

SISTEMAS DIFUSOS EVOLUTIVOS

Francisco Herrera

Dpto. de Ciencias de la Computación e Inteligencia Artificial
 E.T.S. de Ingeniería Informática
 Universidad de Granada
 herrera@decsai.ugr.es

Resumen

Los sistemas difusos han demostrado su utilidad para resolver problemas en diferentes ámbitos de aplicación. El uso de los Algoritmos Genéticos en el diseño de sistemas difusos permite introducir la capacidad de aprendizaje y adaptación.

En esta presentación introduciremos los sistemas difusos evolutivos prestando atención a diferentes aspectos de los algoritmos genéticos que merecen ser destacados por su potencial aplicación en el diseño de sistemas difusos evolutivos. Posteriormente nos centraremos en los sistemas difusos evolutivos, en los modelos clásicos y en las nuevas propuestas que actualmente están siendo desarrolladas, junto con algunos problemas abiertos.

1 INTRODUCCIÓN

Los sistemas difusos se han aplicado con éxito a problemas descubrimiento de información, clasificación, modelado, control, etc., y en un considerable número de aplicaciones.

La Tabla 1 presenta el número de artículos disponibles en las revistas de Elsevier desde Enero de 2003 a Enero de 2004, realizando la búsqueda por las palabras clave “fuzzy rules/fuzzy systems”, para las revistas clasificadas en un área científica concreta (energía, medicina, ...). Ésta nos muestra diferentes campos de aplicación donde podemos encontrar actualmente aplicaciones.

Un sistema difuso evolutivo es un sistema difuso al que se incorpora la capacidad de aprendizaje y adaptación mediante algoritmos genéticos (AGs) [8]. Los AGs [12,14,20] son algoritmos de búsqueda, basados en la evolución natural y la genética, que proporcionan un mecanismo de búsqueda y optimización robusto en sistemas complejos, y ofrecen una vía útil para problemas de optimización que requieran eficiencia y eficacia.

El análisis de la literatura especializada muestra que los sistemas difusos más estudiados y aplicados son los

sistemas basados en reglas difusas (SBRDs) y asociados a éstos están los llamados sistemas basados en reglas difusas evolutivos (SBRDEs), que serán los modelos en los que centremos nuestra atención. La Figura 1 muestra un esquema gráfico de estos modelos.

Tabla 1. Artículos Disponibles desde Enero de 2003 a Enero de 2004 en Revistas de 10 áreas Temáticas de Elsevier

Agricultura y Ciencias Biológicas	39	Ciencias Ambientales	45
Ciencias de los Materiales	29	Inmunología / Microbiología	3
Ingeniería Química	26	Biología	18
Química	16	Medicina	38
Energía	91	Física y Astronomía	44
Tierra y Ciencias Planetarias	32	Ingeniería (fuzzy)	818
		fuzzy rules: 93	
		fuzzy control: 199	
		fuzzy systems: 373	



Figura 1. Sistemas Basados en Reglas Difusas Evolutivos

En esta presentación vamos a mostrar el estado del arte del desarrollo de los SBRDEs. En la Sección 2 introduciremos brevemente los AGs, prestando especial

atención a modelos específicos que merecen ser destacados por su potencial aplicación en el desarrollo de los SBRDEs. En la Sección 3 nos centraremos en los modelos clásicos de SBRDEs y en las propuestas que actualmente están siendo desarrolladas. También hablaremos de algunos problemas abiertos en el estudio de los SBRDEs. Unos breves comentarios finales se presentarán en la Sección 4.

2 ALGORITMOS GENÉTICOS

Los AGs [12,14,21] son procedimientos adaptativos para la búsqueda de soluciones en espacios complejos, inspirados en los procesos genéticos de los organismos naturales y en los principios de la evolución natural de poblaciones.

La idea básica de los mismos es mantener una población de cromosomas, los cuales representan soluciones candidatas a un problema concreto, que evolucionan con el tiempo a través de un proceso de competición y variación controlada. Cada cromosoma tiene una bondad o adaptación asociada, que describe la adecuación de la solución a la que representa. El proceso de competición, denominado mecanismo de selección, utiliza estas adaptaciones para determinar los cromosomas que se usan para crear otros nuevos. Los nuevos cromosomas se generan a través de los operadores genéticos denominados cruce y mutación.

Los AGs se han aplicado con éxito en problemas de búsqueda y optimización. Gran parte de este éxito se debe a su capacidad para explotar la información acumulada sobre un espacio de búsqueda, y así dirigir las siguientes búsquedas hacia los mejores subespacios. Esta es su principal ventaja, sobre todo en espacios grandes, complejos y parcialmente definidos donde las técnicas clásicas de búsqueda no son apropiadas. Los AGs ofrecen una aproximación válida a problemas que requieren técnicas de búsqueda eficaces y eficientes.

Bajo la formulación inicial de los AGs, las soluciones se han representado mediante cadenas binarias de longitud fija. Sin embargo, las buenas propiedades de los AGs no radican en el uso de cadenas de ceros y unos. Por esta razón, frecuentemente se han utilizado representaciones no binarias más adecuadas para cada uno de los problemas particulares de aplicación. Una de las representaciones alternativas más importantes es la representación real. Esta representación parece natural cuando se tratan problemas con variables continuas. El uso de nuevas representaciones y la construcción de nuevos operadores para manipularlas ha propiciado que la concepción actual de AG sea bastante diferente y más general que la idea original.

La estructura básica de un AG se muestra en la Figura 2.

Para la descripción básica de un AG hay que tener en cuenta los siguientes componentes:

- ❖ la representación de las soluciones
- ❖ el mecanismo de selección de soluciones para su reproducción
- ❖ los operadores genéticos de cruce y mutación
- ❖ los parámetros de los AGs (tamaño de la población, probabilidades de cruce y mutación, ...)

Procedimiento Algoritmo Genético

Inicio

$t=0$

Inicializar $P(t)$

Evaluar $P(t)$

Mientras (No se cumpla la condición de parada) **Hacer**

Inicio

$t=t+1$

Seleccionar $P(t)$ desde $P(t-1)$

Aplicar Cruce y Mutación sobre $P(t)$

Evaluar $P(t)$

Final

Final

Figura 2. Estructura de un AG Generacional

El esquema anterior es el conocido AG Generacional que reemplaza un conjunto de soluciones en cada iteración. Existe otro modelo, el AG estacionario (*steady state*) que en cada iteración solo reemplaza a uno o dos individuos. Este es un modelo que provoca mayor convergencia en el proceso de evolución cuando el elemento/s de la población sustituido/s es el peor de la misma. El algoritmo básico generando único descendiente es el siguiente:

- ❖ Seleccionar dos padres de la población.
- ❖ Crear un hijo utilizando cruce y mutación.
- ❖ Evaluar el hijo.
- ❖ Seleccionar un individuo en la población, que puede ser reemplazado por el hijo.
- ❖ Decidir si este individuo será reemplazado.

La descripción anterior de AG generacional y estacionario recoge las ideas básicas para la implementación de un AG. Pero desde su propuesta original, se han desarrollado una variedad creciente de estudios y modelos que permiten el uso de AGs en problemas donde se desea

utilizar un modelo eficiente, se requiere la obtención de múltiples soluciones a un mismo problema o la evaluación de múltiples objetivos, etc. Algunos de estos modelos están empezando a ser utilizados en el ámbito de los SBRDEs, y su uso puede mejorar el comportamiento de los mismos. Algunas de estas ideas y modelos se muestran a continuación.

- Modelos que introducen un buen equilibrio entre diversidad y convergencia en la búsqueda en espacios complejos. Existen múltiples estudios y propuestas sobre técnicas de diversidad que están permitiendo avanzar en el diseño de AGs con buen equilibrio entre diversidad/convergencia junto con buena eficiencia en diferentes problemas. Un modelo ya clásico es el AG CHC [11] que está siendo muy utilizado en diferentes ámbitos de aplicación, y que es un referente en este tipo de estudios. En el ámbito en el que nos encontramos, podemos destacar los AG difusos adaptativos que utilizan SBRDs para guiar el proceso de búsqueda adaptando diferentes componentes/parámetros de los AGs; en [16] se puede encontrar una breve revisión de éstos.
- Modelos que permiten obtener múltiples soluciones de un problema en una ejecución. Para ello se han desarrollado los llamados AG con nichos (*niching GA*), que permiten evolucionar diferentes soluciones a lo largo de una ejecución, cada una de ellas perteneciente a un nicho (subregión del espacio de soluciones). Estas soluciones pueden corresponden a diferentes soluciones de igual calidad o a diferentes óptimos locales del problema, como se muestra en la Figura 2. En [22] se puede encontrar un estudio comparativo reciente de diferentes modelos.

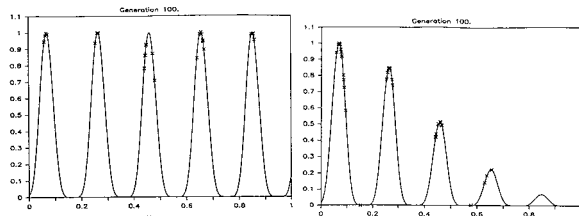


Figura 3. Ejemplos de problemas con múltiples soluciones de igual calidad y múltiples óptimos locales

- Modelos que permiten manejar múltiples objetivos, los algoritmos genéticos multiobjetivo [7,10]. El uso de estos algoritmos conduce a obtener un conjunto de soluciones no dominadas pertenecientes a la frontera del Pareto (ver Figura 4). Su uso en el diseño de SBRDs permite introducir objetivos adicionales a la precisión tales como tamaño, medidas de interpretabilidad, etc.
- En términos de eficiencia y de manejo de problemas complejos podemos hablar de los AG paralelos/distribuidos que en los últimos años se han

desarrollado ampliamente [1]. Estos algoritmos permiten reducir el tiempo de ejecución cuando se ejecutan sobre máquinas distribuidas, y permiten obtener mejoras de comportamiento por la separación espacial de soluciones que evita convergencia prematura en muchos problemas.

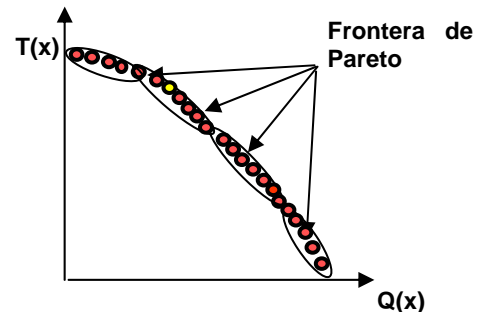


Figura 4. Frontera de Pareto para dos funciones objetivo

- Abordar la solución de un problema como la resolución cooperativa de varios subproblemas es una metodología clásica en el diseño de algoritmos. En la actualidad se están desarrollando ampliamente los llamados AGs coevolutivos que permiten evolucionar en paralelo diferentes partes de un problema (soluciones parciales de subproblemas llamadas especies) con una evaluación global de las mismas. Estos algoritmos se están aplicando en problemas que pueden descomponerse conjuntamente con la solución, y pueden ser una buena alternativa para la evolución de diferentes componentes de los SBRDs. La Figura 5 muestra un modelo de evolución de especies (subproblemas).

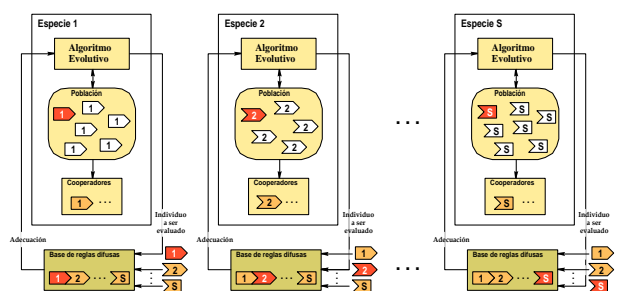


Figura 5. Algoritmo Genético Coevolutivo Cooperativo

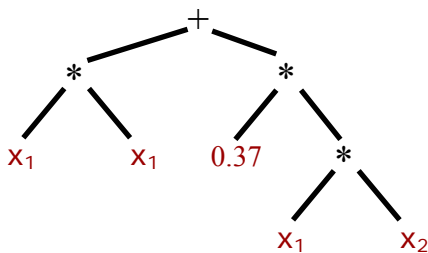
- Podemos definir la Programación Genética (PG) como un caso particular de AGs usados para inducir programas de ordenador de un modo automático, representados mediante un árbol sintáctico de expresiones o como secuencias de instrucciones de máquinas de estados finitos. La representación de programas en forma de árboles de expresión se basa en la existencia de una gramática libre de contexto que define las sentencias válidas del lenguaje.

Naturalmente, los operadores genéticos han de adaptarse a la representación empleada. Las Figuras 6 y 7 muestran un ejemplo con una gramática, un árbol sintáctico, el cromosoma y su expresión numérica asociada.

$$S \rightarrow x_1 \mid x_2 \mid N \mid + \mid S S \mid * \mid S S$$

$$N \rightarrow \Re$$

Figura 6. Gramática con dos variables y 2 operadores



Cromosoma: + * x₁ x₁ * 0.37 * x₁ x₂

Expresión: $x_1^2 + 0.37 * x_1 * x_2$

Figura 7. Árbol sintáctico, cromosoma y expresión

3 SISTEMAS DIFUSOS EVOLUTIVOS

Existe un gran número de trabajos dedicados a la generación automática de Bases de Conocimiento (BC) de un SBRDs con AGs. En [8] se puede encontrar un extenso estado del arte, y en [9] se describen aproximaciones recientes.

El aspecto central para el empleo de un AG para automatizar el diseño de una BC es que el proceso de diseño se pueda analizar como un problema de optimización o búsqueda.

Desde el punto de vista de optimización, el hecho de encontrar una BC apropiada para un problema es equivalente a parametrizar la BC, y así poder encontrar aquellos valores de los parámetros que sean óptimos con respecto al criterio de diseño. Los parámetros de la BC constituyen el espacio de optimización, el cual se transforma en una adecuada representación genética sobre la que opera el proceso de búsqueda.

El primer paso en el diseño de un SBRDE es decidir que parte de la BC será aprendida o ajustada. Básicamente tenemos dos componentes:

- la base de datos (BD) que contiene las funciones de escala de las variables y las funciones de pertenencia asociadas a las etiquetas lingüísticas, y
- una base de reglas (BR) que contiene la colección de reglas difusas.

Por otra parte, hemos de distinguir entre ajuste/adaptación y aprendizaje. El ajuste está asociado con la optimización de un SBRDs que ya existe, mientras que el aprendizaje consiste en un método de diseño automático para extraer reglas.

El proceso de ajuste asume la existencia de una BR predefinida y tiene como objetivo encontrar un conjunto óptimo de parámetros para las funciones de pertenencia y/o las funciones de escala, en definitiva los parámetros de la BD.

El proceso de aprendizaje actúa como un proceso de búsqueda sobre el espacio de posibles reglas (aprendiendo una BR) o sobre el espacio que representa la BC.

A continuación veremos los modelos clásicos de ajuste y aprendizaje. Posteriormente estudiaremos nuevas tendencias de SBRDEs y algunas cuestiones y problemas abiertos.

3.1. MODELOS CLÁSICOS

Los modelos de ajuste siguen un patrón común, mostrándose las diferencias en los diferentes parámetros codificados, en los tipos de función de pertenencia, o en el tipo de sistemas difusos (Mamdani, TSK, ...), entre otros aspectos.

Dentro de los modelos de aprendizaje de SBRDs podemos distinguir tres modelos básicos, el aprendizaje de una BR (Figura 8), el aprendizaje de la BC donde conjuntamente se aprenden las funciones de pertenencia y la BR (Figura 9), y el aprendizaje de la BD mediante AG y el uso de un algoritmo base para el aprendizaje de la BR (actuando este último con procedimiento asociado a la función de evaluación del AG (Figura 10).

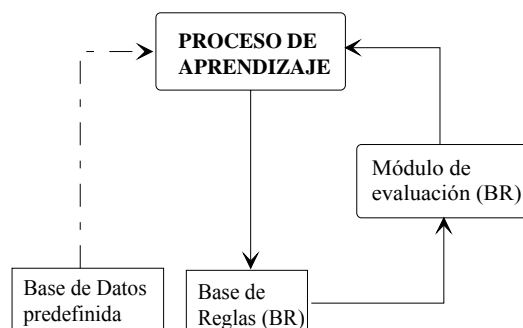


Figura 8. Esquema de aprendizaje evolutivo de la BR

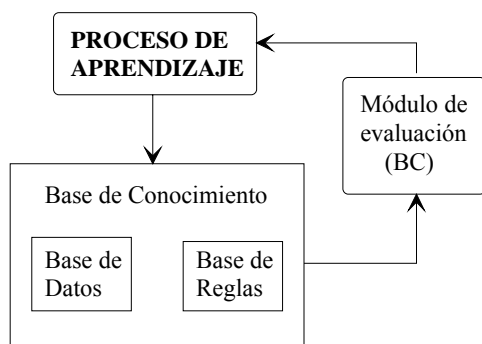


Figura 9. Esquema de aprendizaje evolutivo de la BC

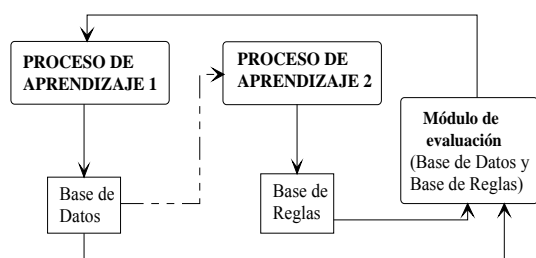


Figura 10. Esquema de aprendizaje evolutivo de la BD

En cuanto a cómo utilizar los AGs para el aprendizaje de la BR/BC, existen tres modelos generales que están fundamentados en el uso de los AGs para aprender SBRDs.

Modelo Pittsburgh. La BR/BC se codifica como un cromosoma. El AG aprende la mejor BR/BC asociada al problema.

Modelo Michigan. Utiliza una BR/BC con mecanismos de refuerzo sobre las reglas (recompensa y penalización de un peso), y sustitución de las reglas con peso bajo por nuevas reglas obtenidas utilizando AGs. Cada cromosoma codifica a una sola regla.

Modelo Iterativo. Es un modelo de aprendizaje evolutivo inductivo, a partir de ejemplos. Al igual que el modelo Michigan, los cromosomas representan reglas, pero al contrario que la anterior, se aprende sólo un individuo en cada ejecución, y se ejecuta el AG secuencialmente mientras no se verifique un criterio de cubrimiento de los ejemplos. No existe cooperación entre las reglas de la población, todos los cromosomas compiten para seleccionar el mejor que formará parte de la población. Posteriormente se aplican procesos de refinamiento para mejorar la cooperación entre las reglas mediante la selección de conjuntos de reglas cooperativas, y mediante el ajuste de los parámetros de las reglas.

El modelo Michigan fué utilizado por uno de los primeros autores que publicaron un modelo de SBRDEs (Valenzuela-Rendón [23]). Sin embargo, no ha sido

adaptado con todo su potencial real al ámbito de los SBRDs (los modelos de SBRDEs Michigan existentes son muy simples). Existe modelos recientes de aprendizaje evolutivo Michigan, tales como XCS [3,19], que son muy útiles para el modelado de sistemas dinámicos, robótica, minería de datos, etc., cuya adaptación a SBRDEs puede ser de gran utilidad en diferentes áreas como las anteriormente citadas.

3.2. NUEVAS TENDENCIAS

Los modelos clásicos que hemos comentado han sido ampliamente desarrollados. En la actualidad se están desarrollando trabajos en nuevas direcciones, explorando nuevos ámbitos de desarrollo de los SBRDs. Entre ellos vamos a destacar:

- Los modelos de selección evolutiva de reglas, encuadrados en el ámbito de la compactificación de SBRDs [18].
- El aprendizaje de la BR/BC utilizando programación genética. Se definen una gramática libre de contexto para la extracción de reglas y funciones y terminales dentro de la gramática que permiten la definición de reglas [2].
- La extracción evolutiva de características combinada con los SBRDs. Esto permite reducir el número de reglas en espacios de alta dimensionalidad, tanto por la reducción de variables dentro de una regla [15] como por la reducción global de variables [4].
- La extracción de SBRDs manteniendo un equilibrio entre interpretabilidad y precisión mediante el uso de AG multiobjetivo [17], donde diferentes objetivos representan los diferentes requerimientos del sistema que suelen ser contradictorios.
- Estudios recientes muestran como el uso de modelos de inferencia paramétricos permiten mejorar la precisión de los SBRDs [21]. Los AGs pueden ser utilizados para aprender los valores de los parámetros del sistema de inferencia o defuzzificación.
- Modelos de SBRDEs que utilizan diferentes estructuras de representación de reglas, incluyendo reglas con pesos, reglas con doble consecuente, reglas jerarquizadas, etc.

Hemos de indicar que estos nuevos modelos, junto con algunas de las diferentes propuestas asociadas a los modelos clásicos, utilizan la ventaja que proporcionan los AGs y por ende los SBRDEs frente a otras técnicas tales como redes neuronales, gradiente descendente, etc. Esta ventaja radica en la posibilidad de presentar en un cromosoma un espacio de soluciones que permite integrar variables de diferente naturaleza, binarias, discretas, reales, etc., con la definición de operadores específicos que permitan evolucionar estas estructuras. Esto permite

combinar mecanismos de selección de reglas con el ajuste de parámetros, aprender la forma de las funciones de pertenencia (gaussianas, triangulares,...) representadas con variables discretas junto con la BR y funciones de pertenencia, etc. La Figura 11 muestra un ejemplo que incluye parámetros reales asociadas a cada variable para el ajuste (2 variables, 3 etiquetas), y genes binarios asociados a cada regla (3 reglas) para su selección.

P_1	M_1	G_1	P_2	M_2	G_2	R_1	R_2	R_3
0.2	0.4	0	-0.2	-0.3	-0.5	1	0	1

Figura 11. Esquema de codificación real y binaria

3.3. ALGUNAS CUESTIONES ABIERTAS

Podemos destacar algunas cuestiones abiertas referentes a los SBRDEs en particular y que son extensibles a los SBRDS en general.

El primer aspecto a destacar es el uso de SBRDEs/SBRDs en problemas con alta dimensionalidad, medida ésta tanto por la existencia de un gran número de patrones/ejemplos como por un gran número de variables, y con relaciones de entrada-salida fuertemente no lineales. El primer elemento de la dimensionalidad (gran número de patrones) puede provocar problemas de eficiencia de los algoritmo junto con el aumento del tamaño de la BR. El segundo elemento (gran número de variables) provoca directamente un aumento del número de reglas necesario para modelar el sistema. El aumento lineal de variables provoca un aumento exponencial en el número de reglas.

Este problema ha sido objeto de estudio reciente por diferentes autores utilizando AGs que introducen la selección de variables en reglas localmente o en el conjunto de variables globalmente. Los AGs están mostrando ser muy útiles tanto para la selección de características como para la selección instancias en el preprocesamiento de datos, y pueden ser una buena herramienta para la reducción de la dimensionalidad conjunta con la extracción de reglas fuzzy. Por otra parte, la definición de gramáticas libres de contexto adecuadas puede permitir el diseño de SBRDEs utilizando programación genética, extrayendo reglas que con pocas variables permitan modelar una parte importante del sistema cuando la dimensionalidad es alta.

Por la importancia de esta situación, abordar el uso de los SBRDs en problemas complejos de alta dimensionalidad, está siendo centro de atención de diferentes estudios, y merece ser estudiado diseñando modelos eficientes y eficaces para estos problemas.

Un segundo aspecto a destacar son los estudios recientes en el diseño de SBRDs que alcancen un buen equilibrio entre precisión e interpretabilidad, dos objetivos que suelen ser contradictorios en el modelado de sistemas [5,6].

Como ya se indicó anteriormente, los AG multiobjetivo se han utilizado recientemente para evolucionar SBRDs con ambos objetivos para obtener un conjunto de soluciones no dominadas.

Esta es igualmente una línea de trabajo que merece ser estudiada en profundidad por el creciente interés en la inclusión de varios objetivos, en particular objetivos asociados a la interpretabilidad del SBRD, una de las cualidades a resaltar de los SBRDs.

Por último, hay que hacer mención a un aspecto esencial para la evaluación de los SBRDs y los SBRDEs, y es la necesidad de procedimientos sistemáticos de evaluación de la calidad de éstos, utilizando procedimientos estadísticos de validación de resultados. La necesidad de problemas conocidos y estándar para la validación de resultados junto con bases de datos de resultados sobre ellos es igualmente una necesidad, lo cual permitiría a cualquier autor evaluar los resultados de sus propuestas frente a estudios previamente realizados.

3.4. KEEL 1.0 HERRAMIENTA SOFTWARE DE MINERÍA DE DATOS

Finalmente, me gustaría presentar brevemente el proyecto "KEEL: Entorno para la Extracción de Conocimiento basado en Algoritmos de Aprendizaje Genético y Evolutivo", TIC-2002-04036-C05. En el proyecto KEEL (Knowledge Extraction based on Evolucionar Learning) se está desarrollando la herramienta software KEEL 1.0 (en Java con código abierto y disponible para cualquier usuario). Ésta nos permite utilizar y construir diferentes modelos para Minería de Datos, y como característica importante hay que destacar que incluye una librería de algoritmos de aprendizaje evolutivo y de aprendizaje de SBRDs. En este proyecto, 3 de los 5 grupos de investigación están desarrollando software de aprendizaje de SBRDs (Granada, Jaén, Gijón).

También se está desarrollando una librería de problemas KEEL-dataset y una página WEB asociada que contiene las bases de datos de los problemas para aprendizaje no supervisado, clasificación y regresión, y que incluirá las particiones utilizadas junto con los resultados obtenidos por los diferentes algoritmos implementados en KEEL. La dirección WEB de acceso al proyecto es <http://sci2s.ugr.es/KEEL> y <http://keel.ugr.es>.

4 COMENTARIOS FINALES

Hay que resaltar la alta potencialidad de los SBRDEs debido a la alta capacidad de representación de diferentes componentes de los SBRDs. Existe un amplio desarrollo de estos modelos en los últimos años que muestra la madurez del tema, e igualmente existen múltiples

cuestiones y problemas abiertos que aventuran un interesante futuro en la evolución del mismo.

Me gustaría terminar con una frase de David Goldberg [13] sobre la integración diferentes metodologías/componentes en un Sistema Inteligente Híbrido, que no está en una simple combinación. Indica que el futuro de la computación inteligente “*lies in the careful integration of the best constituent technologies*”. De acuerdo con Goldberg, el diseño de futuros SBRDEs debe integrar el potencial de abstracción y representación de conocimiento de los sistemas difusos junto con la potencia de los AGs para la búsqueda de buenas soluciones en espacios complejos, lo que nos lleva a un diseño sofisticado que va más allá de una mera puesta en conjunto de ambos.

Agradecimientos

La escritura de esta conferencia plenaria no habría sido posible sin el trabajo desarrollado en SBRDEs con diferentes coautores en estos 10 años dedicados a su estudio. Coautores a los que he tenido la satisfacción de dirigir la tesis doctoral en el desarrollo de SBRDEs (Oscar Cerdón, María José del Jesus, Antonio Peregrin, Pedro Villar, Igor Zwir y Jorge Casillas), y otros coautores con los que ha sido un placer trabajar (Manuel Lozano, Rafael Alcalá, José Luis Verdegay, Luciano Sánchez, Luis Magdalena, Frank Hoffmann y Fernando Gomide).

Este trabajo ha sido escrito bajo la financiación de la CICYT con el proyecto TIC2002-04036-C05-01.

Referencias

- [1] E. Alba, M. Tomassini. Parallelism and evolutionary algorithms. *IEEE Transactions on Evolutionary Computation* 6:5 (2002) 443-462.
- [2] A. Bastian, Identifying fuzzy models utilizing genetic programming, *Fuzzy Sets and Systems* 113:3 (2000) 333-350
- [3] E. Bernadó, J. M. Garrell. Accuracy-based learning classifier systems: models, analysis and applications to classification tasks. *Evolutionary Computation* 11:3 (2003) 209-238.
- [4] J. Casillas, O. Cerdón, M. J. del Jesus, F. Herrera. Genetic feature selection in a fuzzy rule-based classification system learning process for high dimensional problems. *Information Science* 136 (2001) 169-191.
- [5] J. Casillas, O. Cerdón, F. Herrera, L. Magdalena (Eds.). *Accuracy Improvements in Linguistic Fuzzy Modeling*. Springer-Verlag, 2003.
- [6] J. Casillas, O. Cerdón, F. Herrera, L. Magdalena (Eds.). *Interpretability Issues in Fuzzy Modeling*. Springer-Verlag, 2003.
- [7] C.A. Coello, D.A. Van Veldhuizen, G.B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Pub., 2002.
- [8] O. Cerdón, F. Herrera, F. Hoffmann, L. Magdalena. *GENETIC FUZZY SYSTEMS. Evolutionary tuning and learning of fuzzy knowledge bases*. World Scientific, 2001.
- [9] Cerdón O., Gomide F., Herrera F., Hoffmann F., Magdalena L., Ten years of genetic fuzzy systems: Current framework and new trends. *Fuzzy Sets and Systems* 41:1 (2004) 5-31.
- [10] K. Deb. *Multi-Objective using Evolutionary Algorithms*. John Wiley & Sons, 2001.
- [11] J.J. Eshelman. The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. *Foundations of genetic Algorithms 1*, G.J.E. Rawlin (Ed.), Morgan Kaufman, 1991, 265-283.
- [12] D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
- [13] D. E. Goldberg. A meditation on the Computational Intelligence and its future. Technical Report IlliGAL Report #2000019, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, 2000. Foreword of the Int. Symp. on Computational Intelligence, 2000.
- [14] D.E. Goldberg. *The Design of Competent Genetic Algorithms: Steps Toward a Computational Theory of Innovation*. Kluwer Academic Pub., 2002.
- [15] A. González, R. Pérez. Selection of relevant features in a fuzzy genetic learning algorithm. *IEEE Trans. on Systems, Man, and Cybernetics – Part B: Cybernetics* 31:3 (2001) 417-425.
- [16] F. Herrera, M. Lozano. Fuzzy adaptive genetic algorithms: design, taxonomy and future directions. *Soft Computing* 7:8 (2003) 545-562.
- [17] H. Ishibuchi, T. Yamamoto. Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining, *Fuzzy Sets and Systems* 141 (2004) 59-88.
- [18] Y. Jin. Fuzzy modeling of high-dimensional systems: complexity reduction and interpretability improvement. *IEEE Transactions on Fuzzy Systems* 8:2 (2000) 212-221.
- [19] P.L. Lanzi, An analysis of generalization in the XCS classifier system. *Evolutionary Computation* 7 (1999) 125-149.

- [20] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag, New York, 1992.
- [21] C. Moraga, E. Trillas, S. Guadarrama. Multiple-valued logic and artificial intelligence fundamentals of fuzzy control revisited. Artificial Intelligence Review 20 (2003) 169-197.
- [22] E. Perez, F. Herrera, C. Hernandez. Finding multiple solutions in job shop scheduling by niching genetic Algorithm. Journal of Intelligent Manufacturing 14:3-4 (2003) 223-239.
- [23] M. Valenzuela-Rendón, The fuzzy classifier system: motivation and first results. H.P. Schewefel, R.Männner (Eds.), Proc. First Int. Conf. on Parallel Problem Solving from Nature (PPSN I), Springer, Berlin, 1991, 330-334.