An Evolutionary Algorithm for the Off-Line Data Driven Generation of Fuzzy Controllers for Intelligent Buildings*

A. López[1], L. Sánchez[2]
[1]Electric Engineering Department
[2]Computer Science Department
University of Oviedo
33204, Gijón, Asturias, Spain
[amlopez,luciano]@uniovi.es

F. Doctor, H. Hagras, V. Callaghan
Department of Computer Science
University of Essex.
CO4 3SQ, Colchester, UK
[fdocto,hani,vic]@essex.ac.uk

Abstract – *Ambient Intelligence is nowadays an active research field. As a key matter of this concept, several approaches have been proposed for the development of learning architectures for the control of the devices in an intelligent building. In this paper, an evolutionary algorithm is analyzed as a candidate for the initial phases of the design of such architectures: fuzzy controllers for the devices are off-line induced from data sampled from the environment. We will show results obtained using real data gathered from the Essex intelligent dormitory. The proposed algorithm seems to be suited for the task, both due to its accuracy and for the easy and meaningful linguistic interpretation of the solutions it produces.*

Keywords: Ambient Intelligence, Intelligent Buildings, Fuzzy Control, Evolutionary Algorithms, Data Driven Generation.

1 Introduction

Ambient Intelligence (AmI) is nowadays an active research field. AmI deals with the development of a new paradigm where people are immersed in a digital environment that is aware of their presence and context, and which is sensitive, adaptive and reactive to their desires, habits and emotions.

AmI builds on three key concepts: Ubiquitous Computing (integration of computers in daily objects), Ubiquitous Communication (communications among them) and Intelligent user interfaces (voice, gestures). A great deal of work is based on technological developments. Nevertheless, to give a device (e.g. a lamp) processing and communication capabilities does not make it intelligent. A key concept to answer in AmI is how the system can learn about the user behaviors, and be constantly adapted to them. Autonomous adaptive learning systems must be developed to respond to user preferences and desires.

Nevertheless, the design of generic architectures is quite complex. Different AmI scenarios (e.g. industry, workplace, home, machine to machine interaction, etc.) present special characteristics that could make necessary a specific design. In particular, learning systems for intelligent buildings present a unique and specific set of characteristics and our aim is to contribute to the development of these intelligent mechanisms to address those characteristics.

The goal in intelligent buildings is to control the environmental via a set of devices using an intelligent agent which should work in a non-intrusive manner to satisfy the preferences of the user. The agent should observe the user's interaction with the building and learn how to pre-empt them automatically from the information provided by a set of environmental sensors (temperature, humidity, etc) and by the states of the actuators (buttons, heaters, etc).

The main motivation for the work described in this paper is to design a method for the off-line data driven automatic generation of controllers for heterogeneous devices in an intelligent building. Fuzzy controllers have been selected for the internal structure of the agent for mainly two reasons. First, we intend to get interpretable controllers, which could allow one to test if the controller is working properly, performing operations that really reflect the user behavior. Second, imprecision is inherent in the sensor measurements and the user's management of the actuators in the environment. This way, fuzzy controllers seem to be a good alternative because imprecision handling is inherent to the technique and their operation is quite close to the human reasoning process.

These fuzzy controllers, implemented in the agent, could be used later by adaptive learning architectures

*0-7803-8566-7/04/$20.00 © 2004 IEEE.

proposed in the state of the art (e.g. [5]) to control the building, and contribute in improving the man-machine interaction of the user with his environment and to the implementation of adaptive interfaces between the user and the devices of the building.

Among several alternatives analyzed and discussed in a later section, we will choose an evolutionary algorithm based on GA-P [8] for the data based generation of the fuzzy controllers. Let us say for the time being that this technique allows an easy linguistic interpretation of the solutions, a premise of our work.

To validate the technique, several experiments have been performed using real data gathered from the Essex Intelligent Dormitory (iDorm) described in a later section. Results are analyzed in terms of their accuracy at fitting the input-output data and their linguistic interpretability. A preliminar comparison with results obtained from the well known ANFIS system [9] is also described, in order to settle the validity of the GA-P.

The rest of this paper is organized as follows: In section 2 we give an introduction and a brief discussion about the state of the art related to the development of architectures for the intelligent control of devices in intelligent buildings and to the data driven generation of fuzzy controllers. In section 3 the proposed GA-P algorithm is described. Section 4 describes the performed experiments and an analysis of the results. The analysis pays special attention to the linguistic interpretation of solutions and to the numerical accuracy of the technique. Comparative results with ANFIS are shown later. Section 5 provides some conclusions and future directions of work.

2 State of the Art and Discussion

Different approaches for the intelligent control of devices in an intelligent building have been proposed in the state of the art. In [3], they proposed a multi-agent system for the control of buildings, but their approach did not pay much attention to automatic learning. In [14], they used artificial neural networks for the intelligent control of lighting in a building. In [2], they used evolutionary algorithms to develop HVAC (Heating, Ventilation and Air Conditioned) models with control purposes. In [13] the design of multi-agent systems for the real time control of a building based on machine learning was proposed.

As said in the introduction, from our point of view the linguistic interpretability of the controller structure is of high relevance. So we think that neural network or any other black box based approach are not very suited for the task due to the difficulty of analyzing their internal structure. In conclusion, we have focused in a previous work [5], where the Intelligent Inhabited Environments Group of the University (IIEG) of Essex has developed an adaptive system based on the use of fuzzy controllers for the real time control of a room environment. Starting with an initially learnt set of fuzzy rules (an empty rule set in the extreme case), the adaptive learning system is capable of adapting and extending the rules online over a long period of time to fit the changing environmental conditions and user preferences. What we try in this paper is to describe a system for the generation of initially fitted controllers. This way, the learning period could be reduced in time. The more fitted the initial rule set, the shorter the time to adapt to the user desires.

Regarding the data driven generation of fuzzy controllers, several approaches can be found in the state of the art, many of them based on Soft Computing technologies, such as Genetic Algorithms (GA), Genetic Programming (GP) or Artificial Neural Networks (NN). GAs can be applied to the generation of fuzzy controllers in the optimization of the set of parameters of a given rule set, although the problem of extracting rules from data has been also addressed [6]. Regarding NN, ANFIS (a NN where the internal structures represent fuzzy relations) is one of the most popular methods [9]. GP can be used also for the generation of rule sets by means of syntactic trees, where each branch of the tree represents a rule [1]. Nevertheless, more advanced approaches can be also found. Such is the case of EfuNN systems, where rules and fuzzy sets are generated by means of a neural network using genetic algorithms for parameter optimization [10]. GA-P algorithms can be also used for this task. GA-P algorithms are a hybrid between GP and GA, where the GP is used to generate a set of syntactic rules and GA is used to tune the numerical coefficients [4].

In the work described in this paper, no initial rule set is available in advance. So, NN and GP based systems seem to be adequate. Nevertheless, the linguistic interpretability of the solutions is a premise for our work. This seems to be a handicap for NN based systems, that usually pose the problem of the difficulty of interpreting the internal structure of the network once trained. So, a GP based system was selected for the automatic generation of controllers. But, in the context of application, both the rule set and the membership functions are unknown. So, it was decided to apply the GA-P algorithm with the aim of getting both components of the controller.

3 GA-P Generation of FLCs

A classical single output fuzzy logic controller (FLC) [12] divides the space of characteristics of the output variables into a series of linguistic labels. It also implements a set of rules for the activation of such labels in the basis of the inputs to the system. The final result, i.e. the action to take over the output variables, would be a combination of the activation values for the different labels proposed by the rules.

GA-P algorithms [8] are hybrids between a genetic

CONTROLLER:	if CONDITION then (y is k_{o_1}) if CONDITION then (y is k_{o_2}) ... if CONDITION then (y is $k_{o_{n_o}}$) | if CONDITION then (y is 0.25) if CONDITION then (y is 0.75)
CONDITION:	ASSERT$_{a_1}$ | ASSERT$_{a_2}$ | .... | ASSERT$_{a_m}$ |ASSERT$_{d_1}$ | ASSERT$_{d_2}$ | .... |ASSERT$_{d_{m'}}$ | CONDITION OR CONDITION | CONDITION AND CONDITION
ASSERT$_{a_1}$:	left_trapezium(a_1, K_{11}, K_{12}) | triangle(a_1, K_{11}, K_{12}, K_{13}) | triangle(a_1, K_{12},K_{13}, K_{14}) | right_trapezium(a_1, $K_{1_{n1-1}}$, $K_{1_{n1}}$)
...	
ASSERT$_{a_m}$:	left_trapezium(a_m, K_{m1}, K_{m2}) | triangle(a_m, K_{m1}, K_{m2}, K_{m3}) | triangle(a_m, K_{m2},K_{m3}, K_{m4}) | right_trapezium(a_m, $K_{m_{nm-1}}$, $K_{m_{nm}}$)
ASSERT$_{d_1}$:	zero(d_1) | one(d_1)
...	
ASSERT$_{d'_m}$:	zero($d_{m'}$) | one($d_{m'}$)
PARTITION_CONSTANTS:	$K_{11} \dots K_{1n1} \dots\dots K_{m_1} \dots K_{m_{nm}} K_{o_1} \dots K_{o_{n_o}}$

Figure 1: Fuzzy Controller Grammar.

algorithm and a genetic program initially applied in symbolic regression problems. Individuals in GA-P are composed of two parts: a tree and a vector of numerical parameters. In contrast with canonical GP, terminals never store numerical values but linguistic identifiers that act as pointers to positions in the vector of numerical parameters.

Both the tree and the vector of coefficients are evolved during the process in the basis of the usual genetic operators (reproduction, crossover and mutation). This way, a parallel search in both the structural and parametrical components is made.

In our work, a generational GA-P for the generation of fuzzy controllers is used. The structure represents the rule set and the parametrical component represents the coefficients for the partitions of the fuzzy sets. Reproduction, structure (crossover and mutation) and parametric (crossover and mutation) operators are used to evolve the population[11].

The algorithm is applied over a set of n inputs (m analog inputs and m' binary inputs) and takes a single output. To generalize the technique to the generation of controllers for several outputs, it should be applied independently for each output.

On the basis of previous work [15], a grammar for the fuzzy controller has been declared to define the genotype of the individuals (see fig. 1). From our purposes, a fuzzy controller will be a valid chain from the context free grammar defined by the production rules shown in fig. 1, where $a_1, \dots a_m$ are analog inputs, $d_1, \dots d_{m'}$ are binary inputs, n_i is the number of partitions for the input space of analog variable a_i, y is the output variable and n_o is the number of partitions for the output in case of this to be analog. For binary variables (input and output), singleton membership functions are used for the partitions.

Figure 2: The intelligent dormitory.

The output value of the controller is calculated by means of a weighed average of the output of each condition. If the output variable is binary, a post-processing is applied rounding the result to 0 or 1.

4 Experiments

The experimental tests described in this paper are based on real data taken from the Essex Intelligent Dormitory (iDorm) (see fig. 2). It is used for experiments in the field of intelligent buildings [7]. A set of sensors and actuators are available for the intelligent control of the devices of the iDorm.

A PhD student had stayed in the iDorm for several days. During his stay his actions were monitored through seven sensors (External Light Level, Internal Light Level, External Temperature, Internal Temperature, Desk Chair Pressure, Bed Pressure and Time). The user controlled ten actuators (Light 1 Action Level, Light 2 Action Level, Light 3 Action Level, Light 4 Action Level, Desk Lamp State, Bed Light State, Blind State, Heating State, MS Word State and MS Media Player State).

In the following sections, a detailed description of the structure of the tests performed with GA-P is given. Next, results are analyzed in terms of the deviation between the value proposed by the fuzzy controller and the real value of the actuator. Two controllers will be also analyzed in depth. In the last part, preliminar comparative results with ANFIS are shown.

4.1 GA-P Test Structure

The GA-P was applied using a population of 200 individuals that evolves for 200 generations. Aptitude is calculated by the Root Mean Squared Error (RMSE). The reproduction fraction was set at 0.1, structural crossover and parametric crossover were set at 0.4, and structural and parametric mutation fractions were set at 0.05. Maximum height for the evolved individuals was set at 4 levels. Initial experiments showed that the GP sometimes converges to local minima. So, each experiment was repeated three times and the best solution from the three was selected.

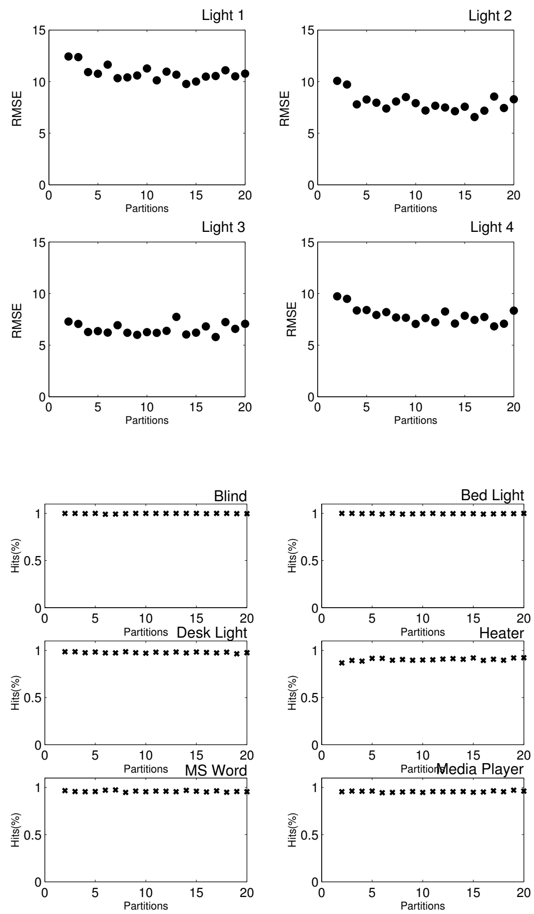

Figure 3: Effect of the number of partitions in the search results. Averaged results over the 6 sample configurations.

The available sampled data was composed of 408 samples. It was divided into three subsets making up what we call a *configuration sample*: training (2/3 of the total used for training), validation (1/6 of the total used to select the best from the three available solutions) and test (1/6 of the total used to analyze the accuracy of the best solution found). All subsets were randomly selected from the 408 available samples. Repetition was not allowed. In order to get representative values, six sample configurations were generated. Every single experiment was applied over the six sample configurations and averaged results are shown.

A last consideration was that related to the number of partitions to use for each analog variable. It was decided to specify the same number (k) of partitions for all the analog variables used in a single experiment ($n_i = n_o = k$), repeating the experiment using different values for k (from 2 to 20).

4.2 Summarized Analysis

Fig. 3 shows the averaged results over the test sample in the six sample configurations for the ten outputs. The X axis shows the number of partitions of the analog space used in each experiment. For binary outputs, the percentage of hits is shown. For analog outputs the RMSE is shown.

The first conclusion is that there are no significant variations in the results using different numbers of partitions. Although this analysis is not rigorous, it is enough for the moment to say that good results can be obtained using a low number of partitions of the analog spaces. This is an important matter, knowing that the number of partitions has a direct influence in the number of terminals used in the search (on the basis of the grammar). And a large number of terminals usually degrades the linguistic interpretability of solutions. In conclusion, and in order to get interpretable solutions, we take as a fact that good results can be found using a low number of partitions.

Analyzing in more depth the results for one analog output (e.g. light 3) it can be seen how the RMSE is about 7%. Given that the variable can vary from 0 to 100, the controller seems to perform a good estimation. In other words, if a deviation of about 7% (an admissible value in advance) was allowed in the control of the light, the success of the controller should be near optimum. Results for the rest of analog outputs (lights 1, 2 and 4) are in the same range. For binary outputs the percentage of hits is always about the 100%.

4.3 Analysis of Controllers

At this section, the controllers for one analog output (light 3) and one binary output (blind) are shown. Controllers that use three partitions of the analog variables spaces are shown (labelled as LOW, MEDIUM and HIGH). Although more accurate controllers can be found in the bank of experiments, the use of this low number of partitions favors the interpretability of solutions.

Fig. 4 shows a comparison between the output of the selected fuzzy controllers and the real output values for both devices, using the test sample. It can be seen how a good approximation is made in the prediction of light 3 and a optimum prediction is made for the state of the blind.

Fig. 5 shows also the rule sets for both controllers. These controllers were shown to the student who inhabited the iDorm and analyzed. Conclusions, detailed next, allow to settle that the controllers seem to be really capturing the student preferences when handling both devices.

The existence of variables whose influence could be predicted in advance can be seen. Nevertheless, in the controller for light 3 a strange variable appears: external temperature. The reason can be that the controller

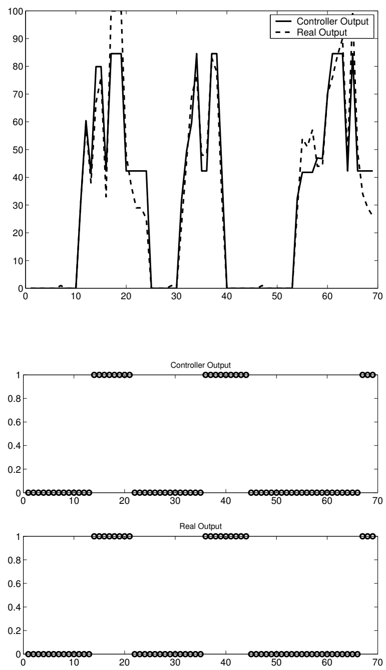

Figure 4: Fuzzy controller and real outputs comparison for light 3 (top) and blind (bottom).

needs to make divisions with the same external light level: as the night advances, external light level is the same but external temperature decreases.

Regarding the controller for light 3, it can be seen that if the external light level is MEDIUM, light 3 intensity is MEDIUM. If external light level is LOW, light 3 intensity is HIGH. Nevertheless, it can be seen also that when the external light level is LOW, combined with other circumstances such as, for example, that the user is in bed (bed pressure is HIGH), the light 3 intensity must be LOW. In this circumstance, rules one and three interact to give an intermediate light level intensity. The student explained this effect as he likes to have this light on to a medium intensity during the night.

Regarding the blind control, it is LOW (closed) when the user is on the bed, and HIGH (open) when the external light level is LOW.

4.4 ANFIS Comparison

A preliminar comparison with ANFIS is shown. Six sample configurations have also been used for this experiments, with the difference that no validation subset is used. So, training sets are composed of 2/3 of the available samples and test subsets are composed of the remaining 1/3 of the samples. ANFIS has been applied to get controllers for the ten outputs using a cluster radii varying from 0.3 to 2.0 with a step of 0.1.

ANFIS and GA-P results for all the ten outputs have been normalized and averaged over the six sample configurations. Fig. 6 shows a comparison between both techniques. No significant differences in accuracy are observed among the techniques. But differences can be observed in other matters. ANFIS generates in some circumstances a high number of rules (up to 39 in some experiments) which can make it difficult for comparing the linguistic interpretations with the GA-P. But ANFIS is in general much faster than the GA-P.

if (BedPressure **is** HIGH) **and**
(InternalLightLevel **is** LOW) **and**
(ExternalLightLevel **is** LOW) **and**
(ExternalTemprature **is** LOW)
then (Light3 is LOW)

if (ExternalLightLevel **is** MEDIUM)
then (Light3 is MEDIUM)

if (ExternalLightLevel **is** LOW)
then (Light3 is HIGH)

(a) Light 3

if (BedPressure **is** HIGH)
then (BlindState **is** LOW)

if (ExternalLightLevel **is** LOW)
then (BlindState **is** HIGH)

(b) Blind State

Figure 5: Rule sets for two controllers.

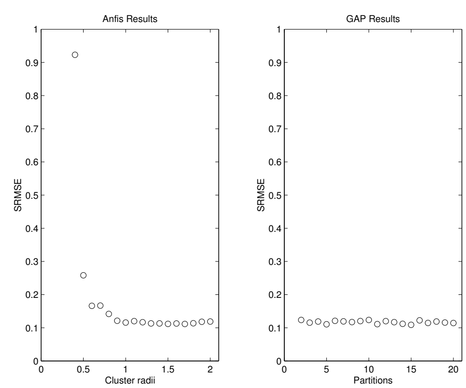

Figure 6: ANFIS vs. GA-P. SRMSE stands for Scaled Root Mean Square Error.

5 Conclusions and Future Work

Although the work described at this paper has not been too exhaustive, it can be used to state that the GA-P is a good candidate technique for the off-line generation of fuzzy controllers for an intelligent building. The accuracy of the controllers is good and the rule sets are easy to interpret. They seem to be capable of extracting from the data the knowledge about the logic for the controlling of the devices based on the user preferences. A preliminar comparison with ANFIS has been also made. Though the differences are not significant, each of them present advantages over the other.

With regard to future work, further analysis could be based on a comparison of the GA-P approach with a wider spectrum of learning techniques, not only ANFIS. Another matter to investigate is to use the rules provided by the GA-P as the initial set for an adaptive technique for the real time control of a building (it could be applied to the iDorm). The impact of starting from a learned set of rules could be analyzed.

In the immediate future, several extensions to the algorithm are planned. The most relevant is perhaps the introduction of reinforcement learning mechanisms in the scheme. The GA-P described in this paper is only valid for the generation of initial controllers. Adding reinforcement mechanisms, the GA-P could also be used also for online adaptation.

Acknowledgements

The authors would like to thank the IIEG of the University of Essex for their support. They would like also to thank the Science and Technology department of the Spain Government for the support of the work through the project TIC2002-04036-C05-05.

References

[1] Enrique Alba, Carlos Cotta, and Jose J. Troyo. Type-constrained genetic programming for rule-base definition in fuzzy logic controllers. In John R. Koza, David E. Goldberg, David B. Fogel, and Rick L. Riolo, editors, *Genetic Programming 1996: Proceedings of the First Annual Conference*, pages 255–260, Stanford University, CA, USA, 28–31 July 1996. MIT Press.

[2] P.P. Angelov and R.A. Buswell. Automatic generation of fuzzy rule-based models from data by genetic algorithms. *Information SciencesInformatics and Computer Science: An International Journal*, 150(1-2):17–31, 2003.

[3] M. Boman, P. Davidsson, N. Skarmeas, 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] S. Garcia, F. Gonzalez, and L. Sanchez. Evolving fuzzy rule based classifiers with GA-P: A grammatical approach. In R. Poli, P. Nordin, W. B. Langdon, and T. C. Fogarty, editors, *EuroGP'99*, volume 1598 of *LNCS*, pages 203–210, Goteborg, Sweden, 26-27 May 1999. Springer-Verlag.

[5] H. Hagras, 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.

[6] Francisco Herrera and Luis Magdalena. Genetic fuzzy systems: A tutorial.

[7] 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.

[8] 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.

[9] Jyh-Sing Roger Jang. ANFIS: Adaptive-Network-Based Fuzzy Inference System. *IEEE Transactions on Systems, Man, and Cybernetics*, 23:665–684, 1993.

[10] N. Kasabov. Evolving fuzzy neural network for supervised/unsupervised on-line, knowledge-based learning, 2001.

[11] A.M. Lopez, H. Lopez, and L. Snchez. Ga-p based search of structures and parameters of dynamical process models. In *Advances in Soft Computing - Engineering, Design and Manufacturing*. 2003.

[12] E. H. Mamdani and S. Assiliang. An experiment in linguistic synthesis of fuzzy controllers. *International Journal of Man-Machine Studies*, 7:61–13, 1975.

[13] N. Minar, M. Gray, O. Roup, R. Krikorian, and P. Maes. Hive: Distributed agents for networking things. In *First International Symposium on Agent Systems and Applications*, Palm Springs, CA, USA, 1999.

[14] M. Mozer. The neural network house: an environment that adapts to its inhabitants. In *Proceedings of the AAAI Spring Symposium on Intelligent Environments*.

[15] 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.