

An evolutionary algorithm for the off-line data driven generation of controllers for intelligent building

Antonio López¹, Faiyaz Doctor², Luciano Sánchez³, Victor Callaghan², and
Hani Hagra²

¹ University of Oviedo. Electrical Engineering Department
Campus Viesques, Ed. 2, 33204, Gijón, Asturias, Spain.
e-mail:antonio@isa.uniovi.es

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

³ University of Oviedo. Computer Science Department
Campus Viesques, Ed. 1, 33202, Gijón, Asturias, Spain
e-mail:luciano@ccia.uniovi.es

Abstract. Ambient Intelligence (AiM) is nowadays an active research field. As a key matter of this concept, several approaches have been proposed for the development of learning techniques for the control of the devices in an intelligent building. In this paper, a GA-P is analyzed and discussed as a candidate algorithm for the design of such learning architectures. Some results obtained from experiments using real data gathered from an intelligent dormitory are shown. The GA-P seems to be a suitable method for the task, both due to its accuracy and for the easy and meaningful linguistic interpretation of the solutions it produces. A comparison with other learning method (Anfis) is also included.

1 Introduction

Ambient Intelligence (AiM) is nowadays an active research field. AmI deals with the development of a new paradigm where people are immersed in a digital environment 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 [1]: 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 AiM is how the system is capable of learning about the user behaviours, 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,

...) present special characteristics that could make necessary an specific addressing. In particular, learning systems for intelligent buildings present a unique and specific set of characteristics and our aim is to try and build these intelligent mechanisms to try and address those characteristics.

The goal in intelligent buildings is to control the operation of a set of devices replacing a human user. The system, in a non-intrusive fashion, observes user's interaction with the building and learns how to pre-empt them automatically. This learning is based in the information provided by a set of environmental sensors (temperature, humidity, ...) and by the state of the actuators (buttons, controllers, ...).

Different approaches with similar goals have been proposed in the state of the art. Partial applications can be found in [2], where a multi-agent system for the control of buildings is proposed, although this approach does not pay much attention to automatic learning; in [3], where artificial neural networks are used for the intelligent control of lighting in a building; in [4], where evolutive algorithms are used to develop HVAC (Heating, Ventilation and Air Conditioned) models with control purposes; etc. Nevertheless, more complete approaches can also be found. In [5] and [6] the design of multi-agent systems for the real time control of a building based on machine learning is proposed.

One of these approaches [6] propose an adaptive learning system based on fuzzy controllers for the real time control of the building. Using an initial set of quite simple fuzzy controllers, the adaptive learning system is capable of enhancing such controllers in order to reach a near to optimum control of the building.

The main motivation for the work described here has been to analyze some methods for the data driven automatic generation of fuzzy controllers for heterogeneous devices in an intelligent building. In this way, the initial controllers provided for adaptive learning systems like the one mentioned in [6] could be more accurate than those used at present and the expected learning time could be reduced.

Among several alternatives analyzed and discussed in a later section, a GA-P based technique has been selected. Let us say for the time being that this technique allows an easy linguistic interpretation of the solutions, permitting one to analyze in depth the rules proposed for the controller. This analysis could allow one to accept or reject the solution on the basis of more advanced criteria than a measure of the error between the response of the controller and the desired value for the actuator devices.

To validate the technique, several experiments have been performed using real data gathered from an intelligent dormitory described next. Results from the GA-P are analysed in depth and a comparison with a well-known technique from the state of the art namely the ANFIS system [7] is shown.

The rest of this paper is organised as follows: In section 2 we give an introduction 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 Automatic Generation of Fuzzy Controllers for Intelligent Buildings

The control of devices of an intelligent building proposed by approaches like [6] is based on fuzzy controllers. A classical simple output fuzzy controller [8] 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.

Several approaches can be found in the state of the art based on the automatic generation of fuzzy controllers from sampled data. Many of them are 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 optimisation of the set of parameters of a given rule set, although the problem of extracting rules from data has been also addressed [9]. Regarding NN, Anfis (a NN where the internal structures represent fuzzy relations) is one of the most popular methods [7]. 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 [10]. 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 optimisation [11]. GA-P algorithms can be also used for the 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 [12].

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 provided by the system could allow an interaction between it and a human operator, making it easier to obtain adequate controllers. 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 The GA-P Algorithm for the Generation of Fuzzy Controllers

GA-P algorithms [13] are hybrids between a genetic 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[14].

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. 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.

```

CONTROLADOR: if CONDITION then (y es  $k_{o1}$ )
              if CONDITION then (y es  $k_{o2}$ )
              ...
              if CONDITION then (y es  $k_{on_o}$ ) |
              if CONDITION then (y es 0.25)
              if CONDITION then (y es 0.75)

CONDITION: ASSERT $_{a_1}$  | ASSERT $_{a_2}$  | ... | ASSERT $_{a_m}$ 
           | ASSERT $_{d_1}$  | ASSERT $_{d_2}$  | ... | ASSERT $_{d_{m'}}$ 
           | CONDITION O CONDITION
           | CONDITION Y CONDITION

ASSERT $_{a_1}$ : trapezio_izquierdo( $a_1, K_{11}, K_{12}$ ) |
           triangle( $a_1, K_{11}, K_{12}, K_{13}$ ) |
           triangle( $a_1, K_{12}, K_{13}, K_{14}$ ) |
           trapezio_derecho( $a_1, K_{1_{n1-1}}, K_{1_{n1}}$ )
           ...
ASSERT $_{a_m}$ : trapezio_izquierdo( $a_m, K_{m1}, K_{m2}$ ) |
           triangle( $a_m, K_{m1}, K_{m2}, K_{m3}$ ) |
           triangle( $a_m, K_{m2}, K_{m3}, K_{m4}$ ) |
           trapezio_derecho( $a_m, K_{m_{nm-1}}, K_{m_{nm}}$ )

ASSERT $_{d_1}$ : zero( $d_1$ ) |
           one( $d_1$ )
           ...
ASSERT $_{d_{m'}}$ : zero( $d_{m'}$ ) |
             one( $d_{m'}$ )

PARTITION_CONSTANTS:  $K_{11} \dots K_{1_{n1}} \dots K_{m1} \dots K_{m_{nm}} K_{o1} \dots K_{on}$ 

```

Fig. 1. Fuzzy Controller Grammar.



Fig. 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

This section begins with a description of the intelligent dormitory used for our tests. Following this a detailed description of the structure of the tests performed with GA-P is given. In the analysis of the results the focus will be set at the deviation between the value proposed by the fuzzy controller and the real value of the actuator. Two controllers will be analysed in depth and in the last part, comparative results with ANFIS are shown.

4.1 The iDorm

The experimental tests described in this paper are based on real data taken from an intelligent dormitory (iDorm) at the University of Essex (see fig. 2). It is used for experiments in the field of intelligent buildings [16]. A set of sensors and actuators are available for the intelligent control of the devices of the idorm.

A student was using the iDorm in a normal fashion for several days. Seven sensors (External Light Level, Internal Light Level, External Temperature, Internal Temperature, Desk Chair Pressure, Bed Pressure and Time) and 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) were sampled to get data for the experiments.

4.2 GA-P Test Estructure

The GA-P was applied using a population of 200 individuals that evolves for 200 generations. 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. Aptitude is calculated by the Root Mean Squared Error (RMSE).

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

A last consideration is 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).

4.3 Summarized Analysis

Fig. 3 shows the averaged results over the six sample configurations for the ten outputs. The X axes 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.

Regarding the accuracy of the controllers, it can be seen also how some outputs seem to be more predictable than others. The reason could be that some of them were used during the sample process in a more deterministic fashion than others. Another reason could be that there is not enough information in the samples that allow them to predict their behaviour with a greater success.

Analysing in more depth the results for one analog output (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. For binary outputs the percentage of hits is always about the 100%.

4.4 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 (in the following, we will label this partitions as

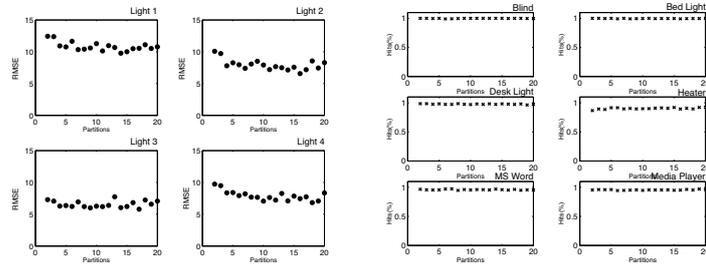


Fig. 3. Effect of the number of partitions in the search results. Averaged results over the 6 sample configurations.

LOW, MEDIUM and HIGH). Although more accurate controllers can be found in the bank of experiments, the use of this low number of partitions favours the interpretability of solutions.

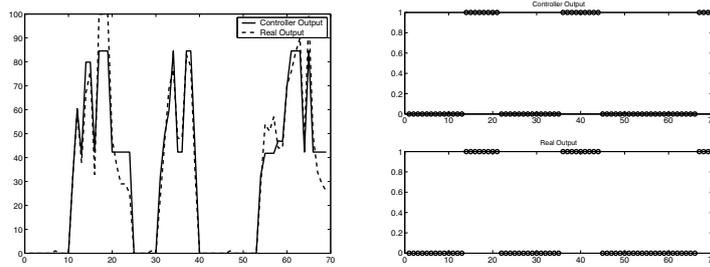


Fig. 4. Comparison of the fuzzy controller and real outputs for light 3 (left) and blind (right).

Fig.4 shows a comparison between the output of the selected fuzzy controllers and the real output vales 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. In both controllers we can see the existence of variables whose influence could be predicted in advance. Such is the case of external and internal light levels, and bed pressure sensor for the control of light 3, and external light level and bed pressure for the control of the blind state. Nevertheless, in the controller for light 3 a strange variable

appears: external temperature. The reason can be that the controller needs to make divisions with a 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 how 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. This could indicate that the user 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.

if (BedPressure is HIGH) and (InternalLightLevel is LOW) and (ExternalLightLevel is LOW) and (ExternalTemperature is LOW) then (Light3 is LOW)	if (BedPressure is HIGH) then (BlindState is LOW)
if (ExternalLightLevel is MEDIUM) then (Light3 is MEDIUM)	if (ExternalLightLevel is LOW) then (BlindState is HIGH)
if (ExternalLightLevel is LOW) then (Light3 is HIGH)	

Fig. 5. Controllers for Light 3 (left) and Blind (right)

4.5 ANFIS Comparison

A 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 general much faster than the GA-P.

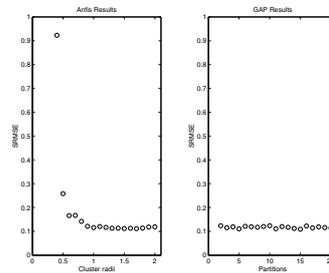


Fig. 6. ANFIS vs. GA-P. SRMSE stands for Scaled Root Mean Square Error

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 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. The technique has been compared with ANFIS. 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 first is to try to introduce reinforcement learning mechanisms in the scheme. The GA-P described in this paper is only valid for the generation of initial controllers, being it necessary to use another mechanism to adapt them in a long-life learning system. Adding reinforcement mechanisms, the GA-P could also be used for such online adaptation. A second extension could be the introduction of dynamics in the controllers. The controllers "learned" by the GA-P in the way described in this paper are only valid for the generation of static mappings from sensor states to actuator states. But we think that dynamic behaviors are present in the management of building devices by a user, and so the controllers need to present such behaviors for a successful control.

6 Acknowledgements

The authors would like to thank the Intelligent Inhabited Environments Group at 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. Ahola, J.: Ambient intelligence. ERCIM News Online Edition (2001)
2. Boman, M., Davidsson, P., Skarmas, N., Clark, K.: Energy saving and added customer value in intelligent buildings. In Nwana, H.S., Ndumu, D.T., eds.: Proceedings of the 3rd International Conference on the Practical Applications of Agents and Multi-Agent Systems (PAAM-98), London, UK (1998) 505–516
3. Mozer, M.: The neural network house: an environment that adapts to its inhabitants. (In: Proceedings of the AAAI Spring Symposium on Intelligent Environments)
4. Angelov, P., Buswell, R.: Automatic generation of fuzzy rule-based models from data by genetic algorithms. *Information Sciences Informatics and Computer Science: An International Journal* **150** (2003) 17–31
5. Minar, N., Gray, M., Roup, O., Krikorian, R., Maes, P.: 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)
6. Hagra, H., Callaghan, V., Clarke, G., Colley, M., Pounds-Cornish, A., Holmes, A., Duman, H.: Incremental synchronous learning for embedded-agents operating in ubiquitous computing environments. In Loia, V., ed.: *Soft Computing Agents. A New Perspective for Dynamic Information Systems*. (2002)
7. Jang, J.S.R.: ANFIS: Adaptive-Network-Based Fuzzy Inference System. *IEEE Transactions on Systems, Man, and Cybernetics* **23** (1993) 665–684
8. Mamdani, E.H., Assiliang, S.: An experiment in linguistic synthesis of fuzzy controllers. *International Journal of Man-Machine Studies* **7** (1975) 61–13
9. Herrera, F., Magdalena, L.: (Genetic fuzzy systems: A tutorial)
10. Alba, E., Cotta, C., Troyo, J.J.: Type-constrained genetic programming for rule-base definition in fuzzy logic controllers. In Koza, J.R., Goldberg, D.E., Fogel, D.B., Riolo, R.L., eds.: *Genetic Programming 1996: Proceedings of the First Annual Conference, Stanford University, CA, USA, MIT Press* (1996) 255–260
11. Kasabov, N.: Evolving fuzzy neural network for supervised/unsupervised on-line, knowledge-based learning (2001)
12. Garcia, S., Gonzalez, F., Sanchez, L.: Evolving fuzzy rule based classifiers with GA-P: A grammatical approach. In Poli, R., Nordin, P., Langdon, W.B., Fogarty, T.C., eds.: *Genetic Programming, Proceedings of EuroGP'99*. Volume 1598 of LNCS., Goteborg, Sweden, Springer-Verlag (1999) 203–210
13. Howard, L.M., D'Angelo, D.J.: The GA-P: A genetic algorithm and genetic programming hybrid. *IEEE Expert* **10** (1995) 11–15
14. Lopez, A., Lopez, H., Sánchez, L.: Ga-p based search of structures and parameters of dynamical process models. In: *Advances in Soft Computing - Engineering, Design and Manufacturing*. (2003)
15. Sanchez, L., Couso, I., Corrales, J.A.: Combining operators with sa search to evolve fuzzy rule based classifiers. *Information Sciences* **136** (2001) 175–191
16. Holmes, A., Duman, H., Pounds-Cornish, A.: The idorm: Gateway to heterogeneous networking environments. Technical report, Intelligent Inhabited Environments Group, University of Essex (1998)