Neurocomputing 75 (2012) 219–225

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Multi-objective learning of white box models with low quality data ☆

José R. Villar [a,*], Alba Berzosa [b], Enrique de la Cal [a], Javier Sedano [b], Marco García-Tamargo [a]

[a] University of Oviedo, Campus de Viesques s/n 33204 Gijón, Spain
[b] Instituto Tecnológico de Castilla y León (ITCL), Polígono Industrial Villalonquéjar. c/López Bravo, 70. 09001. Burgos, Spain

ARTICLE INFO

Available online 5 August 2011

Keywords:
Genetic programming
Genetic algorithm and programming
Low quality data
Multi-objective simulated annealing
Energy efficiency

ABSTRACT

Improving energy efficiency in buildings represents one of the main challenges faced by engineers. In fields like lighting control systems, the effect of low quality sensors compromises the control strategy and the emergence of new technologies also degrades the data quality introducing linguistic values. This research analyzes the aforementioned problem and shows that, in the field of lighting control systems, the uncertainty in the measurements gathered from sensors should be considered in the design of control loops. To cope with this kind of problems Hybrid Intelligent methods will be used. Moreover, a method for learning equation-based white box models with this low quality data is proposed. The equation-based models include a representation of the uncertainty inherited in the data. Two different evolutive algorithms are use for learning the models: the well-known *NSGA-II* genetic algorithm and a multi-objective simulated annealing algorithm hybridized with genetic operators. The performance of both algorithms is found valid to evolve this learning process. This novel approach is evaluated with synthetic problems.

1. Introduction

Improving energy efficiency represents a big challenge in modern engineering [1–3], and more specifically, in the field of lighting control systems included in building automation. In a lighting control system, the typical lighting control loop includes a light sensor, the light ballasts and a light controller. The light sensor measures the amount of light in a room, although the measurements lack hysteresis and saturation [2]. Moreover, recent studies show that the measurements obtained from light sensors are highly dependent on the light sensor unit [4]. This meta information in the data gathered from processes is rarely used, and it is mainly related to non-stochastic noise.

In our opinion, learning models using the meta information in the data will result in more robust models and better control design. Hybrid Intelligent methods [5,6] will be used to cope with this kind of problems. This study shows the presence of such meta information and presents a novel method for learning with this kind of data. The method has been developed for learning equation-based models (hereinafter EB models) but can be easily extended to different models, including neural networks. To evolve the models, two different evolutionary learning strategies have been used: the well-known *NSGA-II* genetic algorithm [7] and the Multi-Objective simulated annealing hybridized with genetic operators [8] (hereinafter MOSA).

The remainder of this manuscript is as follows. Firstly, the problem description and the uncertainties in real-world problems such as the simulation of lighting control systems are presented. A review of the literature concerned with learning models with low quality data (hereinafter LQD) is then shown. In Section 4 the novel method is described. Section 5 deals with the experimentation and results obtained with the proposal. Finally, some conclusions and future work are outlined.

2. Low quality data in real-world processes

The aim of lighting control systems is to control the electrical power consumption of the ballasts in the installation so that the luminance complies with the regulations [9]. In these systems, the luminance is measured through light sensors and variables such as the presence of inhabitants are analyzed, as well. However, the relevance of the former is higher as it is used as the feedback in the lighting control loop.

Nevertheless, the output of such sensors is highly dependent on a number of factors: the amount of sunlight, the fact that measurements vary from one sensor to another or that the repeatability could be compromised, among others. The values

☆ This research has been funded by the Spanish Ministry of Science and Innovation, under project TIN2008-06681-C06-04, and by the ITCL project CONSOCO.

* Corresponding author.
E-mail addresses: villarjose@uniovi.es (J.R. Villar), alba.berzosa@itcl.es (A. Berzosa), delacal@uniovi.es (E. de la Cal), javier.sedano@itcl.es (J. Sedano), marco@uniovi.es (M. García-Tamargo).

0925-2312/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2011.02.025

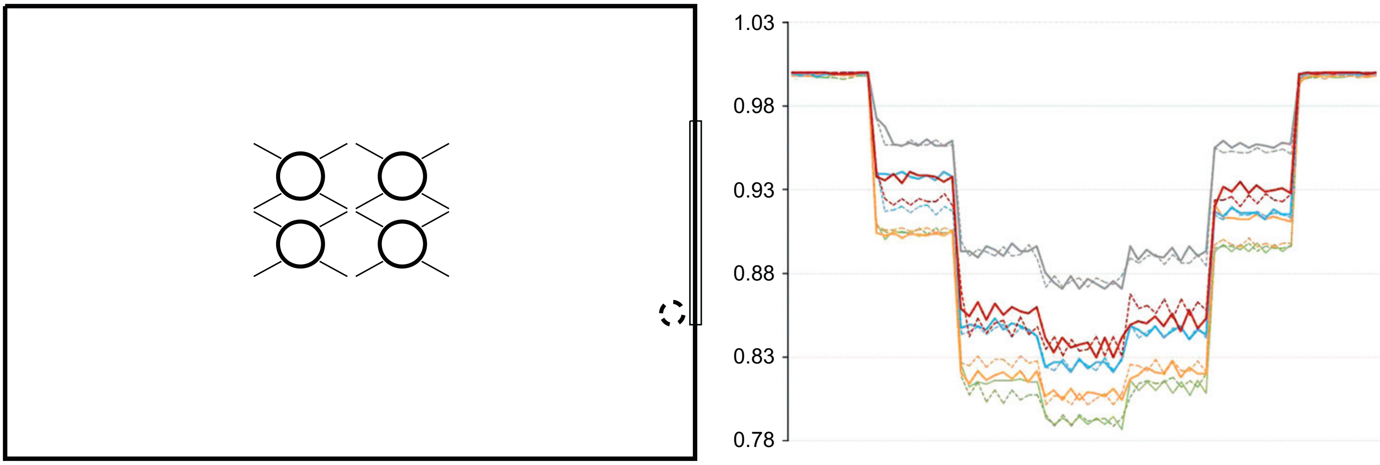


Fig. 1. On the left panel, the lighting system to simulate is shown. Different places for the light sensor could be proposed; the light measured will differ from one case to another. On the right panel, the results obtained in a previous study are depicted and the normalized output of the different light sensors is drawn. The up/down cycle was repeated twice for each one, so the dashed lines correspond to the second run. Ten samples were considered for each step. No transient was considered, the sample rate was 1 sample/second.

gathered from this kind of source are denoted as LQD, that is, data with low degree of accuracy. Consequently, the output of the sensors is usually filtered and then used as the feedback of the control loop, but always as a crisp value.

Lighting system simulations to set and tune PID controllers for lighting control systems use models to estimate the response of the light sensors. These simulations, which have been widely studied [2], need to simulate the light sensors in a room when a certain amount of electrical power is applied for lighting. In the studied literature, when obtaining models for simulation, only crisp values are regarded as the measurements obtained from light sensors. Obviously, the inputs and outputs of the light sensor models obtained are also crisp variables. In the study presented in [4], the behavior of the light sensors measurements were analyzed (see Fig. 1). This experiment analyzed the step response of the light sensor unit in a room with a light controller. Up to four power levels were applied and the measurements obtained from the light sensor units were registered. For this experiment, the available blind was closed, the step response was recorded twice for each unit, and up to five different sensor units were used. As can be seen, the measurements obtained are highly dependent on the sensor itself, but also the hysteresis behavior and the lack of repeatability can be perceived. A controller designed without considering these issues will fail in reaching the fixed set point and, as a result, it will also fail in minimizing the electrical energy consumption when the sensor differs from the assumed mean behavior. Therefore, light controllers cannot be optimum if they are obtained from crisp sensors which are affected with uncertainty. In general cases, controllers must be valid for any available light sensor unit, regardless of their behavior. Consequently, the simulation models of the sensors need to be learned considering the meta-information within the data. It is expected that the energy efficiency in lighting control systems can be improved if LQD is analyzed. More specifically, the methodology from classical control theory proposed by different authors (e.g. [2]) should make use of the meta-information aware models for designing the controllers.

3. Algorithms managing low quality data

The need for algorithms capable of facing LQD is a well-known fact in the literature. As analyzed in [10], several studies have presented the decrease in the performance of crisp algorithms as uncertainty in data increases.

On the other hand, [11] analyzes the complex nature of the data sets in order to choose the best Fuzzy Rule Based System. Several measures are proposed to deal with the complexity of the data sets and the Ishibuchi fuzzy hybrid genetic machine learning method is used to test the validity of these measures. This research also concludes that there is a need to extend the proposed measures to deal with LQD.

By LQD, we refer to the data sampled in the presence of non-stochastic noise or obtained from imprecise sensors. It is worth noting that the main part of the information acquired with sensors and industrial instrumentation can be regarded as LQD. In our opinion, one of the most successful researches in soft computing dealing with LQD is detailed in [12]. These studies show the mathematical basis for learning uncertainty aware genetic fuzzy systems – both classifiers and models. The LQD is assumed as fuzzy data, where each α-cut represents an interval value for each data.

Finally, it is worth pointing out that the fitness functions to train classifiers and models with LQD are also fuzzy valued functions. Hence the learning algorithms should be adapted to such fitness functions [13]. The ideas and principles previously shown have been used in several applications, with both realistic and real-world data sets [14–16]. In the next section, some of these ideas are used in learning EB white box models where imprecise data, such as the data gathered from the light sensors, is available.

4. Learning models with low quality data

As stated in previous sections, data gathered from light sensors is imprecise, behaves with hysteresis and lacks repeatability. Obviously, we can represent this kind of data as non-crisp granules of information (i.e., an interval or a probable radius, etc.), but this approach would introduce higher computational costs and complexity in the model learning process.

Conversely, we will assume that the data set used for learning the models contains imprecise crisp values only. Besides, we will consider the variables responsible for representing the uncertainty inherent in the data. Let us assume that white box EB models are to be learned from LQD; these models contain the equation that relates the output variable with the input features (see Fig. 2).

Genetic programming (GP) is typically used in problems where the learning of equations is dealt with [17]. While GP evolves

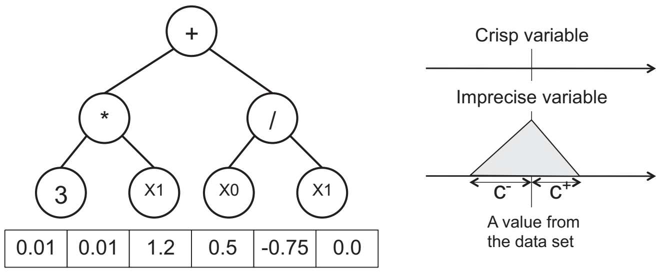


Fig. 2. The representation of an EB model. The tree on the left corresponds with $y = 0.5{*}x_1 + x_0/x_1$. Each model contains a tree of nodes for storing the structure of the equation and a vector of numerical constants. On the right, the evaluation of a crisp variable and of an imprecise variable is depicted. When evaluating a crisp variable, the corresponding value from the data set is returned. When evaluating an imprecise variable, a parameterized fuzzy number is returned.

equations that are self-contained (that is, the equation contains the values of the constants that eventually appear in the formulae) these are usually avoided to reduce the diversity and the time consumption in finding good solutions. Instead of including the constants in the equation, GP is generally hybridized with genetic algorithm (GA): the GP learning for evolving the structure of a model and the GA for evolving its numeric constants [8]. In the latter case, when a constant is to be used in an equation, the node that represents the constant links to a valid position in the vector of constants of the individual.

Learning EB models with LQD using GP hybridized with GA (hereinafter, GAP) has been barely studied [18,19]. Obviously, if we want the vagueness in this meta-information in the data to be learned then EB models should be extended with a representation of the uncertainty while the evolutionary algorithm should manage to learn the uncertainty from the available LQD.

This study proposes a solution to learn uncertainty aware EB models with LQD. The following subsections present the description of a solution. Firstly, the representation of the uncertainty is detailed and, secondly, how the way the uncertainty management influences the learning algorithm is shown. Section 4.2 describes the EB individuals, while Section 4.3 explains the fitness function calculation. The genetic operators used to evolve the EB models are detailed in Section 4.4. Finally, Sections 4.5 and 4.6 outline the proposed learning algorithms: the multi-objective simulated annealing algorithm and the NSGA-II, respectively.

4.1. Representation of vagueness in a GP model

As far as we are learning white box models, we can represent information related with vagueness in the models. According to [12], LQD can be represented with fuzzy numbers. Let us assume we have a priori knowledge of the features of the data set that are considered imprecise. Then, we can assign constants to each imprecise variable to bind its vagueness, as presented in Fig. 2.

For each imprecise variable we assign two constants C^- and C^+, which are to be evolved in the learning process. These constants represent the limits of a triangular membership function for a α-cut $= 1$ which is associated with each imprecise variable. On the left panel of Fig. 2 an imprecise variable is presented with $C^- = 0.01$ and $C^+ = 0.01$, both at the beginning of the constants vector.

Let us suppose that $X1$ is the imprecise variable and that we are to evaluate the model depicted in the mentioned figure. If the training data set is the set $\{d_i^j\}$, with $i = \{0, \ldots, (D-1)\}$ for each input variable (where D stands for the dimension of the input space, $D=2$ in the example in Fig. 2), and $j = \{1, \ldots, N\}$ (with N the number of examples in the data set) then, whenever $X1$ is evaluated for the example j, a fuzzy number with a triangular fuzzy membership defined through the three following values $[d_1^j - C^-, d_1^j, d_1^j + C^+]$ is returned.

It is interesting to mention that if symmetrical membership functions are adopted, then only one constant per imprecise variable is needed. Consequently, for each imprecise variable in the feature space, one or two constants are reserved and positioned as the former constants in the constants vector. Obviously, vagueness can be extended also to the numerical constants of the models although for this study only uncertainty in variables is analyzed.

If imprecise variables are defined a priori and the above detailed representation is used, the evaluation of the nodes in the tree will produce fuzzy numbers, and the output of the model should be calculated using both fuzzy numbers and crisp data. As in classical fuzzy literature, crisp values are extended to fuzzy singletons, so only operations with fuzzy numbers are required. In order to reduce the computational cost, the solution presented in [16] is used, and evaluations are calculated only for certain predefined α-cuts.

As a result, we are to learn EB models using LQD, and the representation of vagueness detailed in this section applies to this. Besides, two major consequences in learning EB models arise. Firstly, the representation of uncertainty is based on the introduction of constants in the constant vector. Thus the number of constants is greater, and greater is the time needed for the convergence of the algorithm. Secondly, as it is explained in the following subsections, the evaluation function of models with imprecise variables is not crisp but fuzzy; therefore, fuzzy operations are to be used instead of classical operations. Both consequences highly increase the computational costs and time needed to converge. In order to reduce the computational costs of calculating with fuzzy numbers simulated annealing is proposed as the evolutionary strategy [8].

4.2. Representation of an individual

An individual in this study is compounded of the equation representation, the constants vector and the specification of the uncertainty. Readers must recall the assumption that there is a priori knowledge of which variables behave with uncertainty. The specification of uncertainty is provided with, firstly, the number of constants used to represent the uncertainty and, secondly, with the specification of the imprecise variables that must deal with LQD.

The number of constants to represent the uncertainty in each variable could be 0 if no imprecise variable is assumed, 1 if symmetrical triangular membership functions are used or 2 if asymmetrical triangular membership functions are assumed. The specification of which variables behave with uncertainty is a parameter that contains the list of the indexes of the imprecise input variables. If at least one input variable is to deal with LQD, then the output of the models is fuzzy, as well. Obviously, once this parameter is given, the number of imprecise variables is known. The uncertainty specification remains unchanged during an experiment and it is the same for all the individuals.

As in GAP models, the equation representation consists of a nodes tree, each internal node corresponds with a valid operator, and the leaf nodes correspond with a variable index or a constant index. The number of constants is predefined, so the constant vector in all individuals has the same size, although the constants may not be used in one individual, readers can just imagine the equation *output*$=X1$. The first group of constants in the constant vector is assigned to the uncertainty management; in other words, let n be the number of imprecise variables and nc the number of constants to represent the uncertainty, then the first $n{*}nc$ constants are reserved for uncertainty management and the remainder are free to be indexed in the equations. The position of the uncertainty constants in the vector is related with the

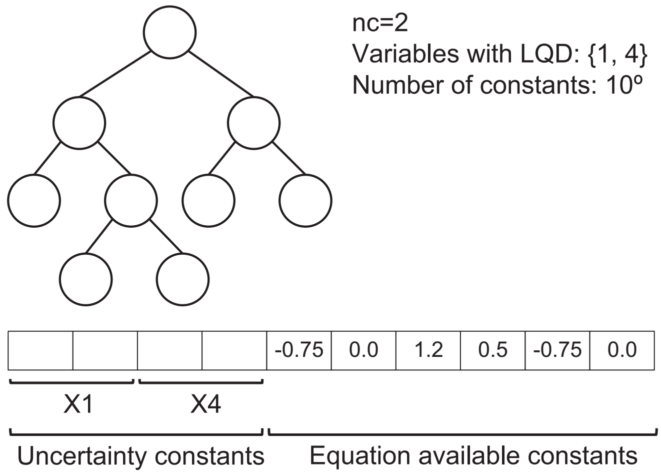


Fig. 3. The representation of an individual: the nodes tree to represent the equation, the uncertainty specification and the constants vector. The latter is divided into the uncertainty constants and the constants that remain in the equation.

position of the corresponding variable in the data set. Fig. 3 shows the representation of an individual.

Although better approaches have been described in the literature, for the sake of simplicity, individuals are generated randomly. Finally, an upper bound to limit the span of the uncertainty constants is predefined as a percentage of the range of values of the variable in the data set. The uncertainty values are randomly fixed within the bound limit.

4.3. *The fitness function*

Evaluating a variable when LQD is given generates a fuzzy number. Subsequent nodes, that is, operator nodes in the tree of nodes representing the equation, need to operate with both crisp and fuzzy numbers. Crisp values are converted to fuzzy singleton numbers. Therefore, the output of an individual when an example from the data set is given is a fuzzy number. As stated in [13], extended measures of the error should be used in order to evaluate the effectiveness of a model and to avoid the use of multi-objective algorithm in single objective problems. In this study, once again for the sake of simplicity, we propose the evaluation of the models for a given α-cut [16]. Therefore, the output of a model is an interval and we calculate the mean square error (MSE) using Eq. (1) with interval arithmetic and bounding the MSE to positive values. Further studies should consider the use of more elaborated MSE measures:

$$\mathrm{MSE} = \frac{\sqrt{\sum (x-\hat{x})^2}}{sqrtn} \tag{1}$$

The MSE measure given by (1) when dealing LQD generates interval values, thus only the individuals with null intersection would be ordered. To cope with individuals with similar MSE we propose the use of the non-specificity of the error as a second fitness function, the lower the non-specificity the better the model. Finally, we proposed the use of a third fitness function measuring the number of desired values covered by the fuzzy output [16], the higher the number of desired values covered by the model, the better it is.

4.4. *Genetic operators*

Evolving EB individuals involves using four genetic operators: two from GP evolution (the GP crossover and mutation) and two from GA (GA crossover and mutation). The GP operators introduce variability in the structure of the model, that is, the equation itself. The GA operators modify the vector of constants. In all the cases, there is a predefined probability of carrying out each of these genetic operations.

The GP operators are involved in searching the structural space by interchanging nodes in the tree. The crossover is defined as follows: two parents are chosen to be crossed using binary tournament; then, for each one an index is randomly generated (the index is in the range from 1 to the number of nodes in the tree); finally, the nodes at the index positions are interchanged. The GP crossover, as usually implemented in GAP, does not interchange the equation constants. It must be remembered that a constant used in an equation is only referred to by its position in the constants vector, so the index, not the constant, is interchanged. As a result, the GP crossover generates two new individuals for which the validity should be checked.

The GP mutation operator makes use of the following semantic grammar to mutate an operator, provided that mutation only varies from one operator type to another operator type of the same arity. A node indexing an equation constant mutates varying its index among the valid indexes for equation constants in the constants vector of the individual. Finally, a node indexing a variable mutates varying its index among the valid indexes of the variables:

$$\mathrm{EXP} \mapsto \mathrm{MonoOperand} \mid \mathrm{BiOperand} \mid \mathrm{Constant} \mid \mathrm{Variable}$$

$$\mathrm{MonoOperand} \mapsto (\sin \mathrm{EXP}) \mid (\cos \mathrm{EXP}) \mid (\textbf{delay}\ \mathrm{EXP})$$

$$\begin{aligned}\mathrm{BiOperand} \mapsto & (+\mathrm{EXP}\ \mathrm{EXP}) \mid (-\mathrm{EXP}\ \mathrm{EXP}) \mid (*\mathrm{EXP}\ \mathrm{EXP}) \\ & \mid (/\mathrm{EXP}\ \mathrm{EXP}) \mid (\min \mathrm{EXP}\ \mathrm{EXP}) \mid (\max \mathrm{EXP}\ \mathrm{EXP})\end{aligned}$$

The GA crossover is a classical two point crossover (the constants vector is then divided in three parts: the initial, the central and the ending parts) that interchanges the constants vector of both individuals. The first offspring contains a constants vector including the first and the ending parts of the first parent and the central part of the second parent and vice versa for the second offspring: its constants vector includes the first and the ending parts of the second parent and the central part of the first one.

On the other hand, for each constant in the constants vector of an individual, the GA mutation operator evaluates whether to mutate or not, according to a predefined probability, and if so the constant is assigned with a random value in the also predefined range of the constant values.

4.5. *Simulated annealing and the multi-objective approach*

Simulated annealing is an optimization technique with a very small computational cost. It has been shown as a good meta-heuristic technique to evolve the model learning when multi-objective problems arise [8,16,20]. In this study, the MOSA presented in [8] is used as the evolutionary strategy. The algorithm makes use of Pareto dominance operators to establish the partial order relations. The only variation is that in each run, that is, in each generation, the type of operation to carry out is chosen. So in each run only GP or GA operations can take place, but never both in the same run. The distance measure between individuals is the edition distance [17,19] normalized with the sum of the number of nodes of the two compared trees.

4.6. *NSGA-II adaptation to LQD learning*

The non-dominated sorting genetic algorithm [7] is a well-known multi-objective genetic algorithm that has been widely used, i.e., in [16]. This algorithm sorts the population in different surfaces according to the Pareto dominance operator, but also using the so-called crowding distance. This latter measure reflects the density of the population in each individual. As this measure is detailed for crisp data, this study should use a natural extension to learn the EB models with LQD. This simple extension makes use of the interval arithmetic, sorts the population using the interval relations of order and binds each of the accumulative operations greater than 0.

Table 1
Formulas for the data sets generation.

$f_1 = x_1 + x_0 * (x_2 - 0.5)$	$t = \{1, \ldots, 100\}$
$f_2 = 2 * x_1 * x_2$	$x_0 = abs(\cos(t))$
$f_3 = \cos(x_0) * (x_2 - x_1)$	$x_1 = abs(\sin(t))$
$f_4 = 2 * x_2 * delay(f_4)$	$x_2 = abs(\cos(t) * \sin(t))$
$f_5 = \cos(x_0) * (delay(f_5) * (x_2 + x1)) + 0.1$	$x_3 =$ random in the range [0, 1]

5. Experiments and results

To test our proposal five different synthetic problems are proposed. The formulas for generating the data sets are presented in Table 1. The input variables $\{x_0, \ldots, x_3\}$ evolve with the time. The formulates $\{f_1 f_2 f_3\}$ are intended to deal with regression problems, while $\{f_4, f_5\}$ represent time series problems. For each problem two data sets of 100 examples each are generated: the precise and the imprecise data sets. The former is obtained directly from the equations, while the latter is obtained by adding a random value in the range $[-1\%SPAN, 2*1\%SPAN]$, where 1%SPAN is the 1% of span of the corresponding variable. Specifically, variable x_1 is imprecise for the three regression problems, while x_2 is imprecise for the time series problem. Both MOSA and NSGA-II are analyzed with the two families of data sets (the precise and the imprecise data sets). The two opposite cases are compared when learning EB

Table 2
Experimentation results mean values. MSE value of the mean central point of the MSE function for the individuals of lowest MSE value individual in the populations (MSE) and the closest to the origin individual in the populations, considering the 10 runs of each experiment. All values should be multiplied by 10^{-3}. Capital letters M or N refer to MOSA and NSGA-II; capital letters P and I refer to whether the model use precise variables (P) or imprecise variables (I).

		Precise data set				Imprecise data set			
Problem		M-P	M-I	N-P	N-I	M-P	M-I	N-P	N-I
f_1	MSE	0.1156	0.1941	0.2101	0.2142	0.1421	0.2468	0.2333	0.2358
	Closest	0.1	1.5	0.2	0.4	0.1	0.9	0.2	0.5
f_2	MSE	0.0042	0.0519	0.1319	0.0091	0.0086	0.0387	0.1269	0.0972
	Closest	0.2	0.4	0.1	0.0	0.0	0.7	0.1	0.2
f_3	MSE	0.0008	0.0057	0.0852	0.0038	0.0035	0.0064	0.1004	0.1014
	Closest	0.0	0.7	0.2	0.0	0.0	1.6	0.1	0.5
f_4	MSE	0.4718	0.6301	0.6744	0.5061	0.5188	0.6599	0.6584	0.4996
	Closest	1.4	3.9	1.8	2.0	1.2	2.8	1.3	2.9
f_5	MSE	0.0476	0.0747	0.0751	0.0639	0.0622	0.0747	0.0737	0.0768
	Closest	0.1	2.6	0.1	0.5	0.1	3.2	0.1	1.0

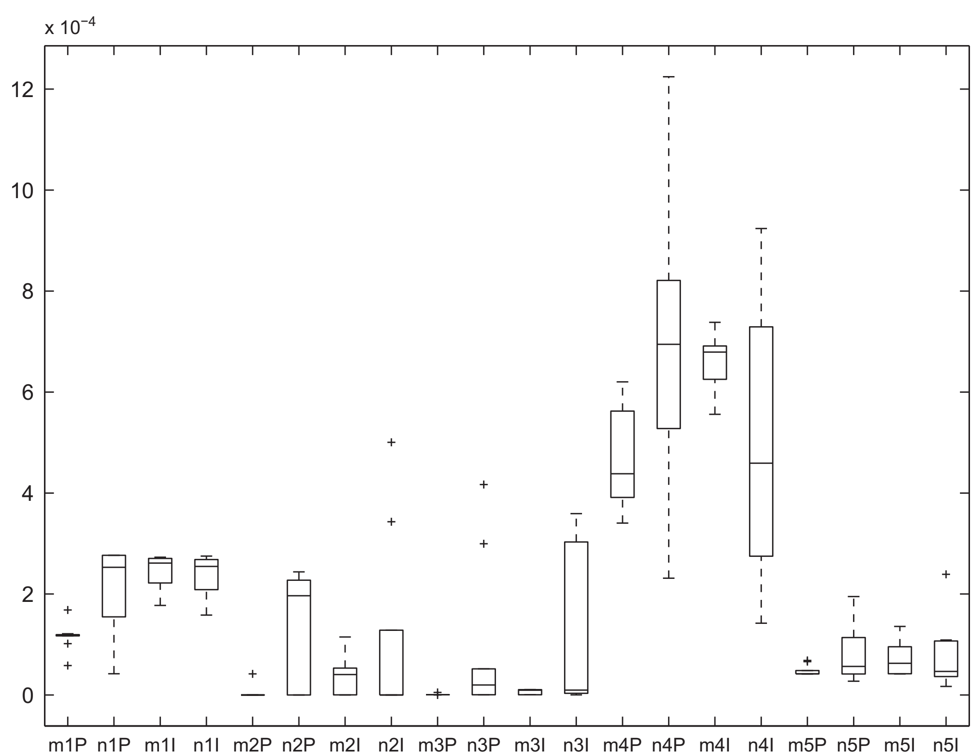


Fig. 4. Boxplot of the central points of the MSE for the best individuals considering only the MSE function. Each box is identified using m for MOSA or n for NSGA-II, a number for the data set identification and a final letter to state whether the precise (P) or the imprecise (I) data set is used.

models without imprecise variables and considering all the input variables as imprecise. Consequently, for each problem eight different experiments have been designed and 10 runs of each experiment have been carried out for statistics purposes. In all the runs, the number of model evaluations have been fixed to 50 000 and kept the same for all the experiments.

The experiments use an α-cut 0.95, a population size of 50 individuals, 10 000 iterations for the NSGA-II (and a much bigger number of iterations for the MOSA). The mutation probabilities are 0.25 while crossovers are always carried out. The maximum number of nodes is set to 10, while the maximum depth is 5. Only one constant per imprecise variable is used, and the number of constants is set to 7. The maximum bias of imprecise variables is set to 0.01%. MOSA uses $\Delta=0.1$, $T_0=1$ and $T_1=0$. Results are shown in Table 2 and in the boxplot of Fig. 4. In the former, the eight different possibilities are compared for the best individual error found and for the individual closest to the origin. As seen, the use of the uncertainty representation does not punish the learning of the models, and the error measure keeps similar to that of the experiments without it. Although some differences have been found, both the MOSA and the NSGA-II are valid to learn the models. Finally, the difference between the results from the precise experiments and those with uncertainty in data and in models are not relevant. Thus, the uncertainty representation does not penalize the model learning.

6. Conclusions

This study proposes learning EB models to deal with LQD. The EB models include a representation of the uncertainty and evaluating a model generates a fuzzy value. The learning of models is defined as a multi-objective problem using Fuzzy fitness functions and two evolutionary learning strategies are proposed. The proposal has been analyzed with synthetic problems. The results show that the uncertainty representation in the EB models learned with the two evolutionary heuristic techniques keeps the same performance index even though LQD is given, although NSGA-II performs with a higher dispersion in the fitness space. Learning EB models with the representation of the uncertainty seems to be valid for modeling real-world LQD sensors. Thus these models could be used in obtaining more robust controllers. Finally, the mentioned extended error measures should be implemented to avoid the use of multi-objective algorithm in single objective problems.

References

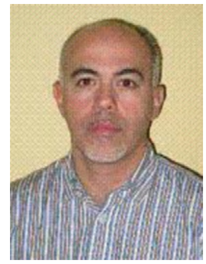
[1] D.H. Li, K. Cheung, S. Wong, T.N. Lam, An analysis of energy-efficient light fittings and lighting controls, Applied Energy 87 (2010) 558–567.
[2] R.D. Keyser, C. Ionescu, Modelling and simulation of a lighting control system, Simulation Modelling Practice and Theory 18 (2010) 165–176.
[3] J.R. Villar, E. de la Cal, J. Sedano, A fuzzy logic based efficient energy saving approach for domestic heating systems, Integrated Computer-Aided Engineering 16 (2009) 151–163.
[4] J.R. Villar, E. de la Cal, J. Sedano, M. García-Tamargo, Analysing the low quality of the data in lighting control systems, in: M. Gra na Romay, E. Corchado, M. García Sebastián (Eds.), Hybrid Artificial Intelligence Systems of Lecture Notes in Computer Science, vol. 6076, Springer, Berlin, Heidelberg, 2010, pp. 421–428.
[5] A. Abraham, E. Corchado, J.M. Corchado, Hybrid learning machines, Neurocomputing 72 (2009) 2729–2730.
[6] E. Corchado, A. Abraham, A.P.L.F. de Carvalho, Hybrid intelligent algorithms and applications, Information Science 180 (2010) 2633–2634.
[7] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast elitist multi-objective genetic algorithm: Nsga-ii, IEEE Transactions on Evolutionary Computation 6 (2000) 182–197.
[8] L. Luciano Sánchez, J.R. Villar, Obtaining transparent models of chaotic systems with multi-objective simulated annealing algorithms, Information Science 178 (2008) 952–970.
[9] I.N. de Seguridad e Higiene en el Trabajo, Real Decreto 486/1997, de 14 de Abril, sobre disposiciones mínimas de Seguridad y Salud en los Lugares de Trabajo (BOE del 23 de abril de 1997) Modificada por Real Decreto 2177/2004, de 12 de noviembre, Spanish Labour Ministry, 2004.
[10] A. Folleco, T.M. Khoshgoftaar, J.V. Hulse, A. Napolitano, Identifying learners robust to low quality data, Informatica (Slovenia) 33 (2009) 245–259.
[11] J. Luengo, F. Herrera, Domains of competence of fuzzy rule based classification systems with data complexity measures: a case of study using a fuzzy hybrid genetic based machine learning method, Fuzzy Sets and Systems 161 (2010) 3–19 Special section: New Trends on Pattern Recognition with Fuzzy Models.
[12] L. Sánchez, I. Couso, Advocating the use of imprecisely observed data in genetic fuzzy systems, IEEE Transactions on Fuzzy Systems 15 (2007) 551–562.
[13] L. Sánchez, I. Couso, J. Casillas, Genetic learning of fuzzy rules based on low quality data, Fuzzy Sets and Systems 160 (2009) 2524–2552 Theme: Learning.
[14] L. Sánchez, J. Otero, Learning fuzzy linguistic models from low quality data by genetic algorithms, in: IEEE Internacional Conference on Fuzzy Systems FUZZ-IEEE, 2007, pp. 1–6.
[15] L. Sánchez, M.R. Suárez, J.R. Villar, I. Couso, Mutual information-based feature selection and partition design in fuzzy rule-based classifiers from vague data, International Journal of Approximate Reasoning 49 (2008) 607–622.
[16] J.R. Villar, A. Otero, J. Otero, L. Sánchez, Taximeter verification using imprecise data from gps, Engineering Applications of Artificial Intelligence 22 (2009) 250–260.
[17] J.R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, Cambridge, MA, USA, 1992.
[18] L. Sánchez, Interval-valued ga-p algorithms, IEEE Transactions on Evolutionary Computation 4 (2000) 64–72.
[19] L. Sánchez, I. Couso, J.A. Corrales, Combining gp operators with sa seach to evolve fuzzy rule classifiers, Information Sciences 136 (2001) 175–192.
[20] H.K. Singh, T. Ray, W. Smith, C-psa: constrained pareto simulated annealing for constrained multi-objective optimization, Information Science 180 (2010) 2499–2513.

José R. Villar obtained his Engineering degree at the University of Oviedo (1992) and his PhD at the University of León (2002). He had worked for several engineering companies between 1992 and 1998. In 1998 he became an Assistant Professor with the Electric and Electronic Department at the University of León. In 2004 he became an Assistant Professor with the Computer Science Department at the University of Oviedo. Since 2008, he is an Associate Professor with this department. He had participated in several research projects and contracts with public and private funding and had published contributions in indexed international journals and ranked conferences as well as several book chapters. He is a member of the IEEE Systems, Man & Cybernetics Chapter Society Spanish Chapter. His research interests are focussed in Genetic Fuzzy Systems, Meta-heuristics and Low Quality Data processing.

Alba Berzosa obtained her Computer Science Engineering Degree at the University of Burgos in 2007. Since 2009 she is working with the Artificial Intelligence and Applied Electronics at the Instituto Tecnológico de Castilla y León. She is the co-author of several contributions to ranked international conferences. Her research interests are focused in Genetic Fuzzy Systems, Meta-heuristics and Low Quality Data processing.

Enrique A. de la Cal Marín received the MSc and PhD degrees in computer science from the University of Oviedo, Oviedo, Spain, in 1995 and 2003,respectively. Previously, during 1995 was contracted like predoctoral researcher in CSIC Daza Valdes Research Institute. In 1996 he is contracted like associated professor in Oviedo University. Currently, he is Professor with the Department of Computer Sciences, the University of Oviedo. His research interests are in the fields of fuzzy-rule-based systems, soft computing industrial problems modeling, energy efficiency, automatic trading, genetic algorithms, and genetic programming.

Dr. Javier Sedano is an expert in the development of electronic systems – hardware – industrial projects and production-systems acquisition and control systems, "also in the design of connectionless models for the identification and modeling of dynamic systems". He has directed and participated in many research and development projects for the development of prototypes in pre-competitive projects, competitive and industrial research. He is the head of the group of Artificial Intelligent and Electronic Applied at the Instituto Tecnológico de Castilla y León. Also, he is part of the Applied Computational Intelligence Group at the University of Burgos, a few years working on projects and publications related to artificial intelligence and system modeling. It also collaborates in the organization of international scientific conferences, program committees and organization. He is a member of the IEEE Systems, Man & Cybernetics Chapter Society Spanish Chapter. It has some international publications, book chapters and participating in conference, as well as records of software and industrial patents.



Marco A. Garcia Tamargo has a degree in Computer Engineering from the University of Oviedo, and PhD from the same university. He has worked in a private company developing and implementing Urban Traffic Control Systems. Since 1996 he works as Associate Professor at the University of Oviedo. His main areas of research so far has been the Traffic Control Systems, Distributed Systems, Multiagent Systems and Intelligent Systems for Risk Assessment based on data mining.