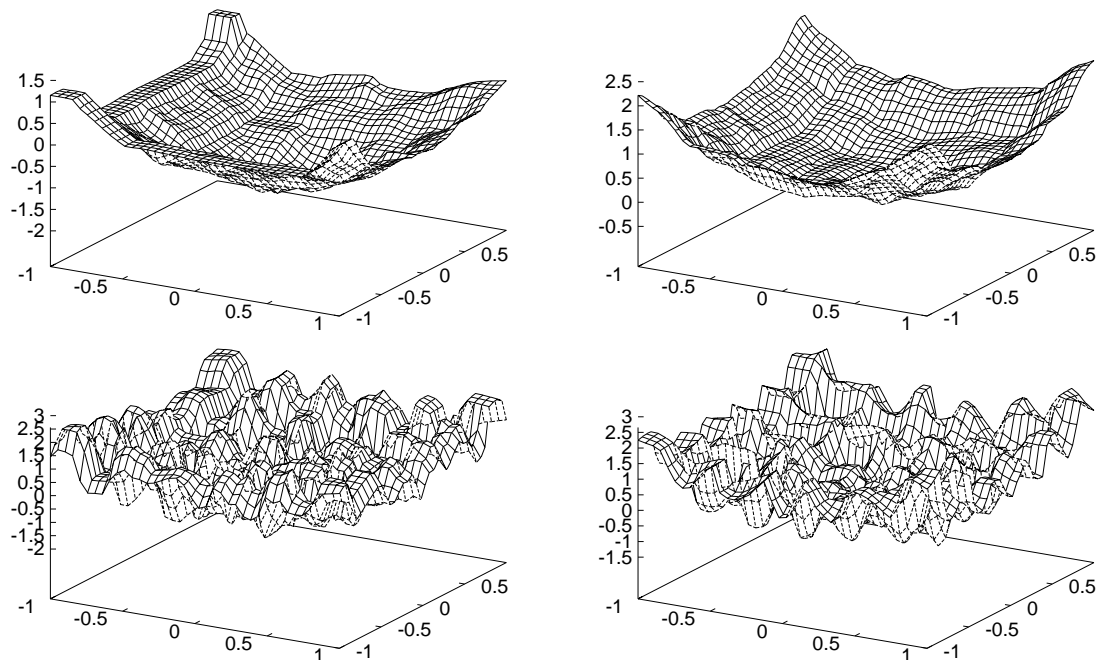


Figura 5.7. Representaciones gráficas de los cuatro modelados obtenidos para la función F_4 empleando los procesos **A1** (arriba a la izquierda), **A2** (arriba a la derecha), **A3** (abajo a la izquierda) y **A4** (abajo a la derecha)

Capítulo 6

Un Sistema Basado en Reglas Difusas Evolutivo para el Aprendizaje de Bases de Conocimiento de tipo TSK

El SBRD de tipo TSK fue propuesto para ser diseñado a partir de ejemplos [TS85], lo que justifica el uso tan extendido que ha llegado a alcanzar. Han sido muchas las técnicas que se han empleado para generar la BC TSK desde que Takagi y Sugeno propusieron su enfoque basado en el método de mínimos cuadrados en [TS85]. Entre otras, podemos encontrar Redes Neuronales [TH91, TSKK92], técnicas de descenso de gradiente [NHW91] y métodos híbridos (reglas de aprendizaje compuestas por los métodos de mínimos cuadrados y de descenso de gradiente [Jan93a]). El uso de los AEs, tanto específicos, AGs [LT93b, LT93a, LT96, YG95] y EEs [LS95], como híbridos [LS95, Wie94], se ha extendido bastante en los últimos años.

Dedicaremos este capítulo a introducir un SBRDE para la generación de BCs de tipo TSK. Para construir dicho proceso evolutivo de aprendizaje, haremos uso de las recomendaciones presentadas en el capítulo 3 para aquellos casos en los que la BC a generar es de este tipo. Recordemos que, en este caso, no es posible estudiar el cubrimiento que la regla difusa manejada produce sobre el conjunto de ejemplos, debido a que ésta no presenta un consecuente difuso. Este hecho impide el empleo de criterios frecuentistas en la etapa de generación de reglas. Así, la función de adaptación considerada en la primera etapa del aprendizaje se puede basar en una medida local de error que permita afrontar parte del problema de cooperación entre reglas en dicha fase del aprendizaje.

Por otro lado, el SBRDE propuesto estará constituido únicamente por dos fases, debido a que la composición del proceso evolutivo de generación de reglas difusas de tipo TSK hace que la BC generada no necesite ser simplificada. En este caso, el proceso evolutivo de refinamiento será el que afronte la tarea de mejorar el nivel de cooperación existente entre las reglas generadas en la primera fase. Dicha mejora se realizará mediante la adaptación de las funciones de pertenencia contenidas en los antecedentes, así como de los parámetros

que componen los consecuentes, pero sin tener que eliminar reglas que cooperen mal con el resto.

El capítulo está formado por cuatro secciones. La primera de ellas introducirá brevemente la estructura en dos etapas del SBRDE global, mientras que las dos siguientes estarán dedicadas a cada una de ellas en particular, al proceso evolutivo de generación de reglas difusas TSK y al proceso evolutivo de refinamiento. Finalmente, la última sección se centrará en el estudio del comportamiento del SBRDE propuesto en el modelado de las cuatro funciones consideradas en los capítulos anteriores.

6.1 Estructura del Sistema Basado en Reglas Difusas Evolutivo en Varias Etapas

Como ya hemos comentado, el SBRDE para el aprendizaje de BCs de tipo TSK presentado en el capítulo actual está formado por los dos procesos siguientes:

1. *Un proceso evolutivo de generación de reglas de tipo TSK* que permite obtener una definición preliminar de una BC formada por reglas de este tipo a partir de la información numérica disponible. Este primer proceso está basado en un *método iterativo de generación de antecedentes*, que sustituye al método iterativo de cubrimiento considerado en los SBRDEs que manejaban BCs de tipo Mamdani, presentados en los capítulos anteriores; y un *método evolutivo de aprendizaje de consecuentes TSK*.

El primero de ellos genera todas las combinaciones de antecedentes posibles, de acuerdo a la partición difusa considerada para los espacios de entrada, y estudia la existencia de ejemplos en los subespacios difusos multidimensionales así obtenidos. Cada vez que se confirma la existencia de datos en un subespacio concreto, el método de generación de antecedentes aplica el método de aprendizaje de consecuentes para determinar la relación lineal parcial de entrada-salida que la regla TSK en cuestión define en el mismo. Para ello, hace uso del conjunto de ejemplos existentes en dicho subespacio. Este último método se basa en una $EE - (\mu, \lambda)$ que hace uso del esquema de representación de consecuentes de tipo TSK propuesto en la sección 3.2.1.4 del capítulo 3, la codificación angular, y que está guiada por una función de adaptación compuesta por la medida local de error propuesta en la sección 3.1.2.3 de ese mismo capítulo. La población inicial del AE se inicializa utilizando el conocimiento contenido en el conjunto de entrenamiento, siguiendo las pautas especificadas por la metodología evolutiva de aprendizaje.

2. *Un proceso evolutivo de refinamiento*, encargado de ajustar tanto las definiciones preliminares de las funciones de pertenencia de los antecedentes, como las de los

parámetros que componen los consecuentes de las reglas contenidas en la BC generada en el proceso anterior. Este proceso se implementa por medio de un AE basado en un AG con codificación real que incluye una $EE - (1 + 1)$ como un operador genético más, tal y como se hacía en el proceso de generación de reglas aproximativas de tipo Mamdani con semántica libre restringida presentado en la sección 5.2.2 del capítulo 5. El algoritmo trabaja sobre cromosomas que codifican la BC al completo y ajustan la definición preliminar obtenida con objeto de conseguir la mejor cooperación posible entre las reglas difusas que la componen. La función de adaptación considerada está compuesta por una medida global de error, más adecuada para el objetivo a alcanzar. De nuevo, se hace uso del conocimiento disponible para generar la población inicial. En este caso, se tiene en cuenta la definición preliminar de la BC TSK para esta inicialización.

Como veremos en las dos secciones siguientes, dedicadas a introducir ambos procesos, el SBRDE propuesto presenta algunos aspectos importantes a destacar:

- Por un lado, como ya hemos indicado, el método iterativo de generación de antecedentes trabaja directamente con los subespacios difusos de entrada, lo que da lugar a que no sea necesario penalizar las reglas difusas ya generadas, como se suele hacer habitualmente en los procesos evolutivos de aprendizaje basados en el enfoque AIR. Este hecho se debe a que, siguiendo esta forma de trabajo, el subespacio difuso de entrada en cuestión no volverá a ser considerado en el proceso de generación, por lo que la penalización de la zona del espacio de búsqueda asociada a dicho subespacio es total.
- Debido a este modo de trabajo, no es necesario llevar a cabo un proceso de simplificación o selección de las reglas que componen la primera definición de la BC TSK obtenida, ya que no existen reglas con un nivel de cooperación tan inadecuado con las restantes, como para dar lugar a que sean eliminadas. La medida local de error provoca que las reglas localizadas en subespacios difusos de entrada vecinos puedan cooperar de forma adecuada, por lo que el empleo del proceso evolutivo de refinamiento es suficiente para obtener una BC final con un buen nivel de cooperación entre las reglas que la componen.

6.2 El Proceso Evolutivo de Generación de Reglas de Tipo TSK

La primera fase del proceso presentado está formada por dos componentes, el *método evolutivo de aprendizaje de consecuentes TSK* y el *método iterativo de generación de*

antecedentes. Dedicaremos las dos secciones siguientes a introducir la composición de cada uno de ellos.

6.2.1 El Método Evolutivo de Aprendizaje de Consecuentes TSK

Como hemos comentado, el método evolutivo de aprendizaje de consecuentes TSK está basado en una $EE - (\mu, \lambda)$. Para los aspectos básicos relacionados con este tipo concreto de AE véase la sección 2.1.2.2 del capítulo 2. En las secciones siguientes introduciremos cada uno de los componentes que definen nuestro algoritmo.

6.2.1.1 El esquema de representación

Cada individuo codificará una definición posible de consecuente para el antecedente en cuestión. Con ello, la dimensión del vector de variables objeto está determinada por el número de variables de entrada del sistema. En el caso en que dicho sistema trabaje con ve variables de entrada, los consecuentes de las reglas contenidas en su BC vienen especificados por $ve + 1$ parámetros, como se observa en la definición general que mostramos a continuación:

$$Y = p_1 \cdot X_1 + \dots + p_{ve} \cdot X_{ve} + p_0 \quad \rightarrow \quad \vec{p} = (p_0, \dots, p_{ve})$$

El esquema de representación se basa en codificar los componentes de este vector de parámetros, \vec{p} , en el vector objeto del individuo, \vec{x} , haciendo uso de la codificación angular propuesta en la sección 3.2.1.4 del capítulo 3. La dimensión del vector objeto \vec{x} considerado será, por tanto, $n = ve + 1$.

6.2.1.2 La generación de la población inicial

La inicialización de la primera población que maneja la EE se efectúa haciendo uso del conocimiento disponible en el conjunto de entrenamiento del que se dispone, E_p . Para ello, obtendremos el conjunto de ejemplos positivos para el antecedente de la regla, E , a partir de la información numérica disponible en dicho conjunto, y calcularemos una serie de índices y un subconjunto de los ejemplos positivos a un grado alto, E_θ . Emplearemos estos elementos para generar los valores de las componentes de las variables objeto en los individuos existentes en la primera población de un modo adecuado. A continuación, presentamos las expresiones de los índices y del subconjunto comentados:

$$y_{med} = \frac{\sum_{e_l \in E} e_l y^l}{|E|} \quad ; \quad y_{min} = \min_{e_l \in E} \{e_l y^l\} \quad ; \quad y_{max} = \max_{e_l \in E} \{e_l y^l\}$$

$$h_{max} = \max_{e_l \in E} \{h^l\} \quad ; \quad E_\theta = \{e_l \in E / h^l \geq \theta \cdot h_{max}\}$$

donde los $e_l = (ex_1^l, \dots, ex_n^l, ey^l)$, $l = 1, \dots, |E|$, son los ejemplos contenidos en E ; y $h^l = T(A_1(ex_1^l), \dots, A_n(ex_n^l))$ es el grado de emparejamiento existente entre las componentes de entrada, ex^l , de cada uno de esos ejemplos y los antecedentes de la regla difusa TSK para la que se está aprendiendo el consecuente (T es una t-norma).

Como puede observarse, los tres primeros representan, respectivamente, los valores medio, mínimo y máximo de la variable de salida en los ejemplos positivos de la regla actual. El conjunto E_θ , en cambio, contiene los ejemplos más significativos de ese conjunto, es decir, aquellos que se emparejan con el antecedente de la regla en un grado lo suficientemente alto. Este grado se define como un porcentaje $\theta \in [0.5, 1]$, especificado por el diseñador del SBRD, del máximo grado de emparejamiento h_{max} existente en el conjunto de ejemplos positivos para la regla.

Así, la generación de los componentes del vector objeto \vec{x} se lleva a cabo mediante los tres pasos siguientes, que permiten obtener una primera población formada por individuos que puedan constituir buenas soluciones al problema planteado:

1. Se genera el vector objeto del primer individuo \vec{x}_1 inicializando los $n - 1$ parámetros asociados a las variables p_i ($i = 1, \dots, ve$) al valor 0, y el término independiente p_0 con el valor de la codificación angular de y_{med} , $C(y_{med})$.
2. Se generan los vectores objeto de los γ individuos siguientes, $\vec{x}_2, \dots, \vec{x}_{\gamma+1}$, con el valor $\gamma \in \{0, \dots, \mu - 1\}$ especificado por el diseñador del SBRD, inicializando los $n - 1$ parámetros p_i , $i = 1, \dots, ve$, al valor 0, y el término independiente p_0 con el valor de la codificación angular $C(u)$ de un uniforme u generado aleatoriamente en $[y_{min}, y_{max}]$.
3. Se generan los vectores objeto de los $\mu - (\gamma + 1)$ individuos restantes, $\vec{x}_{\gamma+2}, \dots, \vec{x}_\mu$, inicializando los parámetros p_i , $i = 1, \dots, ve$, con el valor de la codificación angular $C(u)$ de un uniforme u generado aleatoriamente en $(-\frac{\pi}{2}, \frac{\pi}{2})$, y el término independiente con el valor de la codificación angular del resultado de una operación efectuada sobre un elemento seleccionado aleatoriamente de E_θ , de modo que se asegure que el consecuente generado corta a la nube de puntos definida por E_θ .

Como comentamos en el capítulo 3, la codificación angular está basada en una transformación no lineal. Este hecho ha de ser tenido en cuenta a la hora de generar la población inicial del AE, para lo que se debe emplear una distribución de probabilidad no uniforme. Puesto que con valores angulares pequeños se cubren zonas grandes del espacio de búsqueda, parece interesante generar valores pequeños para los parámetros p_i en este tercer paso. Para ello, haremos uso de una función modificadora que asigna una probabilidad mayor de aparición a los ángulos pequeños de acuerdo a un parámetro q , proporcionado también por el diseñador del sistema. La función considerada es la que mostramos a continuación:

$$f : [0, 1] \times \{-1, 1\} \rightarrow \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$$

$$f(x, z) = z \cdot \frac{\pi}{2} \cdot x^q$$

De este modo, la generación de los individuos en este tercer paso se efectúa del siguiente modo:

Para $j = 1, \dots, \mu - (\gamma + 1)$

- a) Para $i = 1, \dots, ve$
 - a.1) Se genera aleatoriamente x en $[0, 1]$.
 - a.2) Se genera aleatoriamente z en $\{-1, 1\}$.
 - a.3) Se inicializa p_i a $f(x, z)$.
- b) Se genera el valor de p_0 :
 - b.1) Se selecciona aleatoriamente e_j de E_θ .
 - b.2) Se inicializa p_0 a $ey^j - \sum_{k=1}^{ve} C^{-1}(p_k) \cdot ex_k^j$, donde $C^{-1}(\beta) = \tan(\beta)$ es la inversa de C .

La filosofía de generación de individuos considerada provoca que los consecuentes inicializados corten a la nube de puntos definida por los elementos del conjunto de ejemplos más significativos, E_θ , en todos los casos. Este hecho asegura un buen comportamiento de las reglas codificadas desde el primer momento, facilitando así la obtención de buenas soluciones por parte del proceso evolutivo.

Finalmente, todos los componentes de los vectores que especifican los valores de las desviaciones típicas, $\vec{\sigma}$, y los ángulos de rotación, $\vec{\alpha}$ (cuando se consideran estos últimos en el proceso evolutivo), se inicializan respectivamente a 0.001 y a $\arctan(1)$.

6.2.1.3 La función de adaptación

El proceso evolutivo que lleva a cabo la EE está guiado por una función de adaptación compuesta únicamente por una medida local de error. Como comentábamos en el capítulo 3 de esta memoria, el empleo de una medida de este tipo en la generación de reglas de tipo TSK en la primera fase del aprendizaje permite alcanzar los dos objetivos planteados: establecer un buen proceso competitivo entre las reglas codificadas en los individuos de la población que permita obtener reglas con la mayor calidad posible y afrontar el problema de cooperación entre las reglas ya generadas para que la BC presente un buen rendimiento a nivel global.

La expresión de la función de adaptación considerada es la siguiente:

$$F(\vec{x}_i) = \sum_{e_l \in E} h^l \cdot (ey^l - S(ex^l))^2$$

donde E es el subconjunto de ejemplos existentes en el conjunto de entrenamiento E_p que están contenidos en el subespacio difuso de entrada definido por los antecedentes de la regla, $h^l = T(A_1(ex_1^l), \dots, A_n(ex_n^l))$ es el grado de emparejamiento existente entre las componentes de entrada, ex^l , de cada uno de esos ejemplos y los antecedentes de dicha regla. Por último, $S(ex^l)$ es la salida que devuelve la regla difusa TSK codificada en \vec{x}_i al recibir ex^l como entrada.

6.2.1.4 Los operadores evolutivos

El proceso de adaptación se lleva a cabo empleando los operadores habituales asociados a la $EE - (\mu, \lambda)$. Estos operadores de recombinación de padres y de mutación de los distintos componentes de los tres vectores, \vec{x} , $\vec{\sigma}$ y $\vec{\alpha}$, que forman los individuos de la población pueden ser consultados en la sección 2.1.2.2 del capítulo 2.

6.2.2 El Método Iterativo de Generación de Antecedentes

Este método se lleva a cabo por medio de los siguientes pasos:

1. Se obtiene una definición preliminar de las particiones difusas de los espacios asociados a las variables de entrada, bien a partir de conocimiento experto, o bien mediante un proceso de normalización del tipo de los introducidos en el capítulo 1, es decir, dividiendo cada universo de discurso en un número igual o distinto de partes, seleccionando un tipo de función de pertenencia y asignando un conjunto difuso concreto a cada subespacio.
2. Para cada subespacio difuso multidimensional obtenido combinando los subespacios individuales de las variables de entrada mediante el operador de conjunción y (es decir, para cada combinación de antecedentes posible):
 - (a) Se construye el subconjunto E compuesto por los ejemplos de entrenamiento $e_k \in E_p$ localizados en el subespacio en cuestión, es decir, los ejemplos positivos para el antecedente de la regla.
 - (b) Si $|E| \neq 0$, esto es, si existen ejemplos en el subespacio de entrada, se aplica el método de aprendizaje de consecuentes TSK sobre el conjunto de datos E , para determinar la relación parcial de entrada-salida existente en dicho subespacio. De este modo, el método no genera reglas para aquellas combinaciones de antecedentes que no tengan ejemplos positivos en el conjunto de entrenamiento E_p .
 - (c) Se añade la regla generada a la definición preliminar de la BC.

6.3 El Proceso Evolutivo de Refinamiento de Bases de Conocimiento de Tipo TSK

El proceso evolutivo de refinamiento es un proceso de ajuste que toma una BC de tipo TSK como entrada y ajusta las definiciones preliminares de las funciones de pertenencia y de los parámetros empleados en los antecedentes y consecuentes de las reglas que la componen. Este ajuste se efectúa de acuerdo al comportamiento global de la BC que se está adaptando en el problema a resolver, representado en forma de conjunto de entrenamiento. El proceso está compuesto por un AG que incluye una $EE - (1 + 1)$ como un operador genético más para mejorar el proceso de búsqueda, siguiendo las recomendaciones de la metodología evolutiva de aprendizaje. El AE en cuestión está guiado por una función de adaptación basada en una medida global de error.

En lo que respecta a las funciones de pertenencia que componen los antecedentes de las reglas, seguiremos la filosofía de adaptación basada en intervalos de variación variables, introducida en la sección 3.4.1 del capítulo 3. Hemos escogido esta opción en lugar de la basada en intervalos fijos debido a que, aunque los conjuntos difusos obtenidos tras el proceso evolutivo presentan un grado de interpretabilidad menor, permite un mayor grado de libertad en la adaptación lo que repercute en un mejor rendimiento del SBRD diseñado finalmente. De este modo, parece la opción más adecuada para un SBRD de tipo TSK, que tiene asociado de por sí un grado de interpretabilidad menor, debido a la estructura de regla difusa que emplea.

A continuación describiremos más detenidamente los distintos componentes del proceso evolutivo de refinamiento.

6.3.1 El esquema de representación

Cada cromosoma C que codifica una definición distinta de la BC TSK está formado por dos partes distintas, C^1 y C^2 , que representan, respectivamente, las definiciones de las funciones de pertenencia empleadas en los antecedentes y las de los parámetros asociados a los consecuentes de las distintas reglas difusas de la base.

De nuevo, trabajaremos con conjuntos difusos triangulares en las particiones difusas consideradas y haremos uso de la representación en forma de tupla con tres valores reales (a, b, c) para caracterizar cada uno de ellos. Así, representaremos cada partición difusa mediante un vector compuesto por N_i tuplas ($3 \cdot N_i$ valores reales) (a_l, b_l, c_l) , $l = 1, \dots, N_i$, donde N_i es el número de términos lingüísticos existentes en la misma que son empleados en las reglas contenidas en la BC, tal y como hacíamos en el proceso genético de ajuste de BCs descriptivas de tipo Mamdani presentado en la sección 4.4 del capítulo 4. La definición completa de todas las particiones difusas en un problema que tenga asociadas ve variables de entrada se codifica en la primera parte C^1 de cada cromosoma de la población, C_j ,

concatenando las representaciones de las definiciones parciales de las particiones difusas asociadas a cada una de ellas, como se muestra a continuación:

$$\begin{aligned} C_i^1 &= (a_{i1}, b_{i1}, c_{i1}, \dots, a_{iN_i}, b_{iN_i}, c_{iN_i}) , \\ C^1 &= C_1^1 C_2^1 \dots C_{ve}^1 \end{aligned}$$

Al considerar, como ya hemos indicado, el enfoque basado en intervalos de variación variables, cada conjunto difuso triangular existente en la BD, $D_{ij} = (a_{ij}, b_{ij}, c_{ij})$, $i = 1, \dots, ve$, $j = 1, \dots, N_i$, tendrá asociado un intervalo de variación $[D_{ij}^{min}, D_{ij}^{max}]$ en el que los tres parámetros que lo definen podrán variar libremente, siempre que la forma resultante sea coherente. Los extremos de estos intervalos se calculan antes de la ejecución del proceso evolutivo de refinamiento, haciendo uso de las particiones difusas preliminares consideradas en la fase anterior, procediendo del siguiente modo:

$$[D_{ij}^{min}, D_{ij}^{max}] = [a_{ij} - \frac{b_{ij} - a_{ij}}{2}, c_{ij} + \frac{c_{ij} - b_{ij}}{2}]$$

Con ello, el intervalo de rendimiento de cada gen de C^1 dependerá de la función de pertenencia a la que esté asociado. Cada uno de estos intervalos será el intervalo de variación del gen correspondiente, $c_t \in [c_t^i, c_t^d]$. Si $(t \bmod 3) = 1$ entonces c_t es el extremo izquierdo del soporte del conjunto difuso definido por los parámetros (c_t, c_{t+1}, c_{t+2}) y los intervalos de rendimiento asociados a cada uno de ellos son:

$$\begin{aligned} c_t \in [c_t^i, c_t^d] &= [D^{min}, c_{t+1}] \\ c_{t+1} \in [c_{t+1}^i, c_{t+1}^d] &= [c_t, c_{t+2}] \\ c_{t+2} \in [c_{t+2}^i, c_{t+2}^d] &= [c_{t+1}, D^{max}] \end{aligned}$$

donde D^{min} y D^{max} son los extremos del intervalo de variación asociado al conjunto difuso definido por la tupla (c_t, c_{t+1}, c_{t+2}) . Tal y como se comentó en la sección 3.4.1 del capítulo 3, estos valores son los únicos, de entre los que definen los intervalos de variación de los genes, que permanecen constantes a lo largo de la ejecución del AG.

En lo que respecta a la segunda parte del cromosoma, C^2 , es la encargada de codificar los parámetros de los consecuentes de cada regla difusa contenida en la definición preliminar de la BC TSK. Está formada por $T \cdot (ve + 1)$ genes, donde T es el número de reglas existentes en la BC y $ve + 1$ es el número de parámetros necesarios para definir el consecuente de cada regla difusa de tipo TSK:

$$\begin{aligned} C_i^2 &= (p_{i0}, p_{i1}, \dots, p_{i ve}), \quad i = 1, \dots, T ; \\ C^2 &= C_1^2 C_2^2 \dots C_T^2 . \end{aligned}$$

Puesto que estos parámetros se codifican haciendo uso de la codificación angular, el intervalo de variación asociado a todos ellos es el mismo, $(-\frac{\pi}{2}, \frac{\pi}{2})$.

6.3.2 La generación de la población inicial

En esta segunda fase del aprendizaje, se emplea también el conocimiento disponible para la inicialización de la primera población que manejará el AG. En este caso, de lo que se hace uso es de la definición preliminar de la BC TSK de la que se dispone. Sea M el tamaño de la población inicial, la generación de la misma se lleva a cabo mediante los tres pasos siguientes:

1. La definición preliminar de la BC recibida como entrada por el proceso se codifica directamente en el primer cromosoma de la población, C_1 .
2. Los siguientes $\frac{M}{2} - 1$ cromosomas se inicializan generando aleatoriamente los valores de los genes de la primera parte, C^1 , dentro de los intervalos correspondientes y codificando la definición preliminar de los parámetros del consecuente en C^2 .
3. Los restantes $\frac{M}{2}$ se construyen inicializando C^1 de la misma forma seguida en el paso anterior y generando los valores de los genes de C^2 añadiendo un valor aleatorio distribuido según una normal $N(0, d)$ a los valores de los genes de C^2 en los cromosomas anteriores, es decir, a los de la definición preliminar de la BC.

6.3.3 La función de adaptación

Como ya hemos comentado, la función de adaptación está basada en una medida de error global, el ECM, aplicado sobre los ejemplos contenidos en el conjunto de entrenamiento, E_p . De este modo, el valor de adaptación asociado a un individuo C_j se obtiene calculando el error entre las salidas que devuelve el SBRD TSK que emplea la definición de la BC codificada en dicho cromosoma y los valores teóricos existentes en E_p . La expresión de la función de adaptación es, por tanto, la que se muestra a continuación:

$$F(C_j) = \frac{1}{2|E_p|} \sum_{e_l \in E_p} (ey^l - S(ex^l))^2$$

con las equivalencias habituales ya presentadas en esta memoria (véase, por ejemplo, la sección 3.3.1 del capítulo 3).

6.3.4 Los operadores genéticos

El AG hace uso de tres operadores genéticos distintos con los que altera la composición de los cromosomas seleccionados para formar la nueva población en cada generación. Al igual que en el proceso evolutivo de generación de reglas difusas aproximativas con semántica libre restringida presentado en la sección 5.2.2 del capítulo 5, además de los dos operadores

habituales de cruce y mutación, se hará uso de una $EE - (1 + 1)$ para optimizar los mejores individuos generados en cada iteración. Los operadores de cruce y mutación considerados en este caso son el cruce max-min-aritmético y la mutación no uniforme, ya presentados en capítulos anteriores de esta memoria.

Respecto a la EE, es necesario combinar algunos de los aspectos señalados en el capítulo 5 para poder ponerla en práctica. Procederemos del mismo modo que en el proceso evolutivo de generación presentado en dicho capítulo, aplicando esta técnica local de optimización sobre un porcentaje $\alpha \in [0, 1]$ de los mejores individuos obtenidos en cada generación del AG, con objeto de explotar mejor las zonas del espacio en las que se localicen las mejores soluciones obtenidas hasta el momento. De nuevo, es necesario considerar un tamaño de paso distinto asociado a cada una de las variables y, en este caso, esa necesidad es aún más acusada ya que los cromosomas sobre los que se aplica la estrategia codifican dos tipos distintos de información. Procederemos del mismo modo que en casos anteriores, haciendo uso de un factor s_i asociado a cada variable y adaptando únicamente el factor global σ .

Por otro lado, puesto que la adaptación de los conjuntos difusos codificados en la primera parte del cromosoma C^1 está basada en el empleo de intervalos de variación variables, es necesario también efectuar una adaptación incremental de los parámetros individuales con objeto de que el proceso evolutivo que lleva a cabo la $EE - (1 + 1)$ no de lugar a la aparición de conjuntos difusos sin sentido. La adaptación de las funciones de pertenencia codificadas en C^1 se llevará a cabo, por tanto, del mismo modo en el que se efectuaba en el proceso evolutivo de generación de reglas difusas aproximativas con semántica libre no restringida, presentado en la sección 5.1 del capítulo 5.

Al trabajar con la segunda parte del cromosoma C^2 , en cambio, este problema no aparece. En este caso, los valores de los distintos componentes no están relacionados entre sí y la mutación que lleva a cabo la EE puede efectuarse de acuerdo a la forma habitual.

Finalmente, como *mecanismo de selección*, haremos uso del muestreo universal estocástico de Baker, así como de un modelo elitista.

6.4 Aplicación del Sistema Basado en Reglas Difusas Evolutivo Propuesto en el Modelado Difuso de Algunas Funciones Tridimensionales

En esta sección, aplicaremos en primer lugar el SBRDE para el aprendizaje de BCs de tipo TSK al modelado de las cuatro funciones consideradas a lo largo de esta memoria,

para pasar posteriormente a comparar su comportamiento con el de otro SBRDE con características similares, el propuesto por Lee y Takagi en [LT93a, LT93b].

6.4.1 Experimentos Realizados

Hemos efectuado distintos experimentos haciendo uso de dos de las cuatro combinaciones más habituales para determinar las dimensiones de los vectores $\vec{\sigma}$ y $\vec{\alpha}$ considerados en la $EE - (\mu, \lambda)$ empleada en la primera etapa del SBRDE propuesto, el proceso evolutivo de generación de reglas de tipo TSK (véase la sección 2.1.2.2 del capítulo 2), $(n_\sigma, n_\alpha) = \{(n, 0), (n, \frac{n(n-1)}{2})\}$. Asimismo, hemos hecho uso de tres valores distintos para el parámetro α , el cual determina el tanto por ciento de individuos sobre los que se aplica la $EE - (1 + 1)$ en el proceso evolutivo de refinamiento. Los valores considerados han sido $\alpha = \{0, 0.1, 0.2\}$. Para cada combinación de valores de estos parámetros hemos efectuado tres ejecuciones distintas del proceso evolutivo de aprendizaje de BCs TSK propuesto, con distintos valores iniciales para la semilla del generador de números aleatorios, para así obtener más información sobre el rendimiento del SBRDE.

Los valores asociados a los restantes parámetros se recogen a continuación, clasificados según la etapa del aprendizaje en la que se emplean:

1. Proceso evolutivo de generación:

- Número de iteraciones de la $EE - (\mu, \lambda)$: 500.
- Número de padres e hijos a generar en cada iteración: $\mu = 15, \lambda = 100$.
- Número de individuos de la población inicial que son generados en el segundo paso del procedimiento de inicialización: $\gamma = 0.2 \cdot \mu = 3$.
- Valor del parámetro que determina el umbral de pertenencia a partir del cual un ejemplo se considera significativo en el procedimiento de inicialización: $\theta = 0.7$.
- Valor del parámetro empleado en la función modificadora encargada de la generación aleatoria de valores angulares pequeños en el procedimiento de inicialización: $q = 5$.
- Mecanismo de combinación de genes considerado en cada componente del individuo $\vec{r} = (r_{\vec{x}}, r_{\vec{\sigma}}, r_{\vec{\alpha}}) = (3, 2, 0)$.
- Número de padres considerados en el proceso de recombinación en cada componente: $\vec{\zeta} = (\zeta_{\vec{x}}, \zeta_{\vec{\sigma}}, \zeta_{\vec{\alpha}}) = (\mu, \mu, 1)$.

2. Proceso evolutivo de refinamiento:

- Tamaño de la población: $N = 61$.
- Valores de las probabilidades de cruce y mutación: $P_c = 0.6$ y $P_m = 0.1$ (por individuo).

Tabla 6.1. Resultados obtenidos por el SBRDE propuesto en el modelado difuso de la función F_1

| PARAMETROS | | | | GENERACION | | REFINAMIENTO | | MEDIA | |
|------------|------------|----------|-----|---------------|---------------|---------------|---------------|-------------------|-------------------|
| n_σ | n_α | α | Ej. | EC_{entr}^g | EC_{prue}^g | EC_{entr}^r | EC_{prue}^r | EC_{entr}^{med} | EC_{prue}^{med} |
| 3 | 0 | 0 | 1 | | | 0.006592 | 0.006568 | | |
| 3 | 0 | 0 | 2 | 0.050862 | 0.051457 | 0.008108 | 0.009008 | 0.006980 | 0.007504 |
| 3 | 0 | 0 | 3 | | | 0.006240 | 0.006938 | | |
| 3 | 0 | 0.1 | 1 | | | 0.007828 | 0.007737 | | |
| 3 | 0 | 0.1 | 2 | 0.050862 | 0.051457 | 0.006627 | 0.007717 | 0.006921 | 0.007498 |
| 3 | 0 | 0.1 | 3 | | | 0.006310 | 0.007042 | | |
| 3 | 0 | 0.2 | 1 | | | 0.007268 | 0.008018 | | |
| 3 | 0 | 0.2 | 2 | 0.050862 | 0.051457 | 0.006566 | 0.007711 | 0.007150 | 0.007805 |
| 3 | 0 | 0.2 | 3 | | | 0.007618 | 0.007687 | | |
| 3 | 3 | 0 | 1 | | | 0.010610 | 0.011168 | | |
| 3 | 3 | 0 | 2 | 0.073856 | 0.056382 | 0.012539 | 0.013282 | 0.011431 | 0.011646 |
| 3 | 3 | 0 | 3 | | | 0.011145 | 0.010490 | | |
| 3 | 3 | 0.1 | 1 | | | 0.010338 | 0.010349 | | |
| 3 | 3 | 0.1 | 2 | 0.073856 | 0.056382 | 0.011458 | 0.013186 | 0.011173 | 0.011928 |
| 3 | 3 | 0.1 | 3 | | | 0.011725 | 0.012251 | | |
| 3 | 3 | 0.2 | 1 | | | 0.012412 | 0.015150 | | |
| 3 | 3 | 0.2 | 2 | 0.073856 | 0.056382 | 0.010139 | 0.010494 | 0.010868 | 0.012276 |
| 3 | 3 | 0.2 | 3 | | | 0.010055 | 0.011185 | | |

- Número de generaciones del AG: 1000.
- Valores asociados al operador de cruce max-min-aritmético y a la mutación no uniforme: $a = 0.35$ y $b = 5$.
- Número de iteraciones sin obtener mejora antes de finalizar la ejecución de la $EE - (1 + 1)$: 25.
- Valor del parámetro c de la regla de éxito $\frac{1}{5}$ de Rechenberg empleada en la EE: 0.9.
- Valor de la desviación típica empleada para generar los componentes de C_2 en el último paso del proceso de inicialización: $d = 0.001$.

Los resultados obtenidos en los distintos experimentos realizados están recogidos en las tablas 6.1 a 6.4, en las que EC_{entr}^x y EC_{prue}^x se refieren a los valores obtenidos por el SBRD TSK diseñado en la medida de ECM, sobre los conjuntos de entrenamiento y prueba, respectivamente (x puede ser igual a g y r representando los procesos evolutivos de generación y refinamiento). Todas las BCs aprendidas están compuestas por 49 reglas

Tabla 6.2. Resultados obtenidos por el SBRDE propuesto en el modelado difuso de la función F_2

| PARAMETROS | | | | GENERACION | | REFINAMIENTO | | MEDIA | |
|------------|------------|----------|-----|---------------|---------------|---------------|---------------|-------------------|-------------------|
| n_σ | n_α | α | Ej. | EC_{entr}^g | EC_{prue}^g | EC_{entr}^r | EC_{prue}^r | EC_{entr}^{med} | EC_{prue}^{med} |
| 3 | 0 | 0 | 1 | | | 0.016275 | 0.004175 | | |
| 3 | 0 | 0 | 2 | 0.029530 | 0.004223 | 0.015290 | 0.001763 | 0.015954 | 0.003002 |
| 3 | 0 | 0 | 3 | | | 0.016298 | 0.003069 | | |
| 3 | 0 | 0.1 | 1 | | | 0.009134 | 0.011710 | | |
| 3 | 0 | 0.1 | 2 | 0.029530 | 0.004223 | 0.007656 | 0.024311 | 0.007453 | 0.018500 |
| 3 | 0 | 0.1 | 3 | | | 0.005569 | 0.019480 | | |
| 3 | 0 | 0.2 | 1 | | | 0.007190 | 0.027646 | | |
| 3 | 0 | 0.2 | 2 | 0.029530 | 0.004223 | 0.004297 | 0.023401 | 0.005142 | 0.025348 |
| 3 | 0 | 0.2 | 3 | | | 0.003940 | 0.024999 | | |
| 3 | 3 | 0 | 1 | | | 0.014823 | 0.006048 | | |
| 3 | 3 | 0 | 2 | 0.044170 | 0.058803 | 0.014322 | 0.004197 | 0.012686 | 0.004308 |
| 3 | 3 | 0 | 3 | | | 0.008913 | 0.002680 | | |
| 3 | 3 | 0.1 | 1 | | | 0.015229 | 0.009054 | | |
| 3 | 3 | 0.1 | 2 | 0.044170 | 0.058803 | 0.009706 | 0.003222 | 0.011458 | 0.005823 |
| 3 | 3 | 0.1 | 3 | | | 0.009441 | 0.005193 | | |
| 3 | 3 | 0.2 | 1 | | | 0.013696 | 0.006228 | | |
| 3 | 3 | 0.2 | 2 | 0.044170 | 0.058803 | 0.008015 | 0.008940 | 0.009331 | 0.006155 |
| 3 | 3 | 0.2 | 3 | | | 0.006284 | 0.003299 | | |

(la definición inicial considerada para la BD es la recogida en el Apéndice A, basada en particionar los universos de entrada en 7 partes iguales). Los valores finales contenidos en las dos últimas columnas de cada tabla y notados por EC_{entr}^{med} y EC_{prue}^{med} respectivamente, han sido obtenidos calculando la media de las tres ejecuciones realizadas.

A la vista de los resultados obtenidos, podemos extraer distintas conclusiones con respecto a los parámetros empleados por el SBRDE propuesto:

- En primer lugar, es posible observar que, en la mayoría de los casos, los mejores resultados se obtienen cuando no se considera el vector de ángulos de rotación en la $EE - (\mu, \lambda)$ que implementa el método de aprendizaje de consecuentes TSK en el proceso evolutivo de generación, es decir, cuando la combinación empleada es $(n_\sigma, n_\alpha) = (n, 0)$.
- Por otro lado, no es sencillo decidir cuál es el mejor valor para el parámetro α en la segunda etapa del aprendizaje, debido a que el comportamiento del proceso global depende de la aplicación concreta cuando cambia el valor del mismo. A continuación analizamos la influencia de dicho parámetro en los cuatro problemas:

Tabla 6.3. Resultados obtenidos por el SBRDE propuesto en el modelado difuso de la función F_3

| PARAMETROS | | | | GENERACION | | REFINAMIENTO | | MEDIA | |
|------------|------------|----------|-----|---------------|---------------|---------------|---------------|-------------------|-------------------|
| n_σ | n_α | α | Ej. | EC_{entr}^g | EC_{prue}^g | EC_{entr}^r | EC_{prue}^r | EC_{entr}^{med} | EC_{prue}^{med} |
| 3 | 0 | 0 | 1 | | | 27875.61 | 28289.87 | | |
| 3 | 0 | 0 | 2 | 46104.04 | 34909.11 | 28239.66 | 24423.53 | 28666.98 | 26569.31 |
| 3 | 0 | 0 | 3 | | | 29885.68 | 26994.53 | | |
| 3 | 0 | 0.1 | 1 | | | 29403.92 | 30013.45 | | |
| 3 | 0 | 0.1 | 2 | 46104.04 | 34909.11 | 30676.89 | 25773.23 | 29590.67 | 25.208.49 |
| 3 | 0 | 0.1 | 3 | | | 28691.20 | 19838.80 | | |
| 3 | 0 | 0.2 | 1 | | | 27798.84 | 22102.10 | | |
| 3 | 0 | 0.2 | 2 | 46104.04 | 34909.11 | 30608.80 | 23747.40 | 28246.46 | 23299.51 |
| 3 | 0 | 0.2 | 3 | | | 26331.75 | 24049.04 | | |
| 3 | 3 | 0 | 1 | | | 42419.59 | 35670.11 | | |
| 3 | 3 | 0 | 2 | 82619.75 | 56026.21 | 46582.66 | 24168.58 | 41613.67 | 27155.02 |
| 3 | 3 | 0 | 3 | | | 35838.77 | 21626.37 | | |
| 3 | 3 | 0.1 | 1 | | | 43105.20 | 44799.02 | | |
| 3 | 3 | 0.1 | 2 | 82619.75 | 56026.21 | 43585.87 | 32632.88 | 40557.10 | 35086.33 |
| 3 | 3 | 0.1 | 3 | | | 34980.24 | 27827.10 | | |
| 3 | 3 | 0.2 | 1 | | | 35922.06 | 28787.99 | | |
| 3 | 3 | 0.2 | 2 | 82619.75 | 56026.21 | 39166.98 | 21030.00 | 37788.36 | 24416.09 |
| 3 | 3 | 0.2 | 3 | | | 38276.04 | 23430.27 | | |

- No existen diferencias significativas entre los resultados obtenidos en el modelado de F_1 empleando los tres valores considerados, $\alpha = \{0, 0.1, 0.2\}$. El mejor resultado individual se ha obtenido con $\alpha = 0$ mientras que, en media, presenta mejor rendimiento el valor $\alpha = 0.1$, pero de nuevo tenemos que destacar que las diferencias no son demasiado significativas.
- Las conclusiones son diferentes al analizar los resultados obtenidos en la segunda de las aplicaciones, el modelado difuso de F_2 . En este caso, cuanto mayor es el valor de α , es decir, cuantas más veces se aplica la $EE - (1 + 1)$ para optimizar los mejores individuos de la población genética, mejor aproximan el conjunto de entrenamiento los SBRDs TSK generados, pero peor generalizan el de prueba. Con ello, la aplicación de la $EE - (1 + 1)$ produce un sobreaprendizaje no deseado en el proceso evolutivo. Los mejores comportamientos desde un punto de vista individual y en media los presentan los SBRDs diseñados con $\alpha = 0$.
- En cambio, analizando los resultados obtenidos en la tercera aplicación encontramos exactamente el efecto contrario. En esta ocasión, el aumento progresivo

Tabla 6.4. Resultados obtenidos por el SBRDE propuesto en el modelado difuso de la función F_4

| PARAMETROS | | | | GENERACION | | REFINAMIENTO | | MEDIA | |
|------------|------------|----------|-----|---------------|---------------|---------------|---------------|-------------------|-------------------|
| n_σ | n_α | α | Ej. | EC_{entr}^g | EC_{prue}^g | EC_{entr}^r | EC_{prue}^r | EC_{entr}^{med} | EC_{prue}^{med} |
| 3 | 0 | 0 | 1 | | | 0.409240 | 1.033788 | | |
| 3 | 0 | 0 | 2 | 0.501415 | 1.139916 | 0.383350 | 0.958478 | 0.397303 | 1.016707 |
| 3 | 0 | 0 | 3 | | | 0.399321 | 1.057857 | | |
| 3 | 0 | 0.1 | 1 | | | 0.326539 | 1.143749 | | |
| 3 | 0 | 0.1 | 2 | 0.501415 | 1.139916 | 0.373687 | 0.960348 | 0.349551 | 1.090523 |
| 3 | 0 | 0.1 | 3 | | | 0.348427 | 1.167474 | | |
| 3 | 0 | 0.2 | 1 | | | 0.370438 | 1.218274 | | |
| 3 | 0 | 0.2 | 2 | 0.501415 | 1.139916 | 0.329548 | 0.842741 | 0.356530 | 1.047296 |
| 3 | 0 | 0.2 | 3 | | | 0.369606 | 1.080873 | | |
| 3 | 3 | 0 | 1 | | | 0.400272 | 0.973358 | | |
| 3 | 3 | 0 | 2 | 0.501422 | 1.140024 | 0.382709 | 0.976513 | 0.403053 | 0.972606 |
| 3 | 3 | 0 | 3 | | | 0.426180 | 0.967949 | | |
| 3 | 3 | 0.1 | 1 | | | 0.379530 | 0.949584 | | |
| 3 | 3 | 0.1 | 2 | 0.501422 | 1.140024 | 0.364021 | 1.051470 | 0.365522 | 1.006885 |
| 3 | 3 | 0.1 | 3 | | | 0.353017 | 1.019602 | | |
| 3 | 3 | 0.2 | 1 | | | 0.374604 | 1.079302 | | |
| 3 | 3 | 0.2 | 2 | 0.501422 | 1.140024 | 0.354056 | 1.127499 | 0.369014 | 1.094364 |
| 3 | 3 | 0.2 | 3 | | | 0.378383 | 1.076293 | | |

del valor de α da lugar a que el proceso evolutivo de aprendizaje mejore el nivel de generalización de los SBRDs generados. El mejor resultado medio se obtiene con $\alpha = 0.2$ y el mejor individual con $\alpha = 0.1$.

- Finalmente, en lo que respecta al modelado de la cuarta función, cabe destacar, en primer lugar, que es el único caso en el que se consiguen mejores resultados en media al considerar el vector de ángulos de rotación en la $EE - (\mu, \lambda)$ empleada en la primera etapa del aprendizaje. De este modo, el mejor valor medio se obtiene, en este caso, cuando el parámetro α es igual a 0. En cambio, el mejor resultado individual se obtiene cuando no se considera la existencia del vector de ángulos en la primera fase y se emplea el valor $\alpha = 0.2$. El comportamiento general es similar al obtenido en el modelado de la función F_2 , es decir, el aumento del valor de α provoca un sobreaprendizaje que da lugar a que los modelos obtenidos no generalicen correctamente. Este hecho ocurre en todos los casos, excepto cuando se considera la combinación $n_\alpha = 0$ y $\alpha = 0.2$, con la que se obtiene un mejor nivel de generalización en media que con $n_\alpha = 0$ y $\alpha = 0.1$.

6.4.2 Estudio Comparativo y Análisis de los Resultados Obtenidos

En esta sección vamos a comparar el rendimiento del SBRDE presentado con uno de los procesos evolutivos de diseño de SBRDs de tipo TSK más conocidos, el propuesto por Lee y Takagi en [LT93b, LT93a], el cual es capaz de generar una BC TSK al completo, aprendiendo el número y la estructura de los antecedentes y consecuentes de las reglas difusas que la componen. Los aspectos básicos de dicho SBRDE fueron introducidos en la sección 2.4.3.1 del capítulo 2.

Hemos realizado una serie de cambios sobre el proceso comentado para poder efectuar la comparación de un modo adecuado. Con el propósito de mejorar el comportamiento del SBRDE de Lee y Takagi, al mismo tiempo que se mantiene su filosofía original de trabajo, hemos modificado el esquema de codificación, pasando del esquema binario con ocho bits para cada parámetro, empleado por los autores, al esquema real, el cual parece más adecuado para trabajar con un problema en el que las variables están definidas sobre un universo de discurso continuo. Puesto que es clara la necesidad de usar operadores genéticos que exploten adecuadamente la potencialidad de la codificación real, hemos hecho uso de los mismos operadores considerados para el proceso evolutivo de refinamiento que constituye la segunda etapa del SBRDE introducido en este capítulo, el cruce max-min-aritmético y la mutación no uniforme.

Por otro lado, la composición de la función de adaptación ha sido modificada de acuerdo al tipo de problema que se pretende resolver y se ha empleado también la considerada en el proceso comentado. Esto se debe a que la propuesta original de Lee y Takagi empleaba una función de comportamiento, puesto que el SBRDE se aplicaba al control del péndulo invertido y trabajaba directamente sobre un modelo del sistema. Dejando aparte estos cambios, hemos hecho uso de las demás componentes propuestos originalmente, manteniendo el mismo esquema de representación, el mismo proceso de generación de la población inicial basado en el conocimiento disponible y los mismos parámetros asociados al AG.

Notaremos por **T1** al SBRDE de Lee y Takagi, modificado en el modo comentado en el párrafo anterior, y por **T2** al SBRDE en dos etapas propuesto en este capítulo. Los valores de los parámetros considerados para la ejecución del primer proceso son los siguientes: Tamaño de la población: $N = 10$, valores de las probabilidades de cruce y mutación: $P_c = 0.6$ y $P_m = 0.033$ (por gen), número de generaciones: 5000, valores asociados al operador de cruce max-min-aritmético y a la mutación no uniforme: $a = 0.35$ y $b = 5$, número máximo de etiquetas que puede emplear la partición difusa asociada a cada variable lingüística de entrada: 10.

Los resultados obtenidos están recogidos en las tablas 6.5 y 6.6, en las que $\#R$ representa el número de reglas que componen la BC TSK en cuestión.

Los resultados mostrados en dichas tablas demuestran el mejor comportamiento de nuestro SBRDE en todos los casos, excepto en el modelado difuso de la función F_4 , la

Tabla 6.5. Resultados obtenidos en el modelado difuso de las funciones F_1 y F_2

| SBRDE | F_1 | | | F_2 | | |
|-----------|-------------|-------------------|-------------------|-------------|-------------------|-------------------|
| | $\#R^{F_1}$ | $EC_{entr}^{F_1}$ | $EC_{prue}^{F_1}$ | $\#R^{F_2}$ | $EC_{entr}^{F_2}$ | $EC_{prue}^{F_2}$ |
| T1 | 56 | 0.579574 | 0.233781 | 56 | 0.067474 | 0.025405 |
| T2 | 49 | 0.006592 | 0.006568 | 49 | 0.015290 | 0.001763 |

Tabla 6.6. Resultados obtenidos en el modelado difuso de las funciones F_3 y F_4

| SBRDE | F_3 | | | F_4 | | |
|-----------|-------------|-------------------|-------------------|-------------|-------------------|-------------------|
| | $\#R^{F_3}$ | $EC_{entr}^{F_3}$ | $EC_{prue}^{F_3}$ | $\#R^{F_4}$ | $EC_{entr}^{F_4}$ | $EC_{prue}^{F_4}$ |
| T1 | 56 | 68971.09 | 56671.50 | 6 | 0.424333 | 0.494060 |
| T2 | 49 | 28691.20 | 19838.80 | 49 | 0.329548 | 0.842741 |

más compleja de todas las consideradas, en el que el modelo obtenido a partir de nuestro proceso aproxima mejor el conjunto de entrenamiento, pero generaliza peor el de prueba. En los tres modelados restantes, el SBRDE presentado en este capítulo consigue mejores resultados que el SBRDE de Lee y Takagi, tanto en la aproximación del conjunto de entrenamiento como en el nivel de generalización del conjunto de prueba. Además, los modelos difusos generados a partir de nuestro proceso evolutivo de aprendizaje presentan una BC compuesta por un número menor de reglas en los tres problemas comentados. Finalmente, cabe destacar el hecho de que el SBRDE TSK propuesto obtiene los mejores modelados para las dos primeras funciones, F_1 y F_2 , de entre todos los procesos evolutivos de aprendizaje considerados en esta memoria.

Las figuras 6.1 a 6.4 representan gráficamente el comportamiento de los distintos modelos difusos obtenidos a partir de ambos SBRDEs.

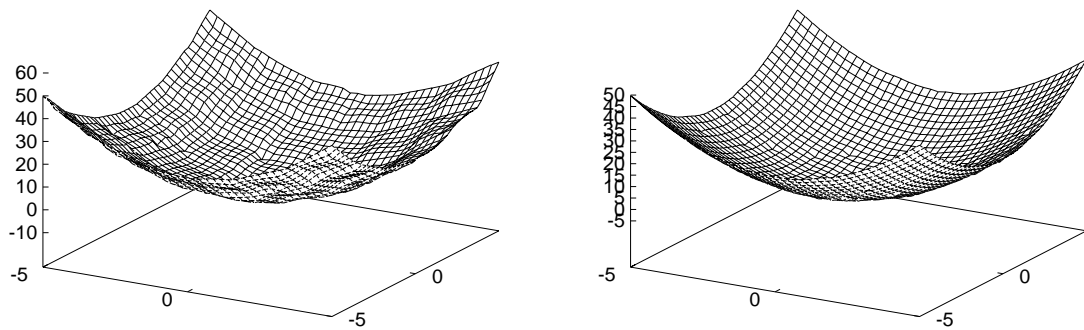


Figura 6.1. Representaciones gráficas de los dos modelados obtenidos para la función F_1 empleando los procesos **T1** (derecha) y **T2** (izquierda)

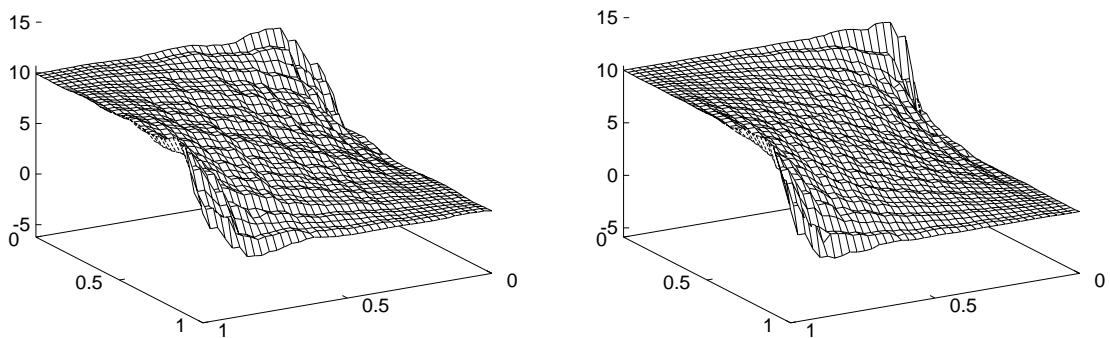


Figura 6.2. Representaciones gráficas de los dos modelados obtenidos para la función F_2 empleando los procesos **T1** (derecha) y **T2** (izquierda)

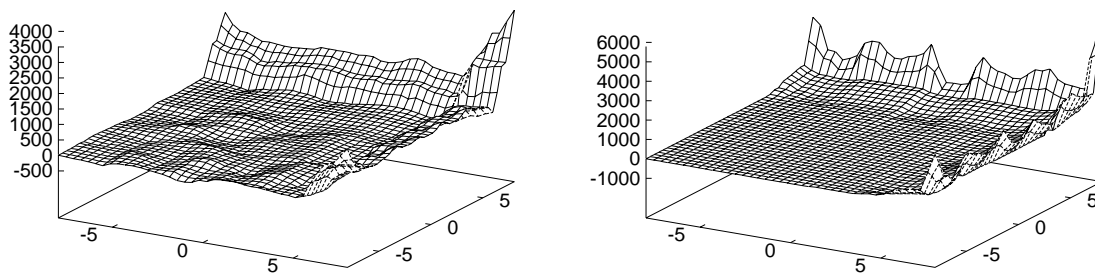


Figura 6.3. Representaciones gráficas de los dos modelados obtenidos para la función F_3 empleando los procesos **T1** (derecha) y **T2** (izquierda)

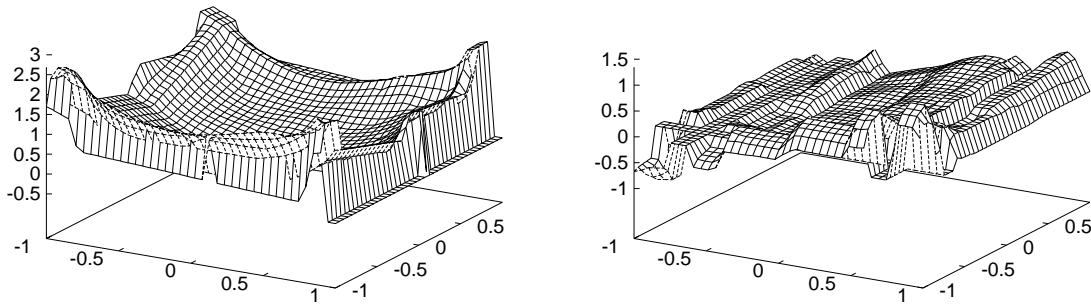


Figura 6.4. Representaciones gráficas de los dos modelados obtenidos para la función F_4 empleando los procesos **T1** (derecha) y **T2** (izquierda)

Capítulo 7

Aplicaciones

Este capítulo está dedicado a estudiar el comportamiento de los distintos SBRDEs contruidos a partir de la metodología evolutiva presentada en la resolución de una serie de problemas reales de modelado.

Hasta el momento, las aplicaciones que se han considerado para evaluar el rendimiento de estos procesos evolutivos de aprendizaje a lo largo de los capítulos 4, 5 y 6 han sido problemas de laboratorio seleccionados para poder demostrar el funcionamiento de cada tipo de SBRD diseñado en un tipo de problema con un nivel de complejidad concreto. Puesto que el fin del paradigma evolutivo de aprendizaje propuesto es el de poder ser aplicado en problemas reales de cualquier tipo en función de las necesidades de modelado que se presenten, las aplicaciones afrontadas en este último capítulo nos van a permitir aportar una visión más clara de las posibilidades de la metodología evolutiva de aprendizaje presentada en esta memoria.

Hemos escogido tres problemas con requerimientos de modelado bien distintos. En primer lugar, trabajaremos con dos problemas de predicción en los que el objetivo será estimar el costo de mantenimiento del cable de baja tensión instalado en los distintos pueblos de una provincia [San97a, San97b] y el costo de instalación de un cableado de media tensión óptimo en las ciudades de la misma [San97c], respectivamente. Estos costos serán función de distintas variables dependiendo del problema concreto. Por ejemplo, en el primero de ellos, el costo de mantenimiento de la línea de baja tensión vendrá determinado directamente por la longitud de cable de este tipo instalado en los núcleos rurales por la compañía. La problemática estriba, en este caso, como veremos a lo largo de este capítulo, en que la medición directa de la cantidad de cable empleado es cara y difícil de realizar, debido a factores tales como la gran cantidad de núcleos rurales a los que sirve la compañía o la orografía del terreno.

Para resolver estos dos problemas, dispondremos de dos conjuntos de datos, el primero de ellos proporcionado por la *Hidroeléctrica del Cantábrico* y el segundo obtenido a partir de un simulador del problema, que contienen información sobre distintas características de 495 núcleos rurales situados en la provincia de Asturias (en este primer caso, los datos

serán reales) y de 1059 núcleos urbanos simulados. Los vectores de datos contenidos en los ficheros tienen asociadas, respectivamente, dos y cuatro variables de entrada, además de la variable de salida. En este caso, el requerimiento principal del modelado será la precisión del modelo obtenido.

La tercera de las aplicaciones consideradas es un problema de clasificación con salidas reales, en el que se trata de valorar la calidad de un tipo de arroz en función de una serie de características del grano [INT⁺94, NIT97]. Se dispone de un conjunto formado por 105 vectores de datos, asociados a otros tantos tipos de arroz distintos. Estos datos están compuestos por los valores de cinco variables de entrada y una de salida, que indica la calidad del tipo de arroz. En este segundo caso, el objetivo principal no es sólo la obtención de una buena aproximación en la clasificación, sino también la interpretabilidad del modelo obtenido. De este modo, aplicaremos únicamente el SBRDE para la generación de BCs descriptivas presentado en el capítulo 4 y compararemos su funcionamiento con el obtenido por alguno de los procesos evolutivos de diseño de SBRDs descriptivos de tipo Mamdani considerados en esta memoria, así como con el método propuesto en [NIT97].

La estructura de este capítulo incluye dos secciones dedicadas, respectivamente, a las dos primeras y a la última de las aplicaciones comentadas. Cada sección estará dividida en tres partes principales, una primera en la que se introduce el problema en cuestión, otra en la que se describe su resolución por medio de los SBRDEs propuestos y se comparan los resultados obtenidos con los asociados a los restantes procesos evolutivos de aprendizaje considerados, y una última en la que se muestran algunas conclusiones.

7.1 Aplicación a la Estimación de Costos de Cable de Baja y Media Tensión en Zonas Rurales y Urbanas

7.1.1 Planteamiento del Problema

La explotación del sistema eléctrico nacional es un problema complejo. El sistema español está compuesto por varios cientos de generadores (centrales térmicas, hidroeléctricas y nucleares, principalmente) y por una gran cantidad de consumidores con requisitos de potencia muy distintos, que abarcan desde las viviendas familiares hasta grandes demandantes de energía como las empresas productoras de aluminio o las siderúrgicas.

La energía se transporta desde los generadores hasta los puntos de consumo mediante líneas eléctricas de distinta capacidad. La tensión a la que opera una línea es, por motivos de eficiencia, mayor cuanto más elevada es la potencia que conduce. Se distingue así entre líneas de alta tensión (100 – 400KV), líneas de media tensión (10KV – 100KV) y líneas

de baja tensión (menor de 10KV). Todos los generadores están interconectados mediante líneas de alta tensión, de modo que no es posible conocer el origen de la energía de la que hace uso un consumidor final.

Por otra parte, las líneas de alta tensión son propiedad de R.E.E. (Red Eléctrica Española), empresa participada por todas las compañías eléctricas españolas, tanto las de titularidad pública como las privadas. Sin embargo, las líneas de baja y media tensión pertenecen a compañías determinadas, que se hacen cargo de su instalación y corren con los gastos de su mantenimiento.

Tanto la imposibilidad de conocer el origen de la energía, como la titularidad compartida de las líneas complicarían enormemente el cálculo de la tarifa que un usuario final debería abonar. Por ello, las leyes españolas obligan a que la factura eléctrica sea cobrada por R.E.E. y establecen unos criterios de reparto del total de los ingresos entre los distintos generadores. Estos criterios tienen en cuenta los costes de funcionamiento de los generadores y de operación de la red, en función de indicadores tales como el número de usuarios, la potencia generada o la longitud de línea mantenida.

Este sistema de reparto está siendo revisado actualmente, ya que algunas compañías no comparten su formulación. Entre otras, se alegan razones tales como que, por motivos históricos y políticos, los criterios de división de los costes de instalación y mantenimiento de las redes de media y baja tensión no se basan en el tendido idóneo del cable, sino en estimaciones de la longitud actual que benefician a las redes más antiguas e ineficientes (con la consiguiente falta de incentivos para su modernización). En otros casos, simplemente se cuestiona la veracidad de la fórmula legal que estima los costes a partir de los descriptores del núcleo de población.

Para estudiar el impacto de esta redistribución es preciso conocer tanto la longitud real de línea tendida en una zona determinada, como la longitud que sería preciso emplear si el trazado se ejecutase desde cero, con la cantidad de línea mínima necesaria para atender a todos los consumidores. Como puede observarse, este hecho plantea la necesidad de elaborar modelos que permitan resolver ambos problemas, debido a la imposibilidad de obtener una solución procediendo de un modo más directo. Por ejemplo, en lo que respecta al primero de ellos, es muy difícil y caro realizar una medición correcta de la longitud de línea eléctrica de baja tensión de la que dispone una compañía, debido a que este tipo de línea es la de mayor longitud y la más frecuentemente reparada; además de otras razones como la orografía del terreno o la gran cantidad de núcleos rurales servidos (en algunas ocasiones, más de 10000).

A modo de ejemplo, se estudiarán dos de estos modelos: el de la longitud real de la línea de baja tensión en núcleos rurales (proporcional al coste total) y el de los costes óptimos –teóricos– de la de media tensión en núcleos urbanos.

Una posible solución al primero de los problemas consiste en emplear un sistema de predicción que sea capaz de determinar la relación existente entre la longitud de la línea de baja tensión tendida en una población y otras variables características de ésta, tales

como su radio o el número de usuarios que viven en ella; y de hacer uso de la relación obtenida para estimar la longitud de cable tendido en otros pueblos a partir de los valores de las variables características, sin que sea necesario efectuar una medición directa de la misma.

En el segundo de los modelos considerados, la problemática es otra. El crecimiento de la línea eléctrica dentro de un núcleo urbano es desordenado. Frecuentemente han de redimensionarse centros de transformación o líneas de transmisión. En estos casos, es interesante conocer cuál sería la extensión óptima de la red, si se pudiese tender toda la línea de nuevo. Para calcular esta medida, se ha desarrollado un simulador que realiza este tendido teórico en varios miles de ciudades tipo, con características definidas aleatoriamente, aunque razonables en las circunstancias españolas. Una vez realizadas estas simulaciones, se desea encontrar la correlación entre los indicadores de las características de la ciudad (como las dimensiones medias de sus manzanas, o su extensión) y la longitud más favorable de la línea.

7.1.2 Obtención de un Modelo Difuso de Zona Rural a partir de los Sistemas Basados en Reglas Difusas Evolutivos Propuestos

En esta sección nos plantearemos la obtención de un modelo difuso que nos permita llevar a cabo la primera de las tareas indicadas en la sección anterior. Para ello, disponemos de un conjunto de datos que recoge los valores de distintas características de 495 pueblos de la provincia de Asturias [San97a, San97b].

Cada vector de datos contenido en el fichero en cuestión está formado por tres componentes, dos asociadas a las variables de entrada y una a la de salida. Las dos primeras son, respectivamente, el *número de abonados*, $E1$, que toma valores en el intervalo $[1, 320]$, y el *radio de la población*, $E2$, definido a su vez en $[60, 1673]$. La variable de salida S es la *longitud de cable de baja tensión* tendido en dicha población que toma valores en $[80, 7675]$. Como ya hemos indicado, el coste de mantenimiento de la línea de baja tensión en zonas rurales depende directamente de la longitud de ésta.

Para obtener el modelo difuso que nos permita predecir la longitud del cable de baja tensión instalado vamos a proceder de la siguiente forma: dividiremos el conjunto de datos del que se dispone en dos subconjuntos distintos que serán empleados, respectivamente, como conjunto de entrenamiento para la generación del modelo difuso y como conjunto de prueba para el análisis de la capacidad de generalización del modelo obtenido. La división se llevará a cabo de forma aleatoria teniendo en cuenta que emplearemos un 80 por ciento de los datos originales para formar el conjunto de entrenamiento y que el 20 por ciento restante pasará a formar parte del conjunto de prueba.

Puesto que el objetivo que se persigue es la obtención de la mayor capacidad de

predicción posible, haremos uso de todos los SBRDEs presentados en esta memoria para tratar de resolver el problema. De este modo, los procesos evolutivos de aprendizaje considerados serán los siguientes:

- D1.** Un SBRDE en dos etapas compuesto por el algoritmo de WM presentado en el Apéndice B, como primera fase, y el proceso genético de ajuste de BCs descriptivas de tipo Mamdani introducido en la sección 4.4 del capítulo 4, como segunda.
- D2.** Un SBRDE en dos etapas compuesto por el proceso evolutivo de aprendizaje de BRLs de Thrift presentado en la sección 2.4.2.1 del capítulo 2, como primera fase, y el proceso genético de ajuste de BCs descriptivas comentado, como segunda.
- D3.** El SBRDE en tres etapas para el aprendizaje de BCs descriptivas de tipo Mamdani presentado en el capítulo 4.
- A1.** Un SBRDE en dos etapas compuesto por el algoritmo de WM como primera fase y el proceso genético de ajuste de BRDs aproximativas de tipo Mamdani introducido en la sección 5.1.3, como segunda.
- A2.** Un SBRDE en dos etapas compuesto por el ACP presentado en el Apéndice C y el proceso genético de ajuste de BRDs aproximativas comentado.
- A3.** El SBRDE en tres etapas basado en semántica libre no restringida propuesto en la sección 5.1 del capítulo 5.
- A4.** El SBRDE en tres etapas basado en semántica libre restringida propuesto en la sección 5.2 del capítulo 5.
- T1.** El SBRDE de Lee y Takagi introducido en la sección 2.4.3.1 del capítulo 2 y modificado en el modo comentado en la sección 6.4.2 del capítulo 6.
- T2.** El SBRDE en dos etapas para el diseño de SBRDs TSK propuesto en el capítulo 6.

Como puede observarse, la notación de cada uno de ellos representa el tipo de SBRD con el que se trabaja. La letra **D** especifica que el sistema en cuestión es descriptivo de tipo Mamdani, la letra **A** se refiere a que dicho sistema es aproximativo de tipo Mamdani y, finalmente, la **T** indica que estamos trabajando con un SBRD de tipo TSK.

En todos los experimentos realizados, la definición inicial de las funciones de pertenencia se ha efectuado llevando a cabo un proceso de normalización en el que se ha particionando cada espacio de entrada y salida (este último en el caso de los SBRDs de tipo Mamdani) en cinco partes iguales. En los SBRDs descriptivos de tipo Mamdani o TSK, se ha asociado a cada una de ellas una etiqueta lingüística del conjunto $\{MP, P, M, G, MG\}$ (con el siguiente significado: $MP = Muy Pequeño$, $P = Pequeño$, $M = Medio$, $G = Grande$ y $MG = Muy Grande$) y un conjunto difuso triangular que especifica la semántica de ésta. La partición difusa resultante se representa en la figura 7.1.

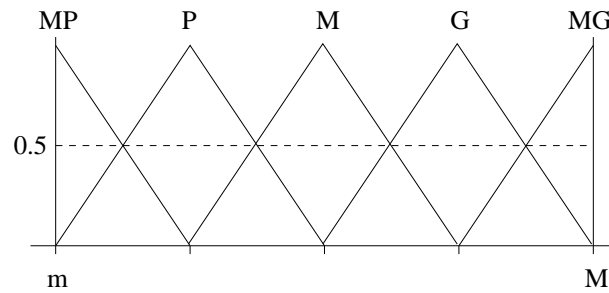


Figura 7.1. Partición difusa considerada en los experimentos realizados

Los parámetros considerados para la ejecución de los distintos SBRDEs son los que se muestran a continuación:

- **SBRDE D1:**

- *Proceso genético de ajuste de BCs descriptivas:* Tamaño de la población: $N = 61$, valores de las probabilidades de cruce y mutación: $P_c = 0.6$ y $P_m = 0.1$ (por individuo), número de generaciones: $Gen = 1000$, valores asociados al operador de cruce max-min-aritmético y a la mutación no uniforme: $a = 0.35$ y $b = 5$.

- **SBRDE D2:**

- *Proceso evolutivo de aprendizaje de BRLs de Thrift:* $N = 61$, $P_c = 0.6$, $P_m = 0.1$ (por individuo) y $Gen = 1000$.
- *Proceso genético de ajuste de BCs descriptivas:* los mismos que en la misma etapa del SBRDE **D1**.

- **SBRDE D3:**

- *Proceso iterativo de generación de reglas lingüísticas:* Valor de los parámetros considerados en la función de adaptación: $\epsilon = 1.5$, $\omega = 0.05$ y $k = 0.1$.
- *Proceso genético de multisimplificación:* $\tau = 0.1$, $N = 61$, $P_c = 0.6$, $P_m = 0.1$ (por individuo), $Gen = 500$; número de soluciones a generar: 3; valor del radio de nicho: $r = 10$ por ciento del número de reglas de la base generada en la primera fase; valor del factor de potencia $\beta = 0.5$.
- *Proceso genético de ajuste de BCs descriptivas:* los mismos que en la misma etapa del SBRDE **D1**.

- **SBRDE A1:**

- *Proceso genético de ajuste de BRDs:* los mismos que en el proceso genético de ajuste de BCs descriptivas considerado en los SBRDEs **D1**, **D2** y **D3**.

● **SBRDE A2:**

- *Algoritmo de Conteo Ponderado:* Para los intervalos del antecedente, consideramos el soporte de los cinco conjuntos difusos existentes en las particiones de entrada. Valor del parámetro que determina el grado de cubrimiento asociado a los ejemplos considerados para determinar la forma del consecuente: $\phi = 0.5$.
- *Proceso genético de ajuste de BRDs:* los mismos que en la misma etapa del SBRDE **A1**.

● **SBRDE A3:**

- *Proceso evolutivo de generación de reglas difusas aproximativas de tipo Mamdani con semántica libre no restringida:* $\epsilon = 1.5$, $\omega = 0.05$, $k = 0.1$; número de iteraciones sin obtener mejora antes de finalizar la ejecución de la EE-(1 + 1): 100; valor del parámetro de la regla de éxito $\frac{1}{5}$ de Rechenberg: $c = 0.9$.
- *Proceso genético de multisimplificación:* los mismos que en la misma etapa del SBRDE **D3**.
- *Proceso genético de ajuste de BCs descriptivas:* los mismos que en la misma etapa del SBRDE **A1**.

● **SBRDE A4:**

- *Proceso evolutivo de generación de reglas difusas aproximativas de tipo Mamdani con semántica libre restringida:* $\epsilon = 1.5$, $\omega = 0.05$ y $k = 0.1$, $N = 61$, $P_c = 0.6$, $P_m = 0.1$ (por individuo), $Gen = 100$, $a = 0.35$, $b = 5$; porcentaje de individuos sobre los que se aplica la EE-(1 + 1) en cada generación: $\alpha = 0.2$; número de iteraciones sin obtener mejora antes de finalizar la ejecución de la EE-(1 + 1): 25; valor del parámetro de la regla de éxito $\frac{1}{5}$ de Rechenberg: $c = 0.9$.
- *Proceso genético de multisimplificación:* los mismos que en la misma etapa del SBRDE **D3**.
- *Proceso genético de ajuste de BCs descriptivas:* los mismos que en la misma etapa del SBRDE **A1**.

- **SBRDE T1:** $N = 10$, $P_c = 0.6$, $P_m = 0.033$ (por gen), $Gen = 5000$, $a = 0.35$, $b = 5$; número máximo de etiquetas que puede emplear la partición difusa asociada a cada variable lingüística de entrada: 5.

● **SBRDE T2:**

- *Proceso Evolutivo de Generación de Reglas de Tipo TSK:* Número de iteraciones de la EE-(μ, λ): 500; número de padres e hijos a generar en cada iteración: $\mu = 15$ y $\lambda = 100$; número de individuos de la población inicial que son generados en el segundo paso del procedimiento de inicialización: $\gamma = 0.2 \cdot \mu = 3$;

Tabla 7.1. Resultados obtenidos en el modelado de zonas rurales

| SBRDE | #R | Generación | | Multisimplificación | | | Ajuste | |
|-----------|----|-------------|-------------|---------------------|-------------|--------------------|-------------|-------------|
| | | EC_{entr} | EC_{prue} | #R | EC_{entr} | EC_{prue} | EC_{entr} | EC_{prue} |
| D1 | 13 | 298446.0 | 282058.1 | | | | 175337.9 | 180102.7 |
| D2 | 25 | 218591.9 | 204426.8 | | | | 154314.0 | 199551.3 |
| D3 | 47 | 286794.8 | 285869.0 | 22 | 168603.2 | 174246.6 | 143921.9 | 177790.0 |
| | | | | 19 | 166484.8 | 170550.1 | 150559.4 | 166669.7 |
| | | | | 21 | 166484.8 | 170550.1 | 150507.5 | 166877.6 |
| A1 | 13 | 298446.0 | 282058.1 | | | | 149815.5 | 190530.7 |
| A2 | 20 | 356434.3 | 311195.0 | | | | 175887.2 | 180211.4 |
| A3 | 31 | 431904.0 | 435649.5 | 19 | 226403.9 | 222550.9 | 148036.9 | 191339.5 |
| | | | | 20 | 227261.6 | 227105.6 | 142108.6 | 166578.7 |
| | | | | 16 | 227232.9 | 225789.7 | 136826.4 | 177612.3 |
| A4 | 29 | 265612.5 | 254904.8 | 16 | 177105.7 | 170839.0 | 149128.5 | 168209.9 |
| | | | | 18 | 177190.7 | 187461.0 | 149672.7 | 176384.0 |
| | | | | 16 | 177248.2 | 170943.6 | 145461.2 | 171879.1 |
| T1 | 4 | 169761.7 | 160110.0 | | | | | |
| T2 | 20 | 162609.9 | 148514.7 | | | | | |
| | | | | | | ($\alpha = 0$) | 147192.2 | 149689.0 |
| | | | | | | ($\alpha = 0.2$) | 133958.6 | 190293.2 |

valor del parámetro que determina el umbral de pertenencia a partir del cual un ejemplo se considera significativo en el procedimiento de inicialización: $\theta = 0.7$; valor del parámetro empleado en la función modificadora encargada de la generación aleatoria de valores angulares pequeños en el procedimiento de inicialización: $q = 5$; mecanismo de combinación de genes considerado en cada componente del individuo $\vec{r} = (r_{\vec{x}}, r_{\vec{\sigma}}, r_{\vec{\alpha}}) = (3, 2, 0)$; número de padres considerados en el proceso de recombinación en cada componente: $\vec{\zeta} = (\zeta_{\vec{x}}, \zeta_{\vec{\sigma}}, \zeta_{\vec{\alpha}}) = (\mu, \mu, 1)$.

- *Proceso Evolutivo de Refinamiento de BCs TSK*: $N = 61$, $P_c = 0.6$, $P_m = 0.1$ (por individuo), $Gen = 1000$, $a = 0.35$, $b = 5$; porcentaje de individuos sobre los que se aplica la EE-(1 + 1) en cada generación: $\alpha = \{0, 0.2\}$; número de iteraciones sin obtener mejora antes de finalizar la ejecución de la EE-(1 + 1): 25; valor del parámetro de la regla de éxito $\frac{1}{5}$ de Rechenberg empleada en la EE: $c = 0.9$; valor de la desviación típica empleada para generar los componentes de C_2 en el último paso del proceso de inicialización: $d = 0.001$.

Los resultados obtenidos en los distintos experimentos realizados se recogen en la tabla 7.1 en la que, al igual que en el resto de los experimentos realizados en los distintos capítulos de esta memoria, $\#R$ representa el número de reglas existente en la base de reglas difusas correspondiente, y EC_{entr} y EC_{prue} los valores obtenidos en la medida de ECM sobre los conjuntos de entrenamiento y prueba, respectivamente.

7.1.3 Obtención de un Modelo Difuso de Zona Urbana a partir de los Sistemas Basados en Reglas Difusas Evolutivos Propuestos

Para resolver este segundo problema, haremos uso de un conjunto formado por 1059 vectores de datos simulados que recogen las características de otras tantas ciudades [San97c]. En este caso, se consideran cuatro variables de entrada y una de salida, las cuales se muestran a continuación:

- E1*. Un ratio resultante de efectuar el producto del número de manzanas existentes en la población y de la longitud media de una manzana. $E1 \in [0.5, 11]$.
- E2*. Área total del casco urbano de la población. $E2 \in [0.15, 8.55]$.
- E3*. Área construida en dicho casco urbano. $E3 \in [1.64, 142.5]$.
- E4*. Potencia consumida por la población. $E4 \in [1, 165]$.
- S*. Coste de la línea eléctrica de media tensión instalada en la población.
 $S \in [0, 8546.030273]$.

A la vista de estos datos, se observa que el costo de la línea urbana de media tensión depende de factores como la potencia consumida, el número de manzanas existente en la ciudad y la longitud media de éstas, etc. Este hecho se debe, entre otras razones, a que en las ciudades es corriente que el cable se entierre en zanjas que suelen contener más de un tipo de instalación, lo que provoca que el coste tenga dependencias más complejas. Además, en este caso, se han tenido en cuenta otros tipos de costes incluidos en el total, tales como el coste de pérdidas en la línea, el coste de los centros de transformación (mientras que en los pequeños núcleos rurales únicamente existe uno, en las ciudades se suele disponer de varios) y el coste de las zanjas.

Trabajaremos del mismo modo en el que lo hicimos en la sección anterior, es decir, dividiremos el conjunto de datos en dos subconjuntos de entrenamiento y prueba formados respectivamente por el 80 y el 20 por ciento de los datos del conjunto original. Hemos aplicado los mismos procesos evolutivos de aprendizaje considerados en la sección anterior haciendo uso de los mismos parámetros excepto en el caso del valor del radio de nicho empleado en el proceso genético de multisimplificación, segunda etapa del aprendizaje en los SBRDEs **D3**, **A3** y **A4**, que en este caso es $r = 5$ por ciento del número de reglas de la base generada en la primera fase. Los resultados obtenidos se muestran en la tabla 7.2.

Analizando estos primeros resultados, podemos establecer una serie de conclusiones preliminares. En primer lugar, parece que la complejidad de este segundo sistema, el de zona urbana, es menor que la del considerado en la sección anterior, el de zona rural. Deducimos este hecho a la vista de que, dejando aparte el modelo difuso obtenido con

Tabla 7.2. Resultados obtenidos en el modelado de zonas urbanas

| SBRDE | Generación | | | Multisimplificación | | | Ajuste | | |
|-----------|------------|-------------|-------------|---------------------|-------------|-------------|--------------------|-------------|---------|
| | #R | EC_{entr} | EC_{prue} | #R | EC_{entr} | EC_{prue} | EC_{entr} | EC_{prue} | |
| D1 | 66 | 71294.4 | 80933.9 | | | | 20318.6 | 27615.8 | |
| D2 | 548 | 39763.8 | 43438.6 | | | | 21013.2 | 24636.3 | |
| D3 | 91 | 102001.1 | 106471.7 | 53 | 57071.8 | 59060.9 | 24480.6 | 28442.1 | |
| | | | | 57 | 58848.3 | 56786.8 | 21709.1 | 25290.3 | |
| | | | | 56 | 57121.8 | 55023.9 | 23636.9 | 26638.0 | |
| A1 | 66 | 71294.4 | 80933.9 | | | | 23396.7 | 27665.7 | |
| A2 | 273 | 159789.9 | 355979.8 | | | | 98111.9 | 96439.0 | |
| A3 | 42 | 125032.5 | 155758.1 | 23 | 88748.7 | 112373.5 | 46828.9 | 56531.8 | |
| | | | | 21 | 89313.9 | 120540.0 | 44233.6 | 57876.1 | |
| | | | | 23 | 89353.4 | 118055.7 | 48870.2 | 70952.2 | |
| A4 | 42 | 434993.2 | 430741.1 | 17 | 149709.5 | 137176.4 | 44105.0 | 51207.6 | |
| | | | | 17 | 149796.3 | 137456.2 | 48498.3 | 52877.0 | |
| | | | | 19 | 149903.0 | 137704.3 | 58295.3 | 63658.3 | |
| T1 | 6 | 149144.5 | 128942.2 | | | | | | |
| T2 | 268 | 45052.5 | 53300.9 | | | | ($\alpha = 0$) | 25579.9 | 26450.0 |
| | | | | | | | ($\alpha = 0.2$) | 11073.8 | 11836.2 |

el SBRDE **T2**, los distintos SBRDs descriptivos de tipo Mamdani diseñados son los que mejor comportamiento presentan en el modelado del mismo; además de que el mejor modelo difuso (no cualitativo) ha sido generado a partir del SBRDE **A1**, el cual presenta un proceso de generación de BCs descriptivas como primera fase.

Por otro lado, en lo que respecta a los resultados obtenidos por el SBRDE para el aprendizaje de BCs descriptivas de tipo Mamdani propuesto en esta memoria, **D3**, dicho proceso presenta un comportamiento correcto, ya que el mejor modelo obtenido a partir del mismo generaliza mejor (25290.3 contra 27615.8) que el generado mediante el proceso evolutivo de aprendizaje basado en WM, **D1**, y hay poca diferencia (25290.3 contra 24636.3) con el obtenido usando el SBRDE basado en el proceso de generación de BRLs de Thrift, **D2**.

En todos los casos, nuestro proceso genera menos reglas, 57, que los dos restantes, 66 con **D1** y 548 con **D2**. Por esta razón, nos planteamos realizar una nueva experimentación del mismo aumentando el valor del parámetro ϵ . Como se ha comentado a lo largo de esta memoria, este cambio debe conllevar la obtención de un modelo más complejo, es decir, con un número mayor de reglas en la base, y con mejor rendimiento.

Los resultados obtenidos, presentados en la tabla 7.3, corroboran este hecho. Como puede observarse en la misma, nuestro SBRDE permite generar tres modelos cualitativos que, con un número de reglas igual en un caso o menor en los dos restantes que el presentado por el modelo obtenido a partir del proceso **D1**, mejoran a los modelos generados

Tabla 7.3. Resultados obtenidos por el SBRDE **D3** con $\epsilon = 2$ en el modelado de zonas urbanas

| #R | Generación | | Multisimplificación | | | Ajuste | |
|-----|-------------|-------------|---------------------|-------------|-------------|-------------|-------------|
| | EC_{entr} | EC_{prue} | #R | EC_{entr} | EC_{prue} | EC_{entr} | EC_{prue} |
| 120 | 99299.3 | 107039.5 | 65 | 55642.8 | 56941.3 | 19030.7 | 24456.4 |
| | | | 66 | 53803.9 | 54126.6 | 20283.7 | 24270.5 |
| | | | 63 | 53751.1 | 54353.3 | 19679.0 | 22590.8 |

a partir de **D1** y **D2**, tanto en su nivel de aproximación del conjunto de entrenamiento, como en el de generalización del de prueba.

7.1.4 Conclusiones

7.1.4.1 Modelo Difuso de Zona Rural

Analizando los resultados obtenidos en los distintos experimentos desarrollados para la resolución del primer problema planteado, hemos llegado a las siguientes conclusiones:

- Los cuatro SBRDEs propuestos en esta memoria han presentado un buen comportamiento. En todos los casos, han obtenido el mejor modelado en comparación con los restantes procesos evolutivos de aprendizaje considerados para cada uno de los tipos de SBRDs existentes, tanto en la aproximación del conjunto de entrenamiento, como en el nivel de generalización presentado sobre el conjunto de prueba.
- El tipo de SBRD que ha permitido obtener mejores resultados ha sido el TSK. Los dos procesos evolutivos de diseño de SBRDs de este tipo considerados, **T1** y **T2**, han generado los dos mejores modelos difusos desde el punto de vista de la capacidad de generalización, presentando mejor comportamiento el segundo de ellos, presentado en el capítulo 6 de esta memoria, con respecto al primero, propuesto por Lee y Takagi en [LT93a, LT93b].

Aún así, cabe destacar que la mejor capacidad de predicción se obtiene con el modelo generado por nuestro proceso tras la primera etapa del aprendizaje, ya que la segunda, el proceso evolutivo de refinamiento, provoca un sobreaprendizaje en la aproximación del conjunto de entrenamiento que repercute en una menor capacidad de predicción en el modelo obtenido finalmente. Este hecho es muy acusado en la segunda ejecución realizada de dicho proceso, que consideraba el valor $\alpha = 0.2$. En la primera ($\alpha = 0$), sin embargo, este efecto es mucho menor y se obtiene un modelo que consigue la mejor aproximación del conjunto de entrenamiento de entre todos los generados, perdiendo muy poco en su capacidad de generalización, que sigue siendo

Tabla 7.4. Tablas de decisión de los modelos cualitativos obtenidos mediante el SBRDE **D1** para el problema del modelado de zonas rurales

| | | | | | | |
|-----------|-----------|-----------|----------|-----------|----------|-----------|
| | | <i>E2</i> | | | | |
| | | <i>MP</i> | <i>P</i> | <i>M</i> | <i>G</i> | <i>MG</i> |
| <i>E1</i> | <i>MP</i> | <i>MP</i> | <i>P</i> | <i>P</i> | <i>P</i> | <i>M</i> |
| | <i>P</i> | <i>MP</i> | <i>P</i> | <i>P</i> | <i>M</i> | |
| | <i>M</i> | | <i>P</i> | <i>MG</i> | <i>M</i> | |
| | <i>G</i> | | | | | |
| | <i>MG</i> | | | <i>M</i> | | |

Tabla 7.5. Tablas de decisión de los modelos cualitativos obtenidos mediante el SBRDE **D2** para el problema del modelado de zonas rurales

| | | | | | | |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | | <i>E2</i> | | | | |
| | | <i>MP</i> | <i>P</i> | <i>M</i> | <i>G</i> | <i>MG</i> |
| <i>E1</i> | <i>MP</i> | <i>MP</i> | <i>P</i> | <i>P</i> | <i>M</i> | <i>M</i> |
| | <i>P</i> | <i>MP</i> | <i>P</i> | <i>M</i> | <i>M</i> | <i>G</i> |
| | <i>M</i> | <i>P</i> | <i>P</i> | <i>MG</i> | <i>M</i> | <i>MG</i> |
| | <i>G</i> | <i>P</i> | <i>MP</i> | <i>G</i> | <i>G</i> | <i>M</i> |
| | <i>MG</i> | <i>G</i> | <i>P</i> | <i>M</i> | <i>MP</i> | <i>MG</i> |

mejor que la de los modelos difusos restantes. Este sobreaprendizaje se ha presentado también en la última ejecución del SBRDE **A4**, con un efecto muy similar al comentado.

- El SBRDE **D3**, propuesto en el capítulo 4 de esta memoria, ha permitido la generación de un modelo cualitativo con muy buen rendimiento, que presenta tan solo 19 reglas en su BC. Además, los tres modelos obtenidos mediante su ejecución consiguen mejores resultados sobre ambos conjuntos de datos que los generados a partir de los procesos **D1** y **D2**, estando formados, en todos los casos, por una BC con menor complejidad que la obtenida a partir de este último. Las tablas 7.4 a 7.6 muestran la composición de las BRLs de los tres modelos descriptivos comentados y permiten observar las diferencias existentes entre ellas.
- Del mismo modo, los dos SBRDEs para el aprendizaje de BRDs aproximativas de tipo Mamdani propuestos en esta memoria obtienen mejores resultados en la aproximación de ambos conjuntos que los otros dos procesos evolutivos de aprendizaje de BRDs de este tipo considerados. Ambos sistemas permiten generar modelos difusos con una capacidad de generalización del orden de la obtenida por el modelo

Tabla 7.6. Tablas de decisión de los modelos cualitativos obtenidos mediante el SBRDE **D3** para el problema del modelado de zonas rurales

| | | | | | | |
|-----------|-----------|--------------|---------------------------------------|---|----------|-------------|
| | | <i>E2</i> | | | | |
| | | <i>MP</i> | <i>P</i> | <i>M</i> | <i>G</i> | <i>MG</i> |
| <i>E1</i> | <i>MP</i> | $2 \cdot MP$ | $\begin{matrix} MP \\ P \end{matrix}$ | <i>P</i> | <i>P</i> | $2 \cdot M$ |
| | <i>P</i> | | $2 \cdot P$ | $\begin{matrix} P \\ M \\ G \end{matrix}$ | <i>M</i> | |
| | <i>M</i> | | $2 \cdot P$ | $2 \cdot MG$ | | |
| | <i>G</i> | | | | | |
| | <i>MG</i> | | | <i>M</i> | | |
| | | | | | | |

cualitativo comentado en el punto anterior (algo mejor en el caso del SBRDE **A3** y un poco peor en el caso de **A4**), pero con una mejor aproximación del conjunto de entrenamiento.

7.1.4.2 Modelo Difuso de Zona Urbana

Las conclusiones asociadas al segundo de los problemas analizados en esta sección son las que se muestran a continuación:

- El mejor modelado se ha obtenido haciendo uso del SBRDE para el aprendizaje de BCs de tipo TSK presentado en el capítulo 6 de esta memoria. El modelo generado ha conseguido el mejor resultado global en la aproximación de ambos conjuntos, con una diferencia bastante significativa con respecto a los restantes. Al contrario que en la aplicación estudiada en la sección anterior, en este caso el empleo de la $EE - (1 + 1)$ para optimizar el 20 por ciento ($\alpha = 0.2$) de los individuos de la población genética en el proceso de refinamiento ha dado lugar a la obtención del mejor comportamiento.
- Como ya hemos comentado anteriormente, el problema del modelado de zonas urbanas parece presentar menor complejidad que el de zonas rurales. Esta conclusión ha sido obtenida a la vista de que, sin tener en cuenta el resultado obtenido por el SBRDE para el aprendizaje de BCs TSK presentado en esta memoria, se han dado las dos circunstancias siguientes:
 - Todos los modelos cualitativos obtenidos a partir de los SBRDEs para el aprendizaje de BCs descriptivas de tipo Mamdani se comportan mucho mejor que los modelos difusos generados a partir de los distintos procesos evolutivos de aprendizaje de BRDs aproximativas.

- El SBRD aproximativo que mejor rendimiento presenta es el generado a partir del SBRDE que emplea el proceso de generación de reglas descriptivas de WM como primera fase del aprendizaje. Los demás han obtenido resultados bastante peores en la resolución del problema.

Este hecho se ratifica a la vista del comportamiento presentado por los dos SBRDEs introducidos en el capítulo 5, ya que el basado en semántica libre restringida, **A4**, permite obtener mejores resultados que el basado en la no restringida, **A3**. Recordemos que el primero de ellos era más adecuado para trabajar con problemas de complejidad intermedia, mientras que el segundo se comportaba mejor en problemas de alta complejidad. Por ello, el hecho de que el primero de ellos presente mejor rendimiento en la aplicación actual es un resultado coherente.

- El SBRDE para el aprendizaje de BCs descriptivas presentado en el capítulo 4 de esta memoria ha permitido obtener tres modelos cualitativos con mejor comportamiento y menor complejidad que los generados a partir de los otros dos SBRDEs de la misma familia considerados. Además, se ha demostrado de nuevo la coherencia de los parámetros considerados en la metodología presentada en el capítulo 3 con el hecho de que los SBRDs obtenidos empleando el valor $\epsilon = 2$ presentan mejor rendimiento y contienen más reglas en la BC que los generados tomando $\epsilon = 1.5$.

7.2 Valoración de la Calidad del Arroz

7.2.1 Planteamiento del Problema

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 denominado *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:

E1. Sabor.

E2. Aspecto.

E3. Gusto.

E4. Grado de pegajosidad.

E5. Dureza.

S. 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. Además, el objetivo no es únicamente el conseguir un modelado con buen rendimiento, sino también que el modelo obtenido pueda aportar alguna información sobre el proceso de razonamiento que lleva a cabo el experto humano para evaluar un tipo de arroz de una forma concreta. Por todas estas razones, en esta sección nos planteamos la obtención de un modelo cualitativo que nos permita resolver el problema planteado.

7.2.2 Obtención de un Modelo Cualitativo para la Valoración de la Calidad del Arroz a partir de los Sistemas Basados en Reglas Difusas Evolutivos Propuestos

Para poner en práctica la tarea comentada en la sección anterior, vamos a emplear el conjunto de datos presentado en [NIT97]. Dicho conjunto está compuesto por 105 vectores de datos que recogen valoraciones subjetivas de las seis variables en cuestión, efectuadas por expertos sobre otros tantos tipos de arroz cultivados en Japón (como por ejemplo el Sasanishiki, 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, por lo que toman valores en el intervalo real $[0, 1]$.

Para la resolución de este último problema, vamos a proceder del mismo modo en el que los autores lo hacen en los trabajos [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 cualitativos 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. En concreto, vamos a generar diez particiones distintas compuestas por 75 datos en el conjunto de entrenamiento y 30 en el de prueba. El tamaño de las particiones coincide, por tanto, con el considerado en [INT+94, NIT97].

Por otro lado, puesto que en este caso el objetivo es, no sólo obtener la estimación más precisa posible, sino que el modelo obtenido finalmente sea interpretable, vamos a trabajar únicamente con procesos que permitan la generación de modelos cualitativos. Consideraremos dos de los tres SBRDEs de este tipo presentados en esta memoria, así como el

proceso de aprendizaje inductivo presentado por Nozaki y sus colegas en [NIT97]. Con ello, los procesos de aprendizaje considerados son los siguientes:

- D1.** El proceso inductivo de aprendizaje presentado en [NIT97].
- D2.** El SBRDE en dos etapas compuesto por el algoritmo de WM presentado en el Apéndice B, como primera fase, y el proceso genético de ajuste de BCs descriptivas de tipo Mamdani introducido en la sección 4.4 del capítulo 4, como segunda.
- D3.** El SBRDE en tres etapas para el aprendizaje de BCs descriptivas de tipo Mamdani presentado en el capítulo 4.

El hecho de no considerar el tercero de los SBRDEs para el aprendizaje de BCs descriptivas de tipo Mamdani con el que hemos trabajado en esta memoria, el basado en el proceso evolutivo de aprendizaje de BRLs de Thrift, se debe a que, como hemos podido observar en los experimentos realizados en las dos secciones anteriores, dicho proceso suele obtener un número excesivo de reglas en los modelos cualitativos que genera. Puesto que el problema con el que estamos trabajando presenta cinco variables de entrada, la gran cantidad de combinaciones posibles en el antecedente daría lugar a que se obtuvieran BRLs realmente intratables a partir del proceso evolutivo de aprendizaje comentado.

Al igual que se hacía en [NIT97], hemos considerado las mismas particiones difusas para las seis variables manejadas. Dichas particiones se han obtenido mediante un proceso de normalización en el que se ha dividido sucesivamente el universo de discurso de cada variable en 2, 3, 4 y 5 partes iguales, y se ha asociado a cada una de ellas un conjunto difuso triangular.

Los parámetros empleados en la ejecución de los dos SBRDEs considerados son los que se muestran a continuación:

- **SBRDE D2:**

- *Proceso genético de ajuste de BCs descriptivas:* Tamaño de la población: $N = 61$; valores de las probabilidades de cruce y mutación: $P_c = 0.6$ y $P_m = 0.1$ (por individuo); número de generaciones: $Gen = 1000$; valores asociados al operador de cruce max-min-aritmético y a la mutación no uniforme: $a = 0.35$ y $b = 5$.

- **SBRDE D3:**

- *Proceso iterativo de generación de reglas lingüísticas:* Valor de los parámetros considerados en la función de adaptación: $\epsilon = 1.5$, $\omega = 0.05$ y $k = 0.1$.
- *Proceso genético de multisimplificación:* $\tau = 0.1$, $N = 61$, $P_c = 0.6$, $P_m = 0.1$ (por individuo), $Gen = 500$ y número de soluciones a generar: 1.

Tabla 7.7. Resultados obtenidos en la valoración de la calidad del arroz

| <i>N et.</i> | D1 | | | D2 | | | D3 | | |
|--------------|-----------|-------------|-------------|-----------|-------------|-------------|-----------|-------------|-------------|
| | <i>#R</i> | EC_{entr} | EC_{prue} | <i>#R</i> | EC_{entr} | EC_{prue} | <i>#R</i> | EC_{entr} | EC_{prue} |
| 2 | 32 | 0.00862 | 0.00985 | 15 | 0.00110 | 0.00213 | 6 | 0.00107 | 0.00237 |
| 3 | 243 | 0.00251 | 0.00322 | 23 | 0.00073 | 0.00460 | 10.6 | 0.00069 | 0.00301 |
| 4 | 1024 | 0.00133 | 0.00208 | 29.5 | 0.00089 | 0.00272 | 20.5 | 0.00087 | 0.00273 |
| 5 | 3125 | 0.00082 | 0.00210 | 42.9 | 0.00044 | 0.01440 | 37.3 | 0.00042 | 0.01061 |

– *Proceso genético de ajuste de BCs descriptivas*: los mismos que en la misma etapa del SBRDE **D2**.

Antes de presentar los resultados obtenidos por los tres procesos de diseño en la resolución del problema y plantear un pequeño estudio comparativo, es necesario analizar la composición del método **D1**, propuesto por Nozaki y sus colegas en [NIT97]. Dicho proceso de aprendizaje emplea un parámetro, α , que define una función no lineal de escalado (del tipo de las comentadas en la sección 2.3.2.1 del capítulo 2) sobre las entradas y las salidas del sistema. Puesto que los dos SBRDEs considerados, **D2** y **D3**, no incluyen esta posibilidad, hemos decidido no tenerla en cuenta en las ejecuciones realizadas del proceso **D1**, con objeto de que la comparativa efectuada sea lo más realista posible. Por esta razón, hemos considerado el valor $\alpha = 1$ para dichas ejecuciones, con lo que no se aplica ningún tipo de escalado en los SBRDs diseñados.

Los resultados obtenidos en la experimentación realizada están recogidos en la tabla 7.7, en la que *N et.* representa el número de etiquetas considerado en las particiones difusas de las seis variables. Los valores contenidos en las columnas EC_{entr} y EC_{prue} han sido calculados haciendo la media de los errores cuadráticos cometidos por los diez modelos cualitativos generados en cada caso en la aproximación de los conjuntos de entrenamiento y prueba, respectivamente. La columna notada por *#R* contiene el número medio de reglas que componen las BCs de dichos modelos.

7.2.3 Conclusiones

A continuación presentamos las conclusiones obtenidas en la última de las aplicaciones consideradas en este capítulo, la valoración de la calidad del arroz:

- En lo que respecta a los resultados globales, el proceso inductivo de aprendizaje de Nozaki y sus colegas, **D1**, ha obtenido el mejor resultado en media con respecto al nivel de generalización de los distintos conjuntos de prueba, 0.00208, en el caso en el que se consideran 4 etiquetas en las particiones difusas asociadas a las seis

variables del problema. En este aspecto, las diferencias no han sido excesivamente significativas, ya que los SBRDEs **D2** y **D3** han obtenido, respectivamente, los valores 0.00213 y 0.00237; ambos al trabajar con 2 etiquetas.

En cambio, el mejor nivel de aproximación de los conjuntos de entrenamiento, 0.00042, ha sido obtenido por el SBRDE para el aprendizaje de BCs descriptivas de tipo Mamdani presentado en el capítulo 4, **D3**, al trabajar con 5 etiquetas. En este caso, las diferencias no son significativas con respecto al otro SBRDE considerado, **D2**, que ha obtenido el valor 0.00044 al trabajar con 5 etiquetas, pero si en comparación con el proceso **D1**, que sólo logra presentar el valor 0.00082, también con 5 etiquetas.

De todos modos, para analizar de un modo más realista estos resultados, debemos tener en cuenta no sólo la precisión de los modelos cualitativos generados, sino también su complejidad, es decir, el número de reglas lingüísticas que componen sus BCs. Recordemos que el objetivo que se perseguía en esta aplicación no era únicamente obtener un modelo con buen rendimiento, sino conseguir que dicho modelo fuera interpretable por el ser humano. Por esta razón, aunque el proceso de aprendizaje **D1**, al trabajar con 4 y 5 etiquetas, permite generar los modelos con mejor nivel de generalización, éstos no serían útiles en la práctica, ya que sería imposible interpretar un modelo cualitativo cuya BC estuviese compuesta por 3125 reglas o, incluso, por 1024.

En cambio, los modelos cualitativos generados mediante los SBRDEs **D2** y **D3**, al trabajar con 2 etiquetas, son muy adecuados ya que su nivel de generalización es muy similar al de los modelos generados a partir de **D1**, y su complejidad es mucho menor (15 y 6 reglas, en media, respectivamente). Estos modelos sí pueden ser interpretados en la práctica por el ser humano.

- Por otro lado, analizando individualmente cada experimentación, podemos destacar que el proceso de Nozaki, **D1**, obtiene el mejor resultado con respecto al nivel de generalización en dos ocasiones, al trabajar con 4 y 5 etiquetas. En cambio, para las otras dos combinaciones obtiene el peor resultado de los tres procesos estudiados, tanto en el nivel de generalización como en el de aproximación, en el que es el peor en todas las ocasiones, a pesar de presentar siempre el número de reglas más elevado.

El primero de los SBRDEs considerados, **D2**, logra el mejor comportamiento con respecto al nivel de generalización en una ocasión, al trabajar con 2 etiquetas, pero nunca es el mejor en la aproximación, ya que en los cuatro casos obtiene el segundo mejor resultado.

Por último, nuestro SBRDE, **D3**, alcanza el mejor nivel de generalización solamente en un caso, con 3 etiquetas, pero es el que mejor aproxima al conjunto de entrenamiento en las cuatro combinaciones consideradas. Este hecho es bastante curioso al ser el proceso que genera los modelos cualitativos más simples, es decir, con menor número de reglas, de los tres considerados.

- Finalmente, el comportamiento general de los tres procesos en función del número de etiquetas considerado ha sido bastante similar en lo que respecta al nivel de aproximación, pero ha variado algo más en el nivel de generalización. Todos ellos obtienen el mejor valor para el primer caso al trabajar con 5 etiquetas (0.00082, 0.00044 y 0.00042 para **D1**, **D2** y **D3**, respectivamente) y el segundo mejor al trabajar con 4 etiquetas, el primero, y con 3, los otros dos. Parece ser que el empleo de un número elevado de etiquetas beneficia a esta característica.

En cambio, en lo que respecta al nivel de generalización, el comportamiento es algo más variado. El primer proceso obtiene su mejor valor (0.00208) considerando 4 etiquetas, mientras que los dos SBRDEs lo alcanzan cuando hacen uso únicamente de 2 etiquetas (0.00213 y 0.00237, respectivamente).

Comentarios Finales

Resultados Obtenidos y Conclusiones

Hemos presentado una metodología evolutiva de aprendizaje de bases de reglas difusas junto con cuatro SBRDEs concretos basados en la misma. Los procesos evolutivos de aprendizaje propuestos son capaces de llevar a cabo el diseño de SBRDs de los distintos tipos existentes a partir de la información numérica y experta disponible.

Los resultados obtenidos en esta memoria y algunas conclusiones sobre los mismos se resumen en los siguientes apartados:

A. Sistemas Basados en Reglas Difusas Evolutivos

Se ha realizado un estudio de los distintos SBRDEs [CH95, CHHVL96], métodos de aprendizaje automático de bases de reglas difusas basados en el empleo de AEs, analizando los distintos tipos existentes junto con sus principales aspectos. Asimismo, hemos estudiado algunos procesos concretos pertenecientes a cada una de las familias existentes. De este estudio hemos extraído las siguientes conclusiones:

1. El aspecto más destacable de los SBRDEs es su capacidad para combinar las buenas características de la técnica de diseño, el AE, y del sistema que está siendo diseñado, el SBRD; así como su capacidad de incorporar la información disponible en el proceso evolutivo para mejorar el proceso de diseño.
2. El problema principal a resolver en el campo de los SBRDEs es el problema de cooperación-competición (PCC), que se plantea el modo en el que se combinan las características comentadas en el punto anterior; es decir, el modo en que se establece la competición entre los individuos de la población para obtener el mejor nivel de cooperación posible en la base de reglas difusas generada finalmente.
3. La resolución del PCC depende en gran medida del esquema de aprendizaje evolutivo en que esté basado el SBRDE. Mientras que es muy complicado de resolver haciendo uso del enfoque Michigan, tanto el enfoque Pittsburgh como el enfoque de Aprendizaje Iterativo de Reglas (AIR) permiten resolverlo de un modo más sencillo.
4. Existen distintos tipos de SBRDEs según qué componentes de la base de reglas difusas se consideren en el proceso de aprendizaje. De este modo, podemos distinguir

entre SBRDEs para la definición evolutiva de la Base de Datos, para el aprendizaje evolutivo de la Base de Reglas lingüísticas y para el aprendizaje evolutivo de la Base de Conocimiento completa.

5. Existen una serie de aspectos de diseño que han de ser afrontados para construir un SBRDE y que condicionan directamente el comportamiento del mismo. Dichos aspectos son, entre otros, la elección del esquema de representación empleado, el diseño de los operadores evolutivos, el diseño de la función de adaptación, la especificación del modo en que se empleará el conocimiento disponible y la definición del esquema de formación de nichos.

B. Metodología Evolutiva de Aprendizaje de Bases de Reglas Difusas

El capítulo 3 de esta memoria recoge los aspectos básicos del paradigma evolutivo de aprendizaje de bases de reglas difusas propuesto [CdJHL97b]. A continuación resumimos dichos aspectos:

1. La metodología propuesta se basa en el enfoque AIR, con lo que los procesos evolutivos de aprendizaje construidos a partir de la misma serán SBRDEs en varias etapas.
2. Trata de aprovechar al máximo el conocimiento disponible sobre el sistema a diseñar, permitiendo que los SBRDEs construidos a partir de la misma incorporen este conocimiento de una forma sencilla y presenten distintas formas de uso dependiendo del tipo y la cantidad de información disponible.
3. Considera la satisfacción de las propiedades de completitud, cubrimiento y consistencia por parte de las bases de reglas difusas generadas, con objeto de conseguir que éstas presenten mejor comportamiento.
4. Presenta una amplia gama de mecanismos para mejorar la resolución del PCC tales como:
 - Afrontar parte de la cooperación entre las reglas generadas en la primera etapa del aprendizaje empleando criterios basados en nichos [CH96a, CH96b, CH96c, CH97b, CH97c] o en medidas locales de error [CH97a, CH97d].
 - Permite que la primera etapa del aprendizaje, el proceso de generación de reglas difusas, pueda ser implementada mediante un algoritmo inductivo [CHL96c, CH97c] o cualquier tipo de AE [CH96a, CH97c], y no sólo por medio de un AG [CH96b, CH96c, CH97b, CH97d]. Gracias a esto, es posible agilizar la ejecución de este proceso en el caso en que el espacio de búsqueda no sea excesivamente grande.

- Considerar dos procesos evolutivos de postprocesamiento en lugar de solamente uno. De este modo, el postprocesamiento incluye tanto la simplificación de la base de reglas difusas generada en la primera etapa del aprendizaje, como el ajuste de la forma de las funciones de pertenencia, lo que da lugar a que los SBRDEs basados en esta filosofía puedan obtener mejores soluciones.
- Diseñar distintas definiciones de la base de reglas difusas a lo largo del proceso evolutivo de aprendizaje, con el objeto de devolver como salida la que mejor nivel de cooperación presente al finalizar éste. Esta tarea se lleva a cabo mediante un esquema de nichos distinto considerado en el proceso genético de multisimplificación [CH96c].
- Emplear los esquemas de representación y operadores evolutivos más adecuados para el tipo de información que se esté manejando, con objeto de mejorar al máximo posible el proceso de búsqueda efectuado por el AE. Una posibilidad novedosa es la de emplear una EE-(1 + 1) como un operador más de un AG buscando el refinar las mejores soluciones en cada iteración [CH96b, CH96c, CH97b, CH97d].
- Hacer uso de la información disponible para generar de un modo adecuado la población inicial de individuos con la que trabajará el AE [CHL96c, CH96a, CH96b, CH96c, CH97a, CH97b, CH97c, CH97d].

C. Sistemas Basados en Reglas Difusas Evolutivos Construidos

Haciendo uso del modelo evolutivo de aprendizaje presentado, hemos construido cuatro SBRDEs que posibilitan el diseño de todos los tipos de SBRDs existentes que manejan entradas y salidas reales, los descriptivos y aproximativos de tipo Mamdani y los de tipo TSK. Hemos estudiado el comportamiento de dichos procesos en el modelado de cuatro funciones tridimensionales con distintas características y hemos comparado su funcionamiento con el de otros SBRDEs pertenecientes a las mismas familias. Los distintos modelos generados por nuestros procesos han demostrado un buen comportamiento.

Resumimos brevemente los procesos evolutivos de aprendizaje propuestos en los párrafos siguientes:

1. *El SBRDE para el diseño de SBRDs descriptivos de tipo Mamdani* [CHL96c, CH97c]: Este primer SBRDE se basa en el empleo de una estructura especial de BC que permite la existencia de distintas reglas lingüísticas con el mismo antecedente y el mismo o distintos consecuente. Este hecho posibilita la obtención de modelos cualitativos que mejoran el grado de aproximación del sistema real, manteniendo al mismo tiempo el nivel de descripción del modelo.

Está compuesto por tres etapas, el proceso de generación de reglas difusas, que incluye a su vez el método iterativo de cubrimiento y el método de generación de reglas,

implementado por medio de un algoritmo inductivo; el proceso genético de multisimplificación y el proceso genético de ajuste de BCs descriptivas, implementado por medio de un AG con codificación real.

La funcionalidad de este SBRDE se centra en el modelado cualitativo de sistemas con un nivel de complejidad bajo. Es útil en aquellos casos en los que el requisito principal del modelado es la interpretabilidad del modelo, pero el modelo cualitativo obtenido no presenta el grado de precisión deseado. Gracias a este SBRDE es posible generar modelos que aproximen mejor el sistema real, al mismo tiempo que mantienen un nivel de interpretabilidad aceptable.

2. *El SBRDE para el aprendizaje de BRDs aproximativas de tipo Mamdani según el enfoque de semántica libre no restringida* [CH96a, CH97c]: Compuesto por un proceso de generación de reglas difusas basado en el método iterativo de cubrimiento y el método de generación de reglas, implementado por medio de un algoritmo inductivo y una EE-(1 + 1); por el proceso genético de multisimplificación y por un proceso genético de ajuste de BRDs aproximativas, que hace uso de un AG con codificación real.

Puesto que las únicas restricciones impuestas en la generación de las funciones de pertenencia manejadas en las reglas difusas aproximativas son que estén definidas dentro de un intervalo global concreto y que presenten una forma coherente, este segundo proceso evolutivo de aprendizaje está aconsejado para el modelado difuso de sistemas que presenten una complejidad muy alta, en los que un modelo cualitativo no permite obtener el nivel de precisión deseado.

3. *El SBRDE para el aprendizaje de BRDs aproximativas de tipo Mamdani según el enfoque de semántica libre restringida* [CH96b, CH96c, CH97b]: Su composición mantiene las dos últimas fases del proceso anterior. El método de generación de reglas difusas aproximativas se pone en práctica, en este caso, haciendo uso de un algoritmo híbrido entre un AG y una EE-(1 + 1).

El área de aplicación de este SBRDE basado en el enfoque de generación de semántica libre restringida la constituye el modelado difuso de sistemas con un nivel de complejidad intermedio. En estos casos, el requisito principal vuelve a ser la precisión del modelo y no su poder de descripción.

4. *El SBRDE para el diseño de SBRDs de tipo TSK* [CH97a, CH97d]: Este proceso está formado únicamente por dos etapas: un proceso de generación de reglas TSK, compuesto a su vez por un método iterativo de generación de antecedentes y un método de aprendizaje de consecuentes TSK, basado en una EE - (μ, λ) ; y un proceso evolutivo de refinamiento, implementado mediante un AE híbrido entre un AG y una EE - (1 + 1).

El último de los SBRDEs propuestos está recomendado para problemas de modelado difuso en los que se dan dos circunstancias: el sistema a modelar es de baja complejidad y el requisito principal del modelado es la precisión del modelo generado.

Debido a esta razón, es el más adecuado para sustituir al SBRDE para el aprendizaje de BCs descriptivas en aquellos casos en los que no se persiga la obtención de un modelo interpretable.

El siguiente cuadro resume el ámbito de aplicación de los cuatro SBRDEs propuestos en función del grado de descripción que se desee alcanzar en el modelado y de la complejidad del sistema a modelar:

Tabla 7.8. Ámbito de aplicación de los cuatro SBRDEs propuestos

| Grado de descripción | Complejidad del sistema | | |
|-------------------------|-------------------------|------------------------------------|---------------------------------------|
| | BAJA | MEDIA | ALTA |
| ALTO (Mod. cualitativo) | BCs descr. | — | — |
| BAJO (Mod. difuso) | BCs TSK | BRDs aprox. sem. libre restringida | BRDs aprox. sem. libre no restringida |

D. Aplicaciones

Hemos considerado tres aplicaciones reales de modelado que precisaban de distintos requisitos para evaluar el rendimiento de los distintos SBRDEs propuestos. Además de los cuatro procesos evolutivos de aprendizaje presentados en esta memoria, hemos aplicado otros cinco (dos para el diseño de SBRDs descriptivos de tipo Mamdani, dos para el diseño de SBRDs aproximativos de tipo Mamdani y uno para el de SBRDs TSK) en la resolución de los problemas comentados. Las conclusiones obtenidas en cada aplicación han sido las siguientes:

1. *Modelado de zonas rurales* [CHS97]: En este primer problema, en el que el requisito fundamental era la precisión del modelo difuso obtenido, el SBRDE para el aprendizaje de BCs TSK propuesto en esta memoria ha permitido la generación del mejor modelo, tanto en la aproximación del conjunto de entrenamiento como en el poder de predicción sobre el conjunto de prueba.

En todos los casos, los SBRDEs propuestos en esta memoria han obtenido el mejor modelado en comparación con los restantes procesos evolutivos de aprendizaje considerados para cada uno de los tipos de SBRDs existentes. Cabe destacar la obtención de un modelo cualitativo con buen comportamiento y una BC muy simple, 19 reglas, mediante nuestro SBRDE para el diseño de SBRDs descriptivos de tipo Mamdani.

2. *Modelado de zonas urbanas*: El mejor modelo obtenido para este problema se ha generado, de nuevo, a partir del SBRDE para el aprendizaje de BCs TSK presentado

en el capítulo 6, aunque en esta ocasión considerando distintos valores para los parámetros de los que se hace uso en su ejecución.

También hemos podido observar que esta segunda aplicación considerada en el campo de la Ingeniería Eléctrica presentaba un nivel de complejidad menor que la anterior, por lo que los procesos evolutivos de aprendizaje para el diseño de SBRDs descriptivos de tipo Mamdani han obtenido mejores resultados que los que permitían generar SBRDs aproximativos del mismo tipo. Dentro de este grupo, el SBRDE propuesto en el capítulo 4 de esta memoria ha sido el que mejor comportamiento ha demostrado. Además, se ha constatado en la práctica la coherencia de los parámetros empleados en el mismo.

3. *Valoración de la calidad del arroz:* El SBRDE para el aprendizaje de BCs descriptivas presentado en el capítulo 4 nos ha permitido solucionar adecuadamente este tercer problema, en el que no sólo se perseguía la obtención de un modelo preciso del sistema, sino también que dicho modelo fuera interpretable. Mediante este SBRDE, hemos generado modelos cualitativos que, además de presentar un nivel de generalización adecuado, han resultado ser los más simples en todos los casos, característica que facilita su interpretación por parte del ser humano.

Por otro lado, estos modelos han demostrado también el mejor comportamiento en lo que respecta al nivel de aproximación de los conjuntos de entrenamiento, lo que constituye un hecho curioso al ser, precisamente, los que menos reglas lingüísticas contenían en su BC de entre todos los obtenidos a partir de los tres procesos de diseño considerados.

Trabajos Futuros

A continuación, discutiremos alguna líneas de trabajo que quedan aún abiertos sobre los temas tratados en la memoria. Además, consideraremos las extensiones sobre los modelos propuestos que serán objeto de futuros estudios.

Sobre el Criterio de Baja Interacción Entre Nichos Considerado en el Aprendizaje de Bases de Reglas Difusas Aproximativas

Como comentábamos en el capítulo 3, el criterio de baja interacción entre nichos, BIEN, tiene una gran importancia en la metodología propuesta debido a que permite afrontar parte del problema de cooperación entre reglas en la primera etapa del aprendizaje de BRDs aproximativas de tipo Mamdani. En esta memoria hemos hecho uso de una implementación posible de dicho criterio, basada en estudiar el cubrimiento que la regla que está siendo generada en el momento actual produce sobre los centros de las reglas obtenidas anteriormente y penalizar a la regla actual cuando dicho cubrimiento sea excesivo.

Puesto que la filosofía general del criterio posibilita el afrontar su implementación por medio de otras vías distintas nos proponemos analizar éstas y diseñar otras implementaciones de dicho criterio que permitan mejorar el comportamiento del proceso evolutivo de

aprendizaje. En concreto, nos plantearemos la obtención de distintos criterios concretos que presenten una gradualidad en el nivel de penalización que aplican sobre la regla actual.

Sobre la Aplicación de la Metodología para el Diseño de un Nuevo Tipo de Sistema Basado en Reglas Difusas

Tal y como hemos comentado en la sección anterior, el SBRDE para el aprendizaje de BCs TSK presenta un buen rendimiento cuando trabaja en problemas de modelado difuso en los que el sistema a modelar es de baja complejidad. Puesto que este tipo de SBRDs tienen asociado un nivel de interpretabilidad bajo debido a la estructura del consecuente en la regla difusa que manejan, puede parecer interesante disminuir aún más este nivel de descripción, siempre que se logre una ganancia en el rendimiento del modelo obtenido. De este modo, se podría construir un nuevo tipo de SBRDE que pudiera ser aplicado en problemas de modelado difuso de alta complejidad.

Nos proponemos poner en práctica esta idea como trabajo futuro. Para ello, nos planteamos la construcción de un SBRDE que, siguiendo la metodología propuesta, permita diseñar un nuevo tipo de SBRDs, el SBRD aproximativo de tipo TSK. Este SBRD estará basado en una estructura de regla difusa que considerará conjuntos difusos en lugar de etiquetas lingüísticas en el antecedente (siguiendo la filosofía de los SBRDs aproximativos de tipo Mamdani) y una combinación lineal de las entradas en el consecuente (del mismo modo que los SBRDs de tipo TSK).

Sobre la Extensión de la Metodología para el Manejo de Problemas de Clasificación Difusa

El paradigma evolutivo de aprendizaje y los SBRDEs propuestos han presentado un buen comportamiento en su aplicación a problemas en los que la variable de salida del sistema a modelar es real. Existen otro tipo de problemas en los que dicha variable únicamente puede tomar como valor una serie de elementos pertenecientes a un conjunto de valores posibles. Estos problemas suelen ser denominados *problemas de clasificación* y, en ellos, la variable de salida se encarga de asociar una determinada clase, de entre todas las definidas en el problema, a una serie de objetos que presentan unas características concretas. Cuando se afronta la resolución de estos problemas por medio de un SBRD, esta área de trabajo recibe el nombre de *clasificación difusa*.

Debido a la naturaleza especial de la variable de salida en los problemas comentados, la metodología propuesta no puede ser aplicada directamente para su resolución, aunque gran parte de las ideas presentadas son también útiles en este campo. Debido a esta razón, nos proponemos extender el ámbito de aplicación de la misma para que permita la construcción de SBRDEs para el diseño de SBRDs que resuelvan problemas de clasificación. Ya en [CdJH96, CdJHL97a] hemos realizado las primeras aproximaciones a este problema. En particular, hemos modificado el SBRDE para el aprendizaje de BCs descriptivas de tipo Mamdani, presentado en el capítulo 4 de esta memoria, y lo hemos aplicado a distintos problemas de clasificación, obteniendo resultados prometedores. Por otro lado, en el

segundo de los trabajos citados, hemos comenzado a plantear la extensión del paradigma propuesto, considerando la integración de nuevos métodos de razonamiento para problemas de clasificación difusa en la segunda etapa del aprendizaje.

Apéndice A

Funciones Tridimensionales Consideradas para su Modelado en la Experimentación Realizada

En este primer Apéndice presentaremos una serie de funciones tridimensionales que hemos escogido para ser modeladas mediante los SBRDs de distintos tipos obtenidos a partir de los SBRDEs basados en la metodología propuesta en esta memoria y presentados en los capítulos 4, 5 y 6 de la misma. Estas funciones han sido seleccionadas por tener asociados distintos niveles de complejidad que las hacen más o menos adecuadas para cada tipo concreto de SBRD. Describiremos también la forma en la que ha sido obtenida la información numérica necesaria para generar los distintos modelos, así como algunos aspectos generales de la experimentación realizada.

A continuación presentamos las cuatro funciones, así como los intervalos en los que están definidas. Las dos primeras funciones, F_1 , denominada *modelo esférico*, y F_2 , la cual presenta dos discontinuidades en los puntos $(0, 0)$ y $(1, 1)$, son funciones unimodales sencillas. La tercera, F_3 , es una función multimodal de complejidad intermedia. Finalmente, la *función generalizada de Rastrigin*, F_4 , es una función fuertemente multimodal muy compleja. Las características comentadas pueden ser observadas en sus representaciones gráficas mostradas en la figura A.1.

$$F_1(x_1, x_2) = x_1^2 + x_2^2, \\ x_1, x_2 \in [-5, 5], F_1(x_1, x_2) \in [0, 50]$$

$$F_2(x_1, x_2) = 10 \cdot \frac{x_1 - x_1 x_2}{x_1 - 2x_1 x_2 + x_2}, \\ x_1, x_2 \in [0, 1], F_2(x_1, x_2) \in [0, 10]$$

$$F_3(x_1, x_2) = e^{x_1} \cdot \sin^2 x_2 + e^{x_2} \cdot \sin^2 x_1, \\ x_1, x_2 \in [-8, 8], F_3(x_1, x_2) \in [0, 5836]$$

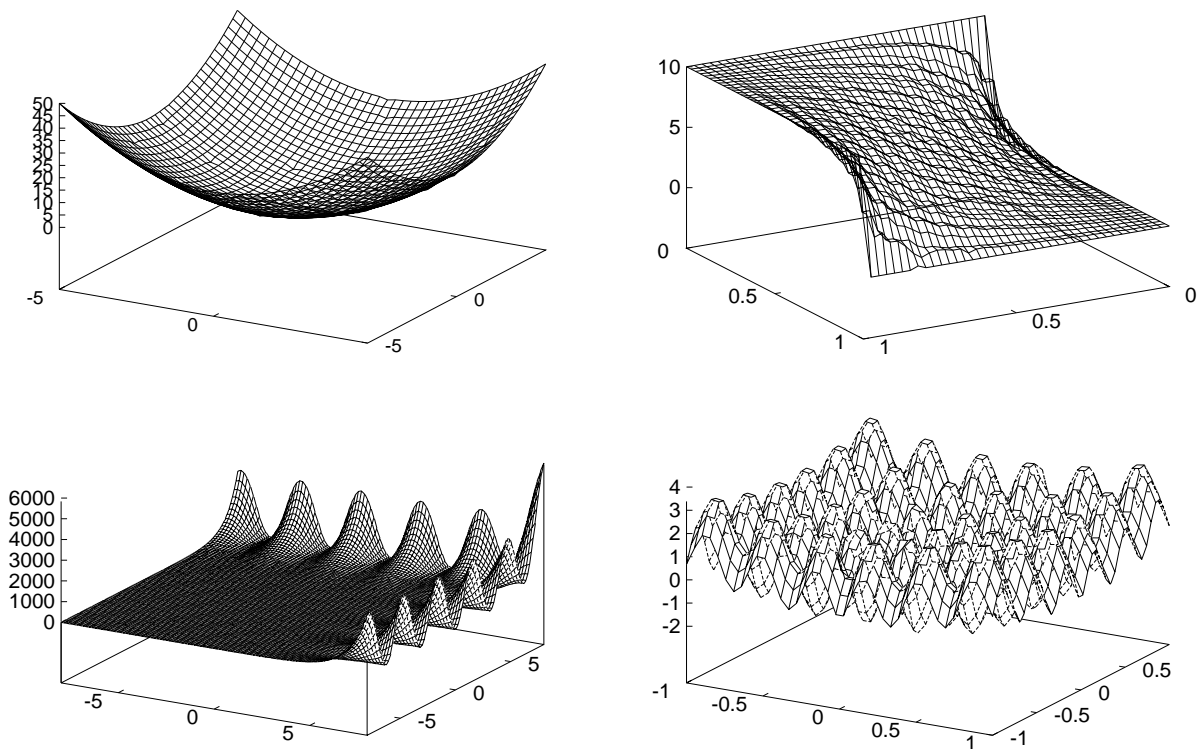


Figura A.1. Representaciones gráficas de las funciones F_1 (arriba a la izquierda), F_2 (arriba a la derecha), F_3 (abajo a la izquierda) y F_4 (abajo a la derecha)

$$F_4(x_1, x_2) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2),$$

$$x_1, x_2 \in [-1, 1], F_4(x_1, x_2) \in [2, 3.5231]$$

Para cada función, hemos obtenido experimentalmente un conjunto de entrenamiento distribuido de forma uniforme en el espacio tridimensional de definición. De este modo, se han generado cuatro conjuntos formados por 1681, 674, 1089 y 1681 datos para las funciones F_1 a F_4 , tomando respectivamente 41, 26, 33 y 41 valores uniformemente distribuidos en los universos de discurso respectivos de las variables de entrada y calculando el valor asociado en la variable de salida mediante la expresión de la función. Como puede observarse, el conjunto de entrenamiento de la función F_2 está formado por 674 valores, en lugar de por 676, debido a que dicha función no está definida en dos puntos del espacio.

Con objeto de evaluar el nivel de generalización de los SBRDs diseñados sin que exista ningún sesgo debido a la composición de los conjuntos de entrenamiento, se han generado otros cuatro conjuntos de datos para ser empleados como conjuntos de prueba. El tamaño de estos nuevos conjuntos es un porcentaje de los anteriores, en concreto, un diez por ciento. Los datos han sido obtenidos generando aleatoriamente los valores de las variables

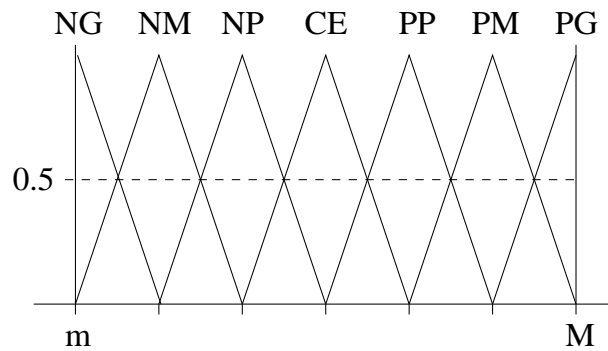


Figura A.2. Partición difusa considerada en los experimentos realizados

de entrada en sus intervalos respectivos y calculando el valor de la variable de salida a partir de la expresión de la función. Con ello, emplearemos los cuatro conjuntos, formados respectivamente por 168, 67, 108 y 168 datos aleatorios, para evaluar el rendimiento de los SBRDs generados calculando el error cuadrático medio cometido por éstos sobre dichos conjuntos.

En todos los experimentos realizados, se ha considerado que no se dispone de conocimiento experto sobre el problema a resolver. De este modo, la definición inicial de las funciones de pertenencia se ha efectuado mediante un proceso de normalización. Cada espacio de entrada y salida ha sido particionado en siete partes iguales. En el caso de trabajar con SBRDs descriptivos de tipo Mamdani o TSK, hemos asociado a cada una de ellas una etiqueta lingüística del conjunto $\{NG, NM, NP, CE, PP, PM, PG\}$ (con el siguiente significado: $NG = \text{Negativo Grande}$, $NM = \text{Negativo Medio}$, $NP = \text{Negativo Pequeño}$, $CE = \text{Cero}$, $PP = \text{Positivo Pequeño}$, $PM = \text{Positivo Medio}$ y $PG = \text{Positivo Grande}$) y un conjunto difuso triangular que especifica la semántica de ésta. La figura A.2 recoge una partición difusa de este tipo. En cambio, cuando el SBRDE se emplea para diseñar SBRDs aproximativos de tipo Mamdani, se considera directamente la partición difusa formada por los siete conjuntos difusos comentados como definición inicial de las funciones de pertenencia empleadas, sin tener en cuenta ningún tipo de etiqueta lingüística.

Apéndice B

El Proceso Inductivo de Generación de Reglas Difusas de Wang y Mendel

En este apéndice introduciremos brevemente el funcionamiento del proceso inductivo de generación de reglas difusas presentado por Wang y Mendel en [WM92]. Dicho proceso se basa en la existencia de un conjunto de datos de entrada-salida que reflejen el comportamiento del problema a resolver y de una definición previa de la BD formada por las particiones difusas de los espacios de entrada y salida del problema en cuestión. La estructura de regla difusa que maneja es la habitual regla lingüística con n entradas y una salida con la forma

$$\text{Si } X_1 \text{ es } A_1 \text{ y } \dots \text{ y } X_n \text{ es } A_n \text{ entonces } Y \text{ es } B,$$

donde X_i e Y son variables lingüísticas de entrada y salida respectivamente, y los A_i y B son etiquetas lingüísticas asociadas con conjuntos difusos que determinan su semántica.

De este modo, las bases de reglas difusas generadas a partir del proceso de Wang y Mendel presentan un comportamiento descriptivo y el número de reglas que las componen está acotado superiormente por el producto del número de términos lingüísticos asociados a las variables de entrada del sistema.

La generación de la BRL se efectúa mediante la aplicación de los tres pasos siguientes:

1. *Generar un conjunto preliminar de reglas lingüísticas:* Este conjunto estará formado por la regla que mejor cubra a cada ejemplo contenido en el conjunto de datos de entrada-salida. La composición de estas reglas se obtiene tomando un ejemplo concreto, es decir, un vector de valores reales con dimensión $n + 1$ (n valores de entrada y uno de salida) y asignando a cada variable lingüística la etiqueta existente en el conjunto de términos de la misma que tenga asociado el conjunto difuso con el que el componente del vector tenga un mayor grado de pertenencia.

2. *Asignar un grado de importancia a cada regla:* Sea la regla $R = \text{Si } X_1 \text{ es } A \text{ y } X_2 \text{ es } B \text{ entonces } Y \text{ es } C$ generada a partir del ejemplo (x_1, x_2, y) , el grado asociado a la misma se obtiene de la forma:

$$G(R) = \mu_A(x_1) \cdot \mu_B(x_2) \cdot \mu_C(y)$$

3. *Construir una BRL final a partir de las reglas anteriores:* En el caso en que todas las reglas existentes en el conjunto preliminar que presenten la misma combinación de antecedentes tengan asociado el mismo consecuente, dicha regla es automáticamente insertada (una sola vez) en la BRL final. En cambio, en el caso en que existan reglas conflictivas, es decir, reglas con el mismo antecedente y distintos valores en el consecuente, la regla insertada será aquella que presente mayor grado de importancia.

Apéndice C

El Algoritmo de Conteo Ponderado para la Generación de Reglas Difusas Aproximativas de tipo Mamdani

Este tercer apéndice está dedicado a describir el Algoritmo de Conteo Ponderado (ACP) presentado en [BD95], el cual permite generar una BRD aproximativa de tipo Mamdani, en la que las reglas manejan conjuntos difusos triangulares, a partir de un conjunto de ejemplos E_p . Para ello, se basa en el principio de generar primero los antecedentes de las reglas y obtener a posteriori los consecuentes con ayuda del subconjunto de datos de entrenamiento en los que dicha regla es aplicable, es decir, de los ejemplos positivos para el antecedente de la regla en cuestión. Suponiendo que los universos de discurso asociados a todas las variables del sistema son continuos, el algoritmo trabaja generando la forma de los conjuntos difusos empleados en los antecedentes, A_{ik} , a partir de una definición previa de los soportes de éstos, e identificando posteriormente los consecuentes correspondientes procediendo del siguiente modo:

1. Se define el soporte (a_{ik}^-, a_{ik}^+) de los conjuntos difusos A_{ik} que pueden formar parte del antecedente de las reglas de la BRD.
2. Para cada regla R_i formada por la combinación de antecedentes A_{ik} , $k = 1, \dots, n$ (donde n es el número de variables de entrada del sistema):
 - (a) Se determina la forma de los k conjuntos difusos A_{ik} , conjuntos difusos triangulares definidos respectivamente por los parámetros $(a_{ik}^-, a_{ik}^1, a_{ik}^+)$, calculando los valores de a_{ik}^1 como la media de las componentes ex_k de todos los ejemplos positivos para el antecedente de la regla i -ésima R_i :

$$a_{ik}^1 = \frac{1}{N_i} \sum_{e \in E_i} ex_k$$

donde $E_i = \{e \in E_p / ex_k \in (a_{ik}^-, a_{ik}^+)\}$ es el subconjunto de E_p formado por los ejemplos positivos para el antecedente de la regla R_i y N_i es el cardinal de dicho conjunto.

- (b) Se calcula el grado de emparejamiento h_j , $j = 1, \dots, N_i$, de los ejemplos contenidos en E_i con el antecedente de la regla R_i generado en el paso anterior.
- (c) Se determina la forma del consecuente de la regla haciendo uso de los ejemplos contenidos en un nuevo subconjunto E_{ϕ_i} formado por aquellos ejemplos de E_i que presentan un grado de emparejamiento mayor que $\phi \in (0, 1]$ con el antecedente de la regla R_i . Dicho consecuente será un conjunto difuso triangular definido por los parámetros (b_i^-, b_i^1, b_i^+) obtenidos de la forma:

$$b_i^- = \min_{e_l \in E_{\phi_i}} e_ly \quad ; \quad b_i^1 = \frac{\sum_{e_l \in E_{\phi_i}} h_l \cdot e_ly}{\sum_{e_l \in E_{\phi_i}} h_l} \quad ; \quad b_i^+ = \max_{e_l \in E_{\phi_i}} e_ly$$

Como puede observarse, b_i^- y b_i^+ son respectivamente el menor y el mayor valor de salida, y b_i^1 es la salida media ponderada por el grado de emparejamiento de los ejemplos contenidos en E_{ϕ_i} .

El valor de ϕ debe ser seleccionado de modo que cada regla tenga asociado un número de elementos suficiente en el conjunto de entrenamiento E_{ϕ_i} . Cuanto mayor sea ϕ , menos elementos serán considerados para definir la forma del conjunto difuso que constituye el consecuente de la regla y más precisas serán las reglas generadas.

Bibliografía

- [Bäc96] Bäck T. (1996) *Evolutionary Algorithms in Theory and Practice*. Oxford University Press.
- [Bak87] Baker J. E. (1987) Reducing bias and inefficiency in the selection algorithm. En *Proc. Second International Conference on Genetic Algorithms (ICGA '87)*, páginas 14–21. Hillsdale.
- [Bas94] Bastian A. (Febrero 1994) How to handle the flexibility of linguistic variables with applications. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 3(4): 463–484.
- [BB91] Belew R. K. y Booker L. B. (Eds.) (1991) *Proceedings of the Fourth International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers.
- [BBM93] Beasley D., Bull D. R., y Martin R. R. (1993) A sequential niche technique for multimodal function optimization. *Evolutionary Computation* 1(2): 101–125.
- [BD95] Bardossy A. y Duckstein L. (1995) *Fuzzy Rule-Based Modeling With Application to Geophysical, Biological and Engineering Systems*. CRC Press.
- [Ben94] Benachenhou D. (1994) Smart trading with FRET. En Deboeck G. (Ed.) *Trading on the Edge: Neural, Genetic, and Fuzzy Systems for Chaotic and Financial Markets*, páginas 215–242. John Willey & Sons.
- [Ber92] Berenji H. (1992) Fuzzy logic controllers. En Yager R. y Zadeh L. (Eds.) *An Introduction to Fuzzy Logic Applications in Intelligent Systems*, páginas 69–96. Kluwer Academic.
- [BFGM93] Baglio S., Fortuna L., Graziana S., y Muscato G. (Septiembre 1993) Membership function shape and the dynamical behavior of a fuzzy system. En *Proc. First European Congress on Fuzzy and Intelligent Technologies (EU-FIT'93)*, páginas 645–650. Aachen.
- [BGH89] Booker L. B., Goldberg D. E., y Holland J. H. (1989) Classifier systems and genetic algorithms. *Artificial Intelligence* 40: 235–282.

- [BN95] Bolata F. y Nowé A. (Julio 1995) From fuzzy linguistic specifications to fuzzy controllers using evolution strategies. En *Proc. Fourth IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'95)*, páginas 1089–1094. Yokohama.
- [Bon93] Bonarini A. (Septiembre 1993) ELF: Learning incomplete fuzzy rule sets for an autonomous robot. En *Proc. First European Congress on Fuzzy and Intelligent Technologies (EUFIT'93)*, páginas 69–75. Aachen.
- [Bon94] Bonissone P. P. (1994) Fuzzy logic controllers: an industrial reality. En Zurada J., Marks R., y Robinson C. (Eds.) *Computational Intelligence: Imitating Life*, páginas 316–327. IEEE Press.
- [Bon96] Bonarini A. (1996) Evolutionary learning of fuzzy rules: competition and cooperation. En Pedrycz W. (Ed.) *Fuzzy Modelling: Paradigms and Practice*, páginas 265–283. Kluwer Academic.
- [Bon97] Bonissone P. P. (1997) Soft computing: the convergence of emerging reasoning technologies. *Soft Computing* 1(1): 6–18.
- [Boo82] Booker L. B. (1982) *Intelligent Behaviour as an Adaption to the Task Environment*. PhD thesis, University of Michigan.
- [BS91] Bäck T. y Schwefel H. P. (1991) Extended selection mechanisms in genetic algorithms. En *Proc. Fourth International Conference on Genetic Algorithms (ICGA '91)*, páginas 2–9. San Diego.
- [BS95] Bäck T. y Schwefel H. P. (1995) Evolution strategies I: variants and their computational implementation. En Periaux J., Winter G., Galán M., y Cuesta P. (Eds.) *Genetic Algorithms in Engineering and Computer Science*, páginas 111–126. John Wiley and Sons.
- [Buc93] Buckley J. J. (1993) Sugeno type controllers are universal approximators. *Fuzzy Sets and Systems* 53: 299–304.
- [Cas95] Castro J. L. (Abril 1995) Fuzzy logic controllers are universal approximators. *IEEE Transactions on Systems, Man, and Cybernetics* 25(4): 629–635.
- [Cas96] Castro J. L. y Delgado M. (Febrero 1996) Fuzzy systems with defuzzification are universal approximators. *IEEE Transactions on Systems, Man, and Cybernetics* 26(1): 149–152.
- [CdJH96] Cordon O., del Jesus M. J., y Herrera F. (Septiembre 1996) Sistema de clasificación con reglas difusas utilizando algoritmos genéticos. En *Proc. VI Congreso Español sobre Tecnologías y Lógica Fuzzy (ESTYLF'96)*, páginas 95–100. Oviedo.

- [CdJHL97a] Cordón O., del Jesus M. J., Herrera F., y López E. (Junio 1997) Selecting fuzzy rule-based classification systems with specific reasoning methods using genetic algorithms. En *Proc. Seventh International Fuzzy Systems Association World Congress (IFSA '97)*, volumen 2, páginas 424–429. Praga.
- [CdJHL97b] Cordón O., del Jesus M. J., Herrera F., y Lozano M. (Septiembre 1997) An evolutionary paradigm for designing fuzzy rule-based systems from examples. En *Proc. Second IEE/IEEE International Conference on Genetic Algorithms and Engineering Systems: Innovations and Applications (GALESIA '97)*, páginas 139–144. Glasgow.
- [CFM96] Carse B., Fogarty T. C., y Munro A. (1996) Evolving fuzzy rule based controllers using genetic algorithms. *Fuzzy Sets and Systems* 80: 273–293.
- [CH95] Cordón O. y Herrera F. (1995) A general study on genetic fuzzy systems. En Periaux J., Winter G., Galán M., y Cuesta P. (Eds.) *Genetic Algorithms in Engineering and Computer Science*, páginas 33–57. John Wiley and Sons.
- [CH96a] Cordón O. y Herrera F. (Julio 1996) Generating and selecting fuzzy control rules using evolution strategies and genetic algorithms. En *Proc. Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'96)*, páginas 733–738. Granada.
- [CH96b] Cordón O. y Herrera F. (1996) A hybrid genetic algorithm-evolution strategy process for learning fuzzy logic controller knowledge bases. En Herrera F. y Verdegay J. (Eds.) *Fuzzy Logic and Soft Computing*, páginas 251–278. Physica-Verlag.
- [CH96c] Cordón O. y Herrera F. (Diciembre 1996) Hybridizing genetic algorithms with sharing scheme and evolution strategies for designing approximative fuzzy logic controllers. Technical Report DECSAI-96126, Universidad de Granada, Departamento de Ciencias de la Computación e Inteligencia Artificial. E.T.S.I. Informática. Granada. España.
- [CH97a] Cordón O. y Herrera F. (1997) Evolutionary design of TSK fuzzy rule-based systems using (μ, λ) -evolution strategies. En *Proc. Sixth IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'97)*, volumen 1, páginas 509–514. Barcelona.
- [CH97b] Cordón O. y Herrera F. (1997) Identification of linguistic fuzzy models by means of genetic algorithms. En Driankov D. y Hellendoorn H. (Eds.) *Fuzzy Model Identification. Selected Approaches*, páginas 215–250. Springer-Verlag.
- [CH97c] Cordón O. y Herrera F. (Diciembre 1997) A three-stage evolutionary process for learning descriptive and approximative fuzzy logic controller knowledge bases from examples. Por aparecer en *International Journal of Approximate Reasoning*. Vol. 17, num. 4.

- [CH97d] Cordon O. y Herrera F. (Mayo 1997) A two-stage evolutionary process for designing TSK fuzzy rule-based systems. Technical Report DECSAI-97115, Universidad de Granada, Departamento de Ciencias de la Computación e Inteligencia Artificial. E.T.S.I. Informática. Granada. España.
- [CHHVL96] Cordon O., Herrera F., Herrera-Viedma E., y Lozano M. (Enero 1996) Genetic algorithms and fuzzy logic in control processes. *Archives of Control Sciences* 5(1): 135–168.
- [CHI91] Chang T. C., Hasegawa K., y Ibbs C. W. (1991) The effects of membership functions in fuzzy reasoning. *Fuzzy Sets and Systems* 44: 169–186.
- [CHL96a] Cordon O., Herrera F., y Lozano M. (Diciembre 1996) A classified review on the combination fuzzy logic-genetic algorithms bibliography. Technical Report DECSAI-95129, Universidad de Granada, Departamento de Ciencias de la Computación e Inteligencia Artificial. E.T.S.I. Informática. Granada. España.
- [CHL96b] Cordon O., Herrera F., y Lozano M. (Marzo 1996) On the bidirectional integration of genetic algorithms and fuzzy logic. En *Proc. Second Online Workshop on Evolutionary Computation (WEC2)*, páginas 13–16. Nagoya.
- [CHL96c] Cordon O., Herrera F., y Lozano M. (1996) A three-stage method for designing genetic fuzzy systems by learning from examples. En Voight H., Ebeling W., Rechemberg E., y Schwefel H. (Eds.) *Proc. Fourth International Conference on Parallel Problem Solving from Nature - PPSN IV. L.N.C.S. 1141*, páginas 720–729. Springer-Verlag, Berlin.
- [CHL97a] Cordon O., Herrera F., y Lozano M. (1997) A classified review on the combination fuzzy logic-genetic algorithms bibliography: 1989-1995. En Sanchez E., Shibata T., y Zadeh L. (Eds.) *Genetic Algorithms and Fuzzy Logic Systems. Soft Computing Perspectives*, páginas 209–241. World Scientific.
- [CHL97b] Cordon O., Herrera F., y Lozano M. (1997) On the combination of fuzzy logic and evolutionary computation: a short review and bibliography. En Pedrycz W. (Ed.) *Fuzzy Evolutionary Computation*, páginas 57–77. Kluwer Academic.
- [CHP97] Cordon O., Herrera F., y Peregrín A. (Febrero 1997) Applicability of the fuzzy operators in the design of fuzzy logic controllers. *Fuzzy Sets and Systems* 86: 15–41.
- [CHS97] Cordon O., Herrera F., y Sánchez L. (1997) Evolutionary learning processes for data analysis in electrical engineering applications. En Quagliarelli D., Periaux J., Poloni C., y Winter G. (Eds.) *Genetic Algorithms in Engineering and Computer Science*. John Wiley and Sons.

- [CK89] Cao Z. y Kandel A. (1989) Applicability of some fuzzy implication operators. *Fuzzy Sets and Systems* 31: 151–186.
- [Cox95] Cox E. (1995) A fuzzy system for detecting anomalous behaviors in healthcare provider claims. En Goonatilake S. y Treleaven P. (Eds.) *Intelligent Systems for Finance and Bussines*, páginas 111–134. John Willey & Sons.
- [CV93] Cooper M. G. y Vidal J. J. (1993) Genetic design of fuzzy logic controllers. En *Proc. Second International Conference on Fuzzy Theory and Technology (FTT'93)*. Durham.
- [Dav91] Davidor Y. (1991) *Genetic Algorithms and Robotics: A Heuristic Strategy for Optimization*. World Scientific.
- [DG89] Deb K. y Goldberg D. E. (1989) An investigation of niche and species formation in genetic function optimization. En *Proc. Third International Conference on Genetic Algorithms (ICGA'89)*, páginas 42–50. Hillsdale.
- [DHR93] Driankov D., Hellendoorn H., y Reinfrank M. (1993) *An Introduction to Fuzzy Control*. Springer-Verlag.
- [DSG93] DeJong K. A., Spears W. M., y Gordon D. F. (1993) Using genetic algorithms for concept learning. *Machine Learning* 13: 161–188.
- [DVV97] Delgado M., Vila M. A., y Voxman W. (1997) On a canonical representation of fuzzy numbers. Por aparecer en *Fuzzy Sets and Systems*.
- [Fog91] Fogel D. B. (1991) *System Identification trough Simulated Evolution. A Machine Learning Approach*. Ginn Press.
- [Fog95] Fogel D. B. (1995) *Evolutionary Computation. Toward a New Philosophy of Machine Intelligence*. IEEE Press.
- [For93] Forrest S. (Ed.) (1993) *Proceedings of the Fifth International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers.
- [FOW66] Fogel L. J., Owens A. J., y Walsh M. J. (1966) *Artificial Intelligence through Simulated Evolution*. John Willey & Sons.
- [GGP97] Gudwin R. R., Gomide F. A., y Pedrycz W. (Junio 1997) Nonlinear context adaption with a genetic algorithm. En *Proc. Seventh International Fuzzy Systems Association World Congress (IFSA'97)*, volumen 2, páginas 300–305. Praga.
- [GH97] González A. y Herrera F. (1997) Multi-stage genetic fuzzy systems based on the iterative rule learning approach. Por aparecer en *Mathware & Soft Computing*.

- [Gol89] Goldberg D. E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley.
- [GP96] González A. y Pérez R. (1996) A learning system of fuzzy control rules based on genetic algorithms. En Herrera F. y Verdegay J. (Eds.) *Fuzzy Logic and Soft Computing*, páginas 202–225. Physica-Verlag.
- [GP97] González A. y Pérez R. (1997) Completeness and consistency conditions for learning fuzzy rules. Por aparecer en *Fuzzy Sets and Systems*.
- [GQ91] Gupta M. M. y Qi J. (1991) Design of fuzzy logic controllers based on generalized T-operators. *Fuzzy Sets and Systems* 40: 473–489.
- [Gre94] Grefenstette J. J. (Ed.) (1994) *Genetic Algorithms for Machine Learning*. Kluwer Academic.
- [GT95] Goonatilake S. y Treleaven P. (Eds.) (1995) *Intelligent Systems for Finance and Bussines*. John Willey & Sons.
- [Hir93] Hirota K. (Ed.) (1993) *Industrial Applications of Fuzzy Technology*. Springer-Verlag.
- [HLV95] Herrera F., Lozano M., y Verdegay J. L. (1995) Tuning fuzzy controllers by genetic algorithms. *International Journal of Approximate Reasoning* 12: 299–315.
- [HLV97a] Herrera F., Lozano M., y Verdegay J. L. (1997) Fuzzy connectives based crossover operators to model genetic algoirhtms population diversity. Por aparecer en *Fuzzy Sets and Systems*.
- [HLV97b] Herrera F., Lozano M., y Verdegay J. L. (1997) A learning process for fuzzy control rules using genetic algorithms. Por aparecer en *Fuzzy Sets and Systems*.
- [HLV97c] Herrera F., Lozano M., y Verdegay J. L. (1997) Tackling real-coded genetic algorithms: operators and tools for behavioural analysis. Por aparecer en *Artificial Intelligence Review*.
- [HM97] Herrera F. y Magdalena L. (1997) Genetic fuzzy systems. En Mesiar R. y Riecan B. (Eds.) *Tatra Mountains Mathematical Publications. Fuzzy Structures. Current Trends. Lecture Notes of the Tutorial: Genetic Fuzzy Systems. Seventh IFSA World Congress (IFSA '97)*, volumen 13, páginas 93–121.
- [HMB93] Harris C. J., Moore C. G., y Brown M. (1993) *Intelligent Control. Aspects of Fuzzy Logic and Neural Networks*. World Scientific.
- [Hol75] Holland J. H. (1975) *Adaptation in Natural and Artificial Systems*. Ann arbor: The University of Michigan Press.

- [Hol86] Holland J. H. (1986) Escaping britleness: the possibilities of general purpose learning algorithms applied to parallel rule-based systems. En Michalski R., Carbonell J., y Michel T. (Eds.) *Machine Learning: An AI Approach. Vol II*, páginas 593–623. Morgan-Kaufmann.
- [HP95] Hoffmann F. y Pfister G. (Julio 1995) A new learning method for the design of hierarchical fuzzy controllers using messy genetic algorithms. En *Proc. Sixth International Fuzzy Systems Association World Congress (IFSA '95)*, páginas 249–252. Sao Paulo.
- [HR78] Holland J. H. y Reitman J. S. (1978) Cognitive systems based on adaptive algorithms. En Waterman D. A. y Hayes-Roth F. (Eds.) *Pattern-Directed Inference Systems*. Academic Press.
- [HT93] Hellendoorn H. y Thomas C. (1993) Defuzzification in fuzzy controllers. *Journal of Intelligent Fuzzy Systems* 1: 109–123.
- [HT94] Hwang W. R. y Thompson W. E. (Junio 1994) Design of fuzzy logic controllers using genetic algorithms. En *Proc. Third IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'94)*, páginas 1383–1388. Orlando.
- [HV96] Herrera F. y Verdegay J. L. (Eds.) (1996) *Genetic Algorithms and Soft Computing*. Physica-Verlag.
- [INT+94] Ishibuchi H., Nozaki K., Tanaka H., Hosaka Y., y Matsuda M. (1994) Empirical study on learning in fuzzy systems by rice test analysis. *Fuzzy Sets and Systems* 64: 129–144.
- [Jan93a] Jang J.-S. R. (Mayo 1993) ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics* 23(3): 665–684.
- [Jan93b] Janikov C. Z. (1993) A knowledge-intensive genetic algorithm for supervised learning. *Machine Learning* 13: 189–228.
- [JS93] Jang J.-S. R. y Sun C. T. (Mayo 1993) Functional equivalence between radial basis function networks and fuzzy inference systems. *IEEE Transactions on Neural Networks* 4(1): 156–159.
- [Kar91a] Karr C. (Marzo 1991) Applying genetics to fuzzy logic. *AI Expert* páginas 38–43.
- [Kar91b] Karr C. (Febrero 1991) Genetic algorithms for fuzzy controllers. *AI Expert* páginas 26–33.

- [KB93] Kropp K. y Baitinger U. G. (Septiembre 1993) Optimization of fuzzy logic controller inference rules using a genetic algorithm. En *Proc. First European Congress on Fuzzy and Intelligent Technologies (EUFIT'93)*, páginas 1090–1096. Aachen.
- [KKK94] Kinzel J., Klawonn F., y Kruse R. (Junio 1994) Modifications of genetic algorithms for designing and optimizing fuzzy controllers. En *Proc. First IEEE Conference on Evolutionary Computation (EC-IEEE'94)*, páginas 28–33. Orlando.
- [KKS85] Kiszka J., Kochanska M., y Sliwinska D. (1985) The influence of some fuzzy implication operators on the accuracy of a fuzzy model - parts I and II. *Fuzzy Sets and Systems* 15: 111–128, 223–240.
- [Koc96] Koczy L. (Septiembre 1996) Fuzzy if ... then rule models and their transformation into one another. *IEEE Transactions on Systems, Man, and Cybernetics* 26(5): 621–637.
- [Kos92] Kosko B. (1992) *Neural Networks and Fuzzy Systems*. Prentice-Hall.
- [Koz92] Koza J. (1992) *Genetic Programming. On the Programming of Computers by Means of Natural Selection*. The MIT Press.
- [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 and Sons.
- [KY95] Klir G. J. y Yuan B. (1995) *Fuzzy Sets and Fuzzy Logic*. Prentice-Hall.
- [LAT96] Lofti A., Andersen H. C., y Tsoi A. C. (1996) Interpretation preservation of adaptive fuzzy inference systems. *International Journal of Approximate Reasoning* 15: 379–394.
- [Lee90] Lee C. C. (Marzo 1990) Fuzzy logic in control systems: fuzzy logic controller – parts I and II. *IEEE Transactions on Systems, Man, and Cybernetics* 20(2): 404–418, 419–435.
- [Lip87] Lippman R. P. (1987) An introduction to computing with neural nets. *IEEE ASSP Magazine* 4: 4–22.
- [LP94] Leitch D. y Probert P. (1994) Context depending coding in genetic algorithms for the design of fuzzy systems. En *Proc. IEEE/Nagoya University WWW on Fuzzy Logic and Neural Networks/Genetic Algorithms*. Nagoya.
- [LS95] Lee M. A. y Saloman R. (Julio 1995) Hybrid evolutionary algorithms for fuzzy system design. En *Proc. Sixth International Fuzzy Systems Association World Congress (IFSA'95)*, páginas 269–272. Sao Paulo.

- [LT93a] Lee M. A. y Takagi H. (Julio 1993) Embedding apriori knowledge into an integrated fuzzy system design method based on genetic algorithms. En *Proc. Fifth International Fuzzy Systems Association World Congress (IFSA'93)*, páginas 1293–1296. Seoul.
- [LT93b] Lee M. A. y Takagi H. (Marzo 1993) Integrating design stages of fuzzy systems using genetic algorithms. En *Proc. Second IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'93)*, páginas 613–617. San Francisco.
- [LT96] Lee M. A. y Takagi H. (1996) Hybrid genetic-fuzzy systems for intelligent systems design. En Herrera F. y Verdegay J. (Eds.) *Fuzzy Logic and Soft Computing*, páginas 226–250. Physica-Verlag.
- [MA75] Mamdani E. H. y Assilian S. (1975) An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies* 7: 1–13.
- [Mag96] Magdalena L. (Julio 1996) Adapting gain and sensibility of FLCs with genetic algorithms. En *Proc. Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'96)*, páginas 739–744. Granada.
- [Mam74] Mamdani E. H. (1974) Applications of fuzzy algorithm for control a simple dynamic plant. *Proceedings of the IEE* 121(12): 1585–1588.
- [Men95] Mendel J. M. (Marzo 1995) Fuzzy logic systems for engineering: a tutorial. *Proceedings of the IEEE* 83(3): 345–377.
- [Mic96] Michalewicz Z. (1996) *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag.
- [MV96] Magdalena L. y Velasco J. R. (1996) Fuzzy rule-based controllers that learn by evolving their knowledge bases. En Herrera F. y Verdegay J. (Eds.) *Fuzzy Logic and Soft Computing*, páginas 172–201. Physica-Verlag.
- [MV97] Magdalena L. y Velasco J. R. (1997) Evolutionary based learning of fuzzy controllers. En Pedrycz W. (Ed.) *Fuzzy Evolutionary Computation*, páginas 249–268. Kluwer Academic.
- [NHW91] Nomura H., Hayashi L., y Wakami N. (Julio 1991) A self-tuning method of fuzzy control by descendent method. En *Proc. Fourth International Fuzzy Systems Association World Congress (IFSA'91)*, páginas 155–158. Bruselas.
- [NHW92] Nomura H., Hayashi L., y Wakami N. (Julio 1992) A learning method of fuzzy inference rules by descent method. En *Proc. First IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'92)*, páginas 203–210. San Diego.

- [Nis93] Nissen V. (1993) Evolutionary algorithms in management science: an overview and list of references. Technical Report # 9303, Papers on Economics and Evolution. Universitaet Goettingen.
- [NIT97] Nozaki K., Ishibuchi H., y Tanaka H. (Febrero 1997) A simple but powerful heuristic method for generating fuzzy rules from numerical data. *Fuzzy Sets and Systems* 86: 251–270.
- [NL94] Ng K. C. y Lee Y. (Junio 1994) Design of sophisticated fuzzy logic controllers using genetic algorithms. En *Proc. Third IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'94)*, páginas 1708–1712. Orlando.
- [PB93] Parodi A. y Bonelli P. (1993) A new approach to fuzzy classifier system. En *Proc. Fifth International Conference on Genetic Algorithms (ICGA'93)*, páginas 223–230. San Mateo.
- [Ped89] Pedrycz W. (1989) *Fuzzy Control and Fuzzy Systems*. John Willey & Sons.
- [Ped96] Pedrycz W. (Ed.) (1996) *Fuzzy Modelling: Paradigms and Practice*. Kluwer Academic.
- [Ped97] Pedrycz W. (Ed.) (1997) *Fuzzy Evolutionary Computation*. Kluwer Academic.
- [San97a] Sanchez L. (Febrero 1997) Estimación de la longitud de cable en un grupo de poblaciones a partir de un modelo de regresión. Technical report, Universidad de Oviedo, Escuela Técnica Superior de Ingenieros Industriales de Gijón. Oviedo. España.
- [San97b] Sanchez L. (Junio 1997) Interval-valued GA-P algorithms. Technical report, Universidad de Oviedo, Escuela Técnica Superior de Ingenieros Industriales de Gijón. Oviedo. España.
- [San97c] Sanchez L. (Junio 1997) Modelos de zona rural y urbana. Technical report, Universidad de Oviedo, Escuela Técnica Superior de Ingenieros Industriales de Gijón. Universidad de Oviedo. Oviedo. España.
- [Sch95] Schwefel H. P. (1995) *Evolution and Optimum Seeking. Sixth-Generation Computer Technology Series*. John Wiley and Sons.
- [SK94] Satyadas A. y Krishnakumar K. (Diciembre 1994) Evolutionary learning techniques for fuzzy controller synthesis. En *Proc. First Industry/University Symposium on High Speed Civil Transport Vehicles*. North Carolina.
- [SKG93] Surmann H., Kanstein A., y Goser K. (Septiembre 1993) Self-organizing and genetic algorithms for an automatic design of fuzzy control and decision systems. En *Proc. First European Congress on Fuzzy and Intelligent Technologies (EUFIT'93)*, páginas 1097–1104. Aachen.

- [Smi80] Smith S. F. (1980) *A Learning System Based on Genetic Adaptive Algorithms*. PhD thesis, Universidad de Pittsburgh.
- [SSZ97] Sanchez E., Shibata T., y Zadeh L. (Eds.) (1997) *Genetic Algorithms and Fuzzy Logic Systems. Soft Computing Perspectives*. World Scientific.
- [SY93] Sugeno M. y Yasukawa T. (Febrero 1993) A fuzzy-logic-based approach to qualitative modeling. *IEEE Transactions on Fuzzy Systems* 1(1): 7–31.
- [Tan95] Tano S. (1995) Fuzzy logic for financial trading. En Goonatilake S. y Treleaven P. (Eds.) *Intelligent Systems for Finance and Business*, páginas 209–224. John Willey & Sons.
- [Tet95] Tettamanzi A. G. (1995) Evolutionary algorithms and fuzzy logic: a two-way integration. En *Proc. Second Joint Conference on Information Sciences*, páginas 464–467.
- [TH91] Takagi H. y Hayashi I. (1991) NN-driven fuzzy reasoning. *International Journal of Approximate Reasoning* 5(3): 191–212.
- [Thr91] Thrift P. (1991) Fuzzy logic synthesis with genetic algorithms. En *Proc. Fourth International Conference on Genetic Algorithms (ICGA'91)*, páginas 509–513. San Diego.
- [TS85] Takagi T. y Sugeno M. (Febrero 1985) Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics* 15(1): 116–132.
- [TSKK92] Takagi H., Suzuki N., Koda T., y Kojima Y. (1992) Neural networks designed on approximate reasoning architecture and their applications. *IEEE Transactions on Neural Networks* 3(5): 752–760.
- [TV85] Trillas E. y Valverde L. (1985) On implication and indistinguishability in the setting of fuzzy logic. En Kacprzyk J. y Yager R. (Eds.) *Management Decision Support Systems Using Fuzzy Logic and Possibility Theory*, páginas 198–212. Verlag TUV Rheinland.
- [UK80] Umbers I. G. y King P. J. (1980) An analysis of human-decision making in cement kiln control and the implications for automation. *International Journal of Man-Machine Studies* 12: 11–23.
- [Ven92] Venturini G. (1992) SIA: a supervised inductive algorithm with genetic search for learning attribute based concepts. En *Proc. European Conference on Machine Learning*, páginas 280–296. Viena.

- [VM95a] Velasco J. R. y Magdalena L. (1995) Genetic algorithms in fuzzy control systems. En Periaux J., Winter G., Galán M., y Cuesta P. (Eds.) *Genetic Algorithms in Engineering and Computer Science*, páginas 141–165. John Wiley and Sons.
- [VM95b] Velasco J. R. y Magdalena L. (Julio 1995) Genetic learning applied to fuzzy rules and fuzzy knowledge bases. En *Proc. Sixth International Fuzzy Systems Association World Congress (IFSA '95)*, páginas 257–264. Sao Paulo.
- [Wan92] Wang L. X. (Julio 1992) Fuzzy systems are universal approximators. En *Proc. First IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'92)*, páginas 1163–1170. San Diego.
- [Wan94] Wang L. X. (1994) *Adaptive Fuzzy Systems and Control*. Prentice-Hall.
- [Was93] Wasserman P. D. (1993) *Advanced Methods in Neural Computing*. Van Nostrand Reinhold.
- [Wie94] Wienholt W. (Junio 1994) Improving a fuzzy inference system by means of evolution strategy. En *Proc. Fourth Fuzzy Days*, páginas 163–172. Dortmund.
- [WM92] Wang L. X. y Mendel J. M. (Noviembre 1992) Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man, and Cybernetics* 22(6): 1414–1427.
- [WS92] Whitley L. D. y Schaffer J. D. (Eds.) (1992) *Proceedings of the International Workshop on Combinations of Genetic Algorithms and Neural Networks*. IEEE Computer Society Press.
- [YG94] Ye Z. y Gu L. (1994) A fuzzy system for trading the shanghai stock market. En Deboeck G. (Ed.) *Trading on the Edge: Neural, Genetic, and Fuzzy Systems for Chaotic and Financial Markets*, páginas 207–214. John Willey & Sons.
- [YG95] Yen J. y Gillespie W. (Julio 1995) Integrating global and local evaluations for fuzzy model identification using genetic algorithms. En *Proc. Sixth International Fuzzy Systems Association World Congress (IFSA '95)*, páginas 121–124. Sao Paulo.
- [YI96] Yagiura M. y Ibaraki T. (1996) Genetic and local search algorithms as robust and simple optimization tools. En Osman I. y Kelly J. (Eds.) *Meta-heuristics: Theory and Applications*, páginas 63–82. Kluwer Academic.
- [YPH93] Yoshinari Y., Pedrycz W., y Hirota K. (1993) Construction of fuzzy models through clustering techniques. *Fuzzy Sets and Systems* 54: 157–165.

-
- [YYY+91] Yuize H., Yagyu T., Yoneda M., Katoh Y., Tano S., Grabisch M., y Fukami S. (Noviembre 1991) Decision support system for foreign exchange trading - practical implementation. En *Proc. International Fuzzy Engineering Symposium (IFES'91)*, páginas 971–982. Yokohama.
- [Zad65] Zadeh L. A. (Febrero 1965) Fuzzy sets. *Information and Control* 8: 338–353.
- [Zad73] Zadeh L. A. (1973) Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics* 3: 28–44.
- [Zim96] Zimmermann H. J. (1996) *Fuzzy Sets Theory and its Applications*. Kluwer Academic.