

# Algoritmo GA-P Difuso para la Generación de Controladores en Edificios Inteligentes

A. López<sup>1</sup>, F. Doctor<sup>2</sup>, L. Sánchez<sup>3</sup>

<sup>1</sup> Universidad de Oviedo. Departamento de Ingeniería Eléctrica.  
C. Universitario de Viesques. Ed. Departamental N<sup>o</sup>.2.  
33204, Gijón, Asturias, España.  
e-mail: antonio@isa.uniovi.es

<sup>2</sup> Computer Science Department. University of Essex.  
Wivenhoe Park. Colchester, CO4 3SQ, UK.  
e-mail: fdoctor@essex.ac.uk

<sup>3</sup> Universidad de Oviedo. Departamento de Informática.  
C. Universitario de Viesques. Ed. Departamental N<sup>o</sup>.1.  
33202, Gijón, Asturias, España.

e-mail: luciano@ccia.uniovi.es

*Resumen*— Recientemente, la investigación en el campo de la Inteligencia Ambiental está tomando gran auge en la comunidad científica. Dentro de este paradigma, el estado del arte recoge diferentes intentos de aplicación de técnicas de aprendizaje al control de los dispositivos de una vivienda en base a un aprendizaje no intrusivo realizado a partir de muestras de datos de diferentes sensores y del estado de los dispositivos de la vivienda. En este artículo se analizan las posibilidades de aplicación de algoritmos GA-P al diseño de tales arquitecturas de aprendizaje, particularizando la implementación a la obtención de controladores difusos de los dispositivos de la vivienda. Se muestran algunos resultados iniciales obtenidos a partir de datos reales tomados en un dormitorio inteligente. El algoritmo GA-P propuesto se muestra como un candidato adecuado en tal ámbito de aplicación, tanto desde el punto de vista de la precisión conseguida en el control de los dispositivos como en lo referente a la interpretabilidad lingüística de las soluciones.

*Palabras clave*— Algoritmos Genéticos, Programación Genética, GA-P, Inteligencia Ambiental, Edificios Inteligentes, Controladores Difusos.

## I. INTRODUCCIÓN

Recientemente, la investigación en el campo de la Inteligencia Ambiental (AmI) está tomando gran auge en la comunidad científica. La AmI se centra en el desarrollo de un nuevo paradigma en el cual la gente se desenvuelve en su vida cotidiana en un entorno digital que es consciente de su presencia y contexto y se muestra sensitivo, adaptativo y reactivo a sus necesidades, hábitos y emociones.

La Inteligencia Ambiental se fundamenta en el desarrollo de tres aspectos clave [1]:

- Computación Ubicua: Integración de microprocesadores en objetos cotidianos como mobiliario, electrodomésticos e, incluso, vestimenta o pinturas.
- Comunicación Ubicua: Interconexión de esos objetos para permitir una comunicación entre

ellos mediante redes inalámbricas o ad-hoc.

- Interfaces de Usuario Inteligentes: Diseño de interfaces que permitan el control e interacción entre los usuarios y los objetos de una forma natural (voz, gestos) y personalizada (dependiente de las preferencias y del contexto).

Como se puede deducir de la anterior clasificación, la mayor parte del trabajo desarrollado dentro del campo de la AmI se centra en consideraciones tecnológicas. Sin embargo, hacer que un dispositivo (p.e. una lámpara) sea capaz de realizar tareas de computación y de comunicarse con otros dispositivos no lo hace inteligente. El concepto clave para el desarrollo de la AmI radica en la forma en que el sistema aprende y se mantiene actualizado de forma autónoma de acuerdo a las necesidades del usuario. Se deben crear, en definitiva, sistemas adaptativos, capaces de responder a los deseos y necesidades de los usuarios de forma autónoma.

Es en este campo donde el aprendizaje automático gana fuerza y donde se pueden encontrar multitud de problemas de interés que no se centran en un mero desarrollo tecnológico. Uno de los principales problemas se centra en el diseño de arquitecturas de aprendizaje válidas para los objetivos perseguidos por el concepto de AmI. Tales arquitecturas han de tratar de adaptarse a los deseos de los usuarios mediante la observación de sus acciones, siendo clave para el éxito del enfoque que tal aprendizaje se realice de una forma no intrusiva.

Sin embargo, el diseño de arquitecturas genéricas es extremadamente complejo, dado que diferentes escenarios de aplicación del concepto de AmI podrían mostrar ciertas peculiaridades que le conferirían una entidad diferencial y por tanto susceptible de ser tratada aisladamente. El interés de este trabajo se basa en el análisis de una

posible arquitectura de aprendizaje válida para la implementación del concepto de AmI en el campo de edificios inteligentes.

La aplicación del concepto de AmI a los edificios inteligentes enlaza de forma directa con otras disciplinas como la domótica, que se define como la integración de tecnologías de la información en los edificios, y ha dado lugar a lo que se conoce como edificios inteligentes de tercera generación, donde el objetivo es automatizar el funcionamiento de algunos dispositivos cotidianos (luces, calefacción, etc.) evitando que el usuario tenga que realizar de forma tediosa tareas repetitivas que, en principio, podrían ser llevadas a cabo por un sistema inteligente, el cual, en base a la observación no intrusiva del comportamiento de los habitantes, podría tratar de reproducir sus acciones. En tales edificios, el entorno estaría compuesto por un conjunto de sensores que medirían diferentes magnitudes, que podría variar entre algunas de fácil detección (luminosidad interior y exterior, temperatura interior y exterior, etc.) y otras de más difícil estimación como podrían ser variables biológicas o emocionales. Asimismo, se dispondría de un conjunto de actuadores (pulsadores, reguladores) que podríamos modificar para actuar sobre diferentes variables del entorno, como el grado de luminosidad o la temperatura del habitáculo.

El problema fundamental se centra en el diseño de un sistema inteligente que en base a la observación de los valores de las variables y del estado de los actuadores (en un principio accionados por el usuario) fuese capaz con el tiempo de reproducir autónomamente esas acciones sin su intervención.

En el trabajo descrito en este artículo, se analiza la posibilidad del uso de una herramienta de aprendizaje basada en el algoritmo GA-P como núcleo de una arquitectura de aprendizaje offline. El algoritmo tratará de proporcionar controladores basados en bancos de reglas difusos adecuados para gobernar los actuadores del edificio.

## II. TRABAJOS PREVIOS

Quizá el trabajo más ambicioso en el área de la AiM sea el desarrollado en el proyecto Oxygen del MIT, el cual abarca todos los aspectos descritos anteriormente en gran profundidad. No obstante, centrándonos en aplicaciones más específicas referidas a edificios inteligentes, el estado del arte recoge diversas propuestas.

En [3] se utilizan sistemas multi-agente para controlar los servicios ofrecidos por un edificio inteligente, aunque no se considera el aprendizaje basado en el comportamiento de los ocupantes.

En [8] se utilizan redes neuronales artificiales para el control inteligente de la iluminación en un edificio. Se consigue reducir el consumo de energía aunque a veces a costa de delimitar el confort del usuario.

En [2] se utilizan métodos evolutivos para extraer modelos difusos a partir de datos, aplicándolo a la extracción del modelo de sistemas calefactores, de ventilación y basados en aire acondicionado (HVAC). El objetivo sería realizar a posteriori un control de los dispositivos en base a los modelos aprendidos.

En [4] se propone una técnica basada en un sistema multiagente para el control y aprendizaje en tiempo real de elementos de una vivienda inteligente.

En [7] se aborda ya una visión más amplia en cuanto a la construcción de una arquitectura de agentes distribuidos para computación ubicua. Una arquitectura similar se propone en [4], aunque este segundo enfoque ya se centra en detalles de implementación de los agentes atendiendo fundamentalmente a las necesidades de compatibilidad, mientras que en el primero tales conceptos no son considerados.

Asimismo, el estado del arte recoge numerosos desarrollos en el campo, aunque la mayoría de ellos no prestan demasiada atención al aprendizaje, describiendo en la mayoría de las ocasiones automatizaciones de componentes de una vivienda. Su mayor interés suele focalizarse en el uso de redes y acceso remoto.

## III. EL ALGORITMO GA-P PARA LA GENERACIÓN DE CONTROLADORES DIFUSOS

Los algoritmos GA-P [6] son un híbrido entre los algoritmos genéticos y la programación genética utilizados inicialmente en problemas de regresión simbólica. Los individuos se componen de dos partes: una representación arbórea para la componente estructural y un vector de parámetros para la componente paramétrica. A diferencia del GP canónico los nodos terminales del árbol no almacenan constantes numéricas sino identificadores lingüísticos que actúan como punteros al vector de parámetros (ver fig. 1).

Tanto la estructura arbórea como el vector de parámetros evolucionan durante el proceso en base a los operadores genéticos habituales como el cruce y la mutación. De esta forma, se realiza una búsqueda en paralelo de la componente estructural y paramétrica de un individuo óptimo.

En este trabajo se ha recurrido al uso de un algoritmo GA-P para la generación de controladores difusos. Tal algoritmo se aplicará sobre un

$$k_2 + k_1 * x = 4 + 3x$$

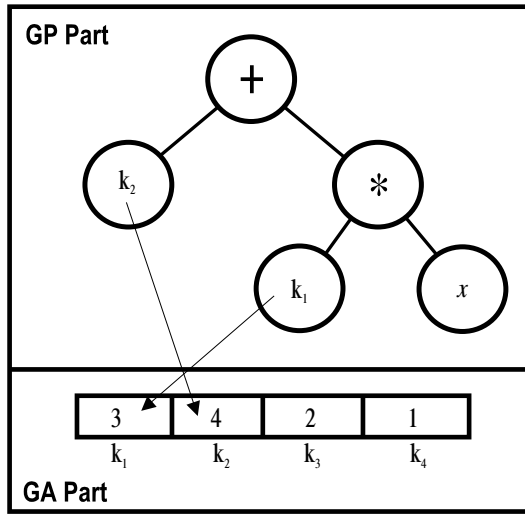


Fig. 1. Representación genérica en el algoritmo GA-P. Los individuos se componen de dos partes: una estructura arbórea y un vector de parámetros numéricos.

conjunto de  $n$  entradas y afectará a una única variable de salida. La generalización a la generación de controladores para varias salidas pasaría, en base a esta definición restringida, por la generación de un controlador independiente para cada salida.

A partir de un conjunto de muestras de las variables de entrada y de las variables de salida el algoritmo propondrá un banco de reglas para el control de las variables de salida.

En base a trabajos previos [9], se ha declarado una gramática del controlador difuso que permita definir el genotipo a utilizar. Para nuestros propósitos, un controlador difuso será una cadena válida según la gramática libre de contexto definida por las reglas de producción mostradas en la figura 2, donde  $x_1, \dots, x_m$  son las variables de entrada,  $y$  es la variable de salida,  $n_i$  es el número de particiones del espacio de entrada de las variables  $x_i$  y  $n_o$  es el número de particiones del espacio de salida.

Los límites de cada partición formarán el vector de parámetros. Para las entradas, estos parámetros marcarán los límites de las particiones trapezoidales o triangulares. Por otra parte, las particiones difusas del espacio de salida se representan por sus correspondientes singletons, por lo que las constantes del vector de coeficientes correspondientes indicarán el valor del singleton asociado a la partición correspondiente.

Un caso especial lo forman las variables binarias. En su manejo, el número de particiones del espacio de entrada es prefijado en dos, utilizando particiones cuadrangulares cuyo límite se fija



Fig. 3. El iDorm de la Universidad de Essex.

en  $1/2$  del rango de la variable. En cuanto a las salidas binarias, se prefijan de antemano los singletons a utilizar en  $1/4$  y  $3/4$  del rango de la variable. Al contrario del resto de elementos del vector de coeficientes, estos valores no se someten a los criterios de evolución del algoritmo.

La determinación del valor de la salida se realiza mediante el criterio habitual de media ponderada (ecuación 1), donde  $w_i$  es el valor de activación de cada condición.

$$output = \frac{\sum_{i=1}^{n_o} w_i \dot{K}_{o_i}}{\sum_{i=1}^{n_o} w_i} \quad (1)$$

Finalmente, en las salidas binarias se realiza un postprocesado de forma que una salida menor a 0.5 proporcionará un output de 0 y de 1 en caso contrario.

#### IV. EL iDORM DE LA UNIVERSIDAD DE ESSEX

El departamento de Computer Science de la Universidad de Essex en el Reino Unido dispone de un "dormitorio inteligente" (ver figura 3) denominado iDorm para sus investigaciones en el campo de edificios inteligentes. Trata de emular funcionalmente los dormitorios de los estudiantes disponibles para su alojamiento en las dependencias de la Universidad, dado que en ellos desarrollan prácticamente todos los aspectos de la vida cotidiana de una persona: descanso, trabajo y, en ocasiones, ocio. Se compone de una cama, un escritorio, una mesita, un armario y un ordenador personal con capacidades multimedia. Incorpora en segundo plano un conjunto de sensores (p.e. temperatura, ocupación, humedad y nivel de iluminación). Asimismo, proporciona el interfaz necesario para actuar sobre dispositivos del dormitorio como podrían ser un calefactor, aire acondicionado, diferentes puntos de luz regulables en intensidad y otros puntos de luz de carácter todo/nada.

CONTROLADOR: si CONDICION entonces ( $y$  es  $k_{o_1}$ )  
si CONDICION entonces ( $y$  es  $k_{o_2}$ )  
...  
si CONDICION entonces ( $y$  es  $k_{o_n}$ )

CONDICION: ASERCION<sub>1</sub> | ASERCION<sub>2</sub> | ... | ASERCION<sub>m</sub>  
| CONDICION O CONDICION  
| CONDICION Y CONDICION

ASERCION<sub>1</sub>: trapezio\_izquierdo( $x_1, K_{11}, K_{12}$ ) |  
triangulo( $x_1, K_{11}, K_{12}, K_{13}$ ) |  
triangulo( $x_1, K_{12}, K_{13}, K_{14}$ ) |  
trapezio\_derecho( $x_1, K_{1_{n-1}}, K_{1_{n1}}$ )

...

ASERCION<sub>m</sub>: trapezio\_izquierdo( $x_m, K_{m1}, K_{m2}$ ) |  
triangulo( $x_m, K_{m1}, K_{m2}, K_{m3}$ ) |  
triangulo( $x_m, K_{m2}, K_{m3}, K_{m4}$ ) |  
trapezio\_derecho( $x_m, K_{m_{nm-1}}, K_{m_{nm}}$ )

CONTANTES\_PARTICION:  $K_{11} \dots K_{1_{n1}} \dots K_{m1} \dots K_{m_{nm}} K_{o1} \dots K_{o_n}$

Fig. 2. Gramática del controlador difuso.

El iDorm incorpora la tecnología necesaria [5] para permitir la automatización de diferentes dispositivos y es utilizado en la actualidad para investigación en agentes inteligentes que permitan el control autónomo de los dispositivos existentes en el mismo.

Para este trabajo, se han utilizado datos reales obtenidos tras la estancia de un estudiante en el dormitorio durante varias jornadas. Se dispone de una muestra de datos con los valores de un conjunto de siete sensores y 8 actuadores. Parte de los sensores y actuadores tendrán una naturaleza analógica (rango de variación continuo) y otros tendrán un carácter digital (todo/nada):

- Sensores:
  - Nivel de iluminación exterior. (Analógico)
  - Nivel de iluminación interior. (Analógico)
  - Temperatura exterior. (Analógico)
  - Temperatura interior. (Analógico)
  - Detector de presión en la silla del escritorio. (Digital)
  - Detector de presión en la cama. (Digital)
  - Hora del día. (Analógico)
- Actuadores:
  - Nivel de acción sobre luz 1. (Analógico)
  - Nivel de acción sobre luz 2. (Analógico)
  - Nivel de acción sobre luz 3. (Analógico)
  - Nivel de acción sobre luz 4. (Analógico)
  - Estado de la lámpara del escritorio. (Digital)
  - Estado de la lámpara de la mesita de noche. (Digital)
  - Estado de la persiana. (Digital)
  - Estado del calefactor. (Digital)

## V. EXPERIMENTOS REALIZADOS

En esta sección se mostrarán algunos experimentos iniciales realizados a fin de determinar la validez del esquema propuesto en la generación de controladores difusos para los elementos del iDorm anteriormente descritos. Se comenzará por detallar la estructura de las pruebas, mostrando algunos resultados resumidos referidos al grado de acierto de los controladores generados en la estimación de las variables. Posteriormente, se analizarán los controladores propuestos en mayor detalle para dos de las variables a estimar, centrandó tal análisis en la respuesta del controlador en comparación con el valor real y en la interpretabilidad de los bancos de reglas que los conforman.

### A. Estructura de las Pruebas

Los experimentos se realizaron para generar controladores para la estimación de las cuatro variables analógicas y de las cuatro variables digitales disponibles.

En los experimentos descritos en este artículo se ha utilizado el algoritmo GA-P descrito con una población de 200 individuos que evoluciona durante 200 generaciones. La tasa de reproducción se ha fijado en el 10 %, las de cruce estructural y paramétrico en el 40 %, y las de mutación estructural y paramétrica en el 5 %. La profundidad máxima de los árboles manejados para la componentes estructural se ha fijado en 4 niveles.

Para la generación de los controladores se dispone de un total de 408 muestras obtenidas, como

se ha comentado anteriormente, por experimentación en el iDorm. Tal muestra ha sido subdividida en varios subconjuntos para la generación del controlador difuso. Parte de las muestras (2/3 del total) constituirá el subconjunto de entrenamiento, que se utilizará para guiar la evolución del algoritmo. Por otra parte, en experimentos iniciales se detectó cierta propensión del método al estancamiento en mínimos locales (característica en principio intrínseca a los algoritmos evolutivos). Para tratar de paliar este fenómeno, cada experimento se repite tres veces y se selecciona el mejor individuo de las tres ejecuciones en base a un segundo subconjunto de muestras, denominado de test, compuesto por un total de 1/6 de las muestras disponibles. Finalmente, se utiliza como medida de eficiencia el valor de aptitud obtenido a partir de un tercer subconjunto de muestras, denominada de validación, que proporcionará un valor indicativo del error esperado, ya que la información contenida en él no es utilizada en ningún momento durante el proceso de entrenamiento. Estos tres subconjuntos formarán lo que se denominará *configuración de muestras* del experimento.

Dada la clara temporalidad existente en la recogida de las muestras, los subconjuntos de entrenamiento, test y validación estarán formados por muestras seleccionadas aleatoriamente de las 408 muestras disponible. No se admite la repetición de muestras en diferentes subconjuntos.

Asimismo, a fin de obtener valores referidos a la eficiencia del método con independencia de la información muestral utilizada, se han generado un total de seis configuraciones de muestras diferentes. Cada experimentos se aplicará sobre cada una de las configuraciones de muestras y se mostrarán como resultados valores promediados.

La medida de aptitud utilizada es el error cuadrático medio entre el valor real de la salida y el sugerido por el controlador para los subconjuntos de muestras utilizados. Aunque en un principio se realizaron pruebas donde para las salidas binarias se utilizaba como medida de aptitud el número de aciertos, se comprobó que el uso del error cuadrático medio proporcionaba resultados similares en cuanto a eficiencia, por lo que se adoptó tal medida como medida de aptitud por coherencia con la usada en salidas analógicas.

Finalmente, dado que es posible especificar a priori el número de particiones deseadas para cada una de las variables analógicas, se siguió el criterio de aplicar el mismo número de particiones a las variables de entrada y de salida, variando tal número de particiones entre experimentos. En concreto, se han realizado experimentos don-

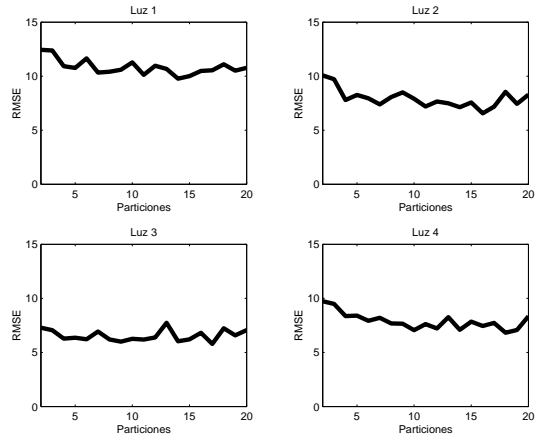


Fig. 4. Efecto del número de particiones de los espacios de las variables analógicas sobre el error obtenido en la estimación del valor de las salidas analógicas. Datos promediados sobre seis configuraciones de muestras.

de el número de particiones utilizado varía entre 2 y 20. En este punto es necesario recordar que tal número solamente afecta a las variables de naturaleza analógica, dado que en las de carácter digital el número de particiones se fija en 2.

### B. Análisis Conjunto

Las figuras 4 y 5 muestran respectivamente el promedio de errores (sobre las diferentes ejecuciones basadas en las seis configuraciones de muestras) obtenido para las cuatro variables de salida de carácter analógico y para las cuatro variables de salida de carácter binario. Se ha variado el número de particiones del espacio de las variables analógicas (tanto para las de entrada como para las de salida) entre dos y veinte. Se puede comprobar cómo el número de particiones influye sobre el error obtenido, pero tal influencia no parece demasiado significativa en el uso de un número de particiones elevado.

Por otra parte se observa cómo algunas variables se predicen mejor que otras ateniéndonos al error obtenido a partir del subconjunto de validación.

### C. Análisis de los Controladores

En esta sección se muestran los controladores obtenidos para una de las variables analógicas (luz 3) y una de las variables digitales (estado de la persiana). Hemos escogido una de las soluciones obtenidas utilizando tres particiones del espacio de entrada de las variables analógicas. Si bien se podrían encontrar experimentos donde la solución obtenida fuese más precisa por el incremento en el número de particiones, se han seleccionado los mostrados dado que un bajo número de particiones favorece la interpretabilidad de los

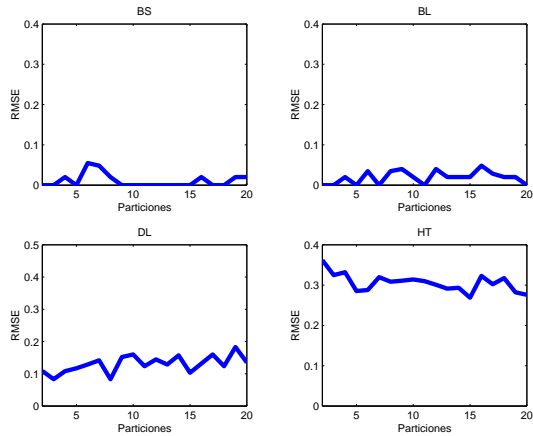


Fig. 5. Efecto del número de particiones de los espacios de las variables analógicas sobre el error obtenido en la estimación del valor de las salidas digitales. Datos promediados sobre seis configuraciones de muestras.

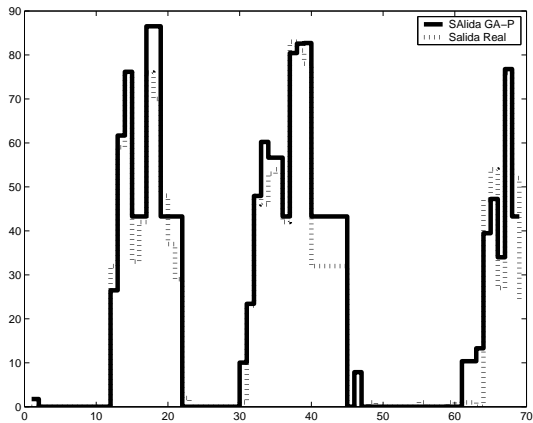


Fig. 6. Comparación del valor sugerido por el controlador difuso y el valor real para luz1.

bancos de reglas del controlador.

Las figuras 6 y 7 muestran una comparación entre los valores sugeridos por el controlador difuso y los valores reales de las variables para la muestra de validación. Se muestra cómo, en el caso de la variable analógica, se realiza una estimación aceptable y en el caso de la variable digital se ha acertado en el 100% de los casos utilizados como subconjunto de validación. Por otra parte, las figuras 8 y 9 muestran los bancos de reglas de los controladores hallados. Dado que el espacio de las variables se ha particionado en tres (variables analógicas) o dos (variables digitales), nos referiremos a tales particiones como BAJO, MEDIO o ALTO. Ambos controladores parecen en principio acordes con la estructura esperada de los mismos.

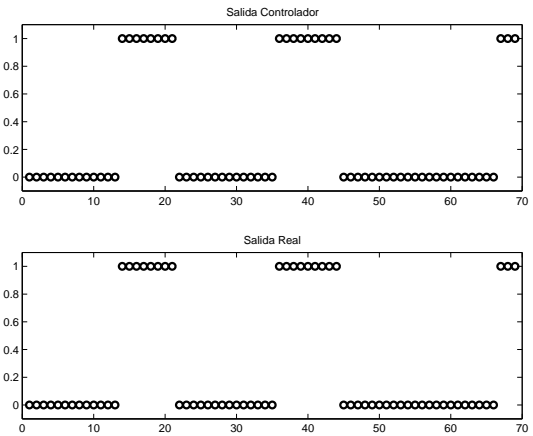


Fig. 7. Comparación del valor sugerido por el controlador difuso y el valor real de la persiana.

```
si (LuzExterior es Alto)
entonces (y es BAJO)
```

```
si (Hora es BAJO) Y
(LuzExterior es BAJO o MEDIO)
entonces (y es MEDIO)
```

```
si (LuzInterior es BAJO)
entonces (y es ALTO)
```

Fig. 8. Controlador para Luz 1.

## VI. CONCLUSIONES Y TRABAJO FUTURO

Si bien el estudio desarrollado en este trabajo no ha sido exhaustivo, se ha mostrado que un algoritmo GA-P es un buen candidato a la generación de controladores para elementos de una vivienda inteligente. El algoritmo se muestra capaz de realizar un ajuste aceptable de los valores de los actuadores y proporciona bancos de reglas de fácil interpretación que, a priori, parecen capturar la lógica de activación de los dispositivos. Tal interpretabilidad se ve favorecida por la obtención de resultados aceptables utilizando un número reducido de particiones de los espacios de las variables analógicas, lo que repercute en una disminución del conjunto de terminales usado en la búsqueda y, en consecuencia, permite obtener

```
si (PresionCama es Alto)
entonces (y es Bajo)
```

```
si(LuzExterior es Bajo)
entonces (y es Alto)
```

Fig. 9. Controlador para DL.

bancos de reglas más simples.

Por otra parte, analizando los resultados obtenidos en la estimación de las diferentes variables de salida analizadas, se puede comprobar cómo para determinadas salidas se realiza un ajuste más preciso que para otras. Si bien este fenómeno aún no se ha estudiado en profundidad, sería interesante estudiar su origen, ya que podría deberse a la insuficiencia de muestras que permitan una estimación más precisa de su estado, a la insuficiencia de información sensorial indicativa del estado de la salida analizada, lo cual haría muy difícil realizar tal predicción por disponer de información incompleta o, quizá, a un uso de carácter no completamente determinista de tales dispositivos.

Las líneas de posible trabajo futuro son múltiples.

Quizá un primer punto pasaría por la comparación con otros métodos de aprendizaje, a fin de validar el ajuste realizado por el algoritmo GA-P. Esta comparación podría extenderse más allá de un simple análisis cuantitativo, donde quizá el GA-P podría verse superado por otras técnicas, ya que se podrían analizar otros aspectos cualitativos como la posibilidad de interpretación analítica de los controladores obtenidos, donde quizá el método GA-P fuese superior a otros enfoques.

La propia interpretabilidad de las soluciones podría hacer pensar en el método como un buen candidato a la sugerencia de bancos de reglas iniciales para el control de la vivienda inteligente, que podrían ser refinados mediante interacción con el usuario. Asimismo, las posibilidades de introducir conocimiento a priori dentro del esquema analizado son muy amplias dada la versatilidad del algoritmo de búsqueda, lo cual podría suponer una clara ventaja respecto a otros métodos de aprendizaje. Un caso simple, sería detallar a priori las particiones exactas a utilizar para algunas variables (p.e. podría parecer lógico dividir la hora del día en unos periodos concretos acordes con la actividad de los habitantes del edificio). Esto podría simplificar la búsqueda a la vez que favorecer, una vez más, la interpretabilidad de las soluciones.

Por otra parte, sería necesario considerar posibles extensiones del algoritmo que lo hiciesen válido para llevar a cabo tareas de aprendizaje por refuerzo, aunque este es quizá uno de los aspectos aún no completamente resueltos en el aprendizaje mediante métodos evolutivos. Tal y como se ha detallado, el método sería válido para sugerir bancos de reglas en base a un conjunto de datos históricos. No obstante, en el entorno

de aplicación que se ha descrito, el aprendizaje debería efectuarse a lo largo del ciclo de vida del sistema, ya que las necesidades o comportamientos de los usuarios de la vivienda inteligente se verían modificados con gran probabilidad en diferentes periodos de uso.

Otro posible campo de trabajo sería introducir consideraciones dinámicas en los controladores utilizados, dado que el esquema de búsqueda se ha diseñado para la generación de controladores de variables estáticas.

## REFERENCIAS

- [1] Jari Ahola. Ambient intelligence. *ERCIM News Online Edition*, (47), 2001.
- [2] P.P. Angelov and R.A. Buswell. Automatic generation of fuzzy rule-based models from data by genetic algorithms. *Information Sciences Informatics and Computer Science: An International Journal*, 150(1-2):17–31, 2003.
- [3] M. Boman, P. Davidsson, N. Skarmeeas, and K. Clark. Energy saving and added customer value in intelligent buildings. In Hyacinth S. Nwana and Divine T. Ndumu, editors, *Proceedings of the 3rd International Conference on the Practical Applications of Agents and Multi-Agent Systems (PAAM-98)*, pages 505–516, London, UK, 1998.
- [4] H. Hagraas, V. Callaghan, G. Clarke, M. Colley, A. Pouds-Cornish, A. Holmes, and H. Duman. Incremental synchronous learning for embedded-agents operating in ubiquitous computing environments. In V. Loia, editor, *Soft Computing Agents. A New Perspective for Dynamic Information Systems*. 2002.
- [5] A. Holmes, H. Duman, and A. Pounds-Cornish. The idorm: Gateway to heterogeneous networking environments. Technical report, Intelligent Inhabited Environments Group, University of Essex, 1998.
- [6] Les M. Howard and Donna J. D'Angelo. The GA-P: A genetic algorithm and genetic programming hybrid. *IEEE Expert*, 10(3):11–15, June 1995.
- [7] Nelson Minar, Matthew Gray, Oliver Roup, Raffi Krikorian, and Pattie Maes. Hive: Distributed agents for networking things. In *First International Symposium on Agent Systems and Applications (ASA'99)/Third International Symposium on Mobile Agents (MA'99)*, Palm Springs, CA, USA, 1999.
- [8] M. Mozer. The neural network house: an environment that adapts to its inhabitants. In *Proceedings of the AAAI Spring Symposium on Intelligent Environments*.
- [9] Luciano Sanchez, Ines Couso, and J. A. Corrales. Combining operators with sa search to evolve fuzzy rule based classifiers. *Information Sciences*, 136(1-4):175–191, 2001.