2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

Multi-Agent Evolutionary Design of Beta Fuzzy Systems

Y. Jarraya, S. Bouaziz, Adel M. Alimi and A. Abraham

Abstract— **This paper provides an overview on a new evolutionary approach based on an intelligent multi-agent architecture to design Beta fuzzy systems (BFSs). The Methodology consists of two processes, a learning process using a clustering technique for the automated design of an initial Beta fuzzy system, and a multi-agent tuning process based on Particle Swarm Optimization algorithm to deal with the optimization of membership functions parameters and rule base. In this approach, dynamic agents use communication and interaction concepts to generate high-performance fuzzy systems. Experiments on several data sets were performed to show the effectiveness of the proposed method in terms of accuracy and convergence speed.**

I. INTRODUCTION

AT present, fuzzy system modeling is considered as one of the most important areas of application in fuzzy theory.

Fuzzy set, a concept proposed by Zadeh [1], has been widely investigated, due to its effectiveness in modeling the imprecision and uncertainty in human reasoning. Fuzzy logic is suitable for the representation of vague data and it provides an appropriate mechanism to describe the static and dynamic behavior of complex systems. In this sense, fuzzy systems are being used successfully in an increasing number of application areas such as classification [2]–[5], regression [6], [7], prediction [8], control problems [9]–[11], and general data mining problems [12], [13].

In general, fuzzy modeling includes identifying both the structure and the parameters of fuzzy systems. In this sense, intelligent optimization techniques such as artificial neural network [14],[15], clustering techniques [8], [13], and evolutionary computation [8], [16]–[20] have been successfully applied in this area in order to obtain the desired fuzzy model from the existing knowledge.

In this context, we introduce a novel methodology to design high-performance fuzzy systems especially Beta fuzzy systems (BFSs) for which the fuzzy basis functions used are Beta functions [21]–[23]. The use of Beta function shows a better performance against the other kind of functions due to its large flexibility and its universal approximation capacity [21]–[23].

Y. Jarraya , S. Bouaziz and Adel M. Alimi are with REsearch Groups on Intelligent Machines (REGIM), University of Sfax, National School of Engineers (ENIS), BP 1173, Sfax 3038, Tunisia; Tel +216-74-274-088; (email: { yosra.jarraya , souhir.bouaziz, adel.alimi}@ieee.org)
A. Abraham is with IT4Innovations, VSB- Technical University of Ostrava, Czech Republic and Machine Intelligence Research Labs (MIR Labs), Scientific Network for Innovation and Research Excellence, WA, USA; e-mail ajith.abraham@ieee.org

The new approach consists of two processes: a learning process and a multi-agent tuning process. In the first one, to approximate the desired output, the data are first clustered by Subtractive clustering algorithm. As a result, an initial structure of the model is obtained with a compact rule base. At this point, we use the optimization capability of Particle Swarm Optimization algorithm enhanced by a multi-agent architecture to tune the parameters of membership functions (shapes) and rules consequents. The basic idea here is to use the multi-agent architecture to improve the accuracy of the system and to accelerate the convergence behavior.

The proposed multi-agent system structure offers a decentralized model based on agents which are able to re-estimate their decisions if needed after negotiations. Information exchange takes place between the different agents to establish a unified decision. This kind of cooperative interaction improves the agent performances, and as a result, we obtain a Fuzzy Rule-Based System (FRBS) with high accuracy. On the other hand, different numbers of Function Evaluations (FEs) were noted in order to evaluate the speed of the proposed approach and to express its complexity.

The paper is planned as follows: Section II briefly introduces the Beat basis function. Section III provides a brief account of PSO algorithm. A description of the learning and tuning steps is detailed in section IV. Then, in section V, we propose a multi-agent architecture to improve the optimization capability of our system. For the purpose of illustration and validation of the approach, time series prediction datasets are used in section VI. Finally, conclusions are drawn in Section VII.

II. THE BETA BASIS FUNCTION

The first idea of using the Beta basis function for the design of fuzzy systems was introduced by Alimi in 1997 [21]. This leads to the appearance of the Beta fuzzy systems. A Beta Fuzzy Logic System is an FLS where the Beta functions are chosen as membership functions of the input variables. The use of Beta function was chosen for many advantages [21]–[23], such as its large flexibility, its ability to generate more rich shapes (linearity, asymmetry, etc) and its universal approximation characteristics. The Beta basis function used in the BFS is defined by:

$\beta(x) = \beta(x, x_0, x_1, p, q) =$

$$\begin{cases} \left(\frac{x - x_0}{c - x_0}\right)^p \left(\frac{x_1 - x}{x_1 - c}\right)^q & \text{if } x \in]x_0, x_1[\\ 0 & \text{elsewhere} \end{cases} \quad (1)$$

Where $p > 0$, $q > 0$, x_0 and x_1 are real parameters, and:

$c = \frac{px_1 + qx_0}{p+q}$ is the center of Beta function.

Let $\sigma = x_1 - x_0$. σ is the width of the Beta function which can be seen as a scale factor for the distance $x - c$.

$$\text{So: } \begin{cases} x_0 = c - \frac{\sigma p}{p+q} \\ x_1 = c + \frac{\sigma q}{p+q} \end{cases} \quad (2)$$

Equations (1) and (2) imply:

$$\begin{cases} \beta(x, c, \sigma, p, q) = \left[1 + \frac{(p+q)(x-c)}{\sigma p}\right]^p \left[1 - \frac{(p+q)(c-x)}{\sigma q}\right]^q \\ \qquad \text{if } x \in \left]c - \frac{\sigma p}{p+q}, c + \frac{\sigma q}{p+q}\right[\\ 0 \qquad \textit{elsewhere} \end{cases} \quad (3)$$

In the multi-dimensional case (dimension = d), the Beta function is defined by:

$$\beta(x, c, \sigma, p, q) = \prod_{i=1}^{d} \beta_i(x_i, c_i, \sigma_i, p_i, q_i) \quad (4)$$

Alimi has shown that if we have a given continuous real function and for any arbitrary precision, there exists a Beta fuzzy basis function expansion that approximates it.

III. PARTICLE SWARM OPTIMIZATION ALGORITHM

The Particle Swarm Optimization algorithm (PSO) developed by Kennedy and Eberhart [24] is a robust stochastic optimization algorithm based on swarm intelligence. It is inspired by the coordinated collective behavior of insects, birds and fish. PSO is based on swarm (group) behavior moving around in the search space in order to find the best particle (candidate solution).

The particles in the swarm co-operate. They exchange information about what they've discovered in the places they have visited. In each time step, each particle of the swarm is accelerated toward its p_{best} (personal best) and the g_{best} (global best) locations. The velocity of each particle *i* changes dynamically according to the following equation:

$$V_i(t+1) = w * V_i(t) + C_1 * R_1 * (p_{best} - X_i(t)) + C_2 * R_2 * (g_{best} - X_i(t)) \quad (5)$$

Having worked out a new velocity, the position of particle *i* is simply equal to its old position plus the new velocity:

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (6)$$

Where:

$V_i(t)$: Velocity of particle *i* at iteration *t*.

$X_i(t)$: Current position of particle *i* at iteration *t*.

C_1 and C_2: Constant weight factors.

w: inertia weight.

R_1 and R_1: Random factors in the range [0,1].

g_{best}: Best position found by the neighbors of particle *i*.

p_{best}: Best position achieved so long by particle *i*.

IV. EVOLUTIONARY DESIGN OF BETA FUZZY SYSTEMS

Fuzzy modeling has the purpose of identifying the parameters of a fuzzy system in order to achieve a desired behavior. In this sense, we propose a new fuzzy modeling method based on Beta membership functions. In the first step, a fuzzy clustering technique is used to derive initial Beta membership functions and fuzzy rules from numerical data. Then, a multi-agent PSO tuning process is applied to refine a preliminary Knowledge Base working at two different levels: adjusting the membership functions parameters and the consequent values of fuzzy rules.

In this section, after detailing the used learning method, we present a centralized tuning process based on PSO (Figure 1). In the next section, we will see how this process can be extended and optimized by introducing a multi-agent approach.

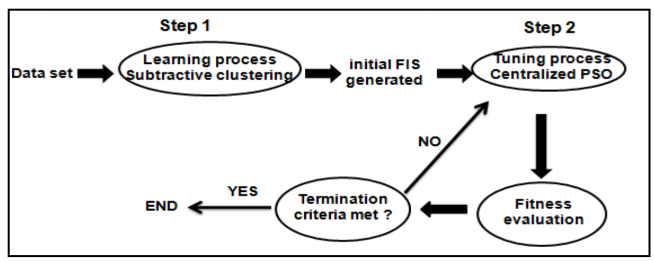

Fig. 1. Fuzzy modeling using centralized PSO as an optimization engine

A. Learning process based on subtractive clustering algorithm

One of the important tasks in fuzzy modeling is how to estimate fuzzy rules (structure identification). Generation of initial fuzzy rules can be done manually or automatically. In the first case, the expert's knowledge usually involves uncertainty and suffers from a loss of accuracy. In general, designers must spend a lot of time tuning fuzzy rules. In our work, instead of using a manual approach, a learning process is applied for the automated design of an initial structure of a fuzzy model from the available data. Here, the idea is to use an efficient clustering technique as the basis of a fast and robust method for fuzzy system identification.

Clustering plays a key role in searching for structures in Data. This technique is used to identify natural grouping of data from a large data set. It attempts to divide data into useful or meaningful subgroups named clusters. Therefore, the results must assure that samples which belong to the same cluster should be similar to one another and different from the samples of other clusters. In this context, we chose the subtractive clustering method by Chiu [25] to extract fuzzy rules from data. This algorithm is one-pass algorithm for estimating the number and location of cluster centers present in a collection of data points. The center candidates correspond to the data samples themselves. This method works by finding the point with the highest number of neighbors as a center for a cluster based on the density of surrounding data points. Hence, each cluster center is translated into a fuzzy rule. As a result, it can be said that the use of such technique simplifies a lot the task of extracting fuzzy rules and reduces the computational effort. Moreover, the resultant fuzzy rules are more adapted to the input data than they are in a fuzzy system generated without clustering. This reduces the problem of large number of rules (combinatorial explosion of rules) when the input data have a high dimension.

On the other hand, it should be noted that the radius of influence of cluster center r_a constitutes an important parameter of the subtractive clustering algorithm. It strongly affects the number of clusters that will be generated. A small r_a can produces excessive number of clusters (resulting in many rules). Generally, specifying large r_a results in fewer clusters (resulting in fewer rules).

To summarize, in our research, the number of cluster centers is equal to the number of the fuzzy rules and the obtained cluster centers will constitute the Beta Membership functions centers. As a result, the initial structure of the fuzzy system will be built up and we obtain a compact rule base with a reduced number of rules. Since the proposed tuning method does not reduce the rule base size, this fact fits well to the design approach.

More details of Subtractive Clustering algorithm are presented in [25].

B. *Centralized tuning process based on Particle Swarm Optimization algorithm*

In the tuning process, we start from a previous existing FRBS derived by the learning method previously described. In this level, PSO is applied for further optimization of the initial generated fuzzy inference system (FIS). The optimization process deals with a tuning of membership functions parameters and rule consequents in order to obtain the most accurate fuzzy system.

The context of fuzzy modeling involves an important consideration of how to encode the solution. In this work, the parameters and the rule base act as particles which look for the global best fitness. As shown in Figure 2, each particle of the swarm represents a fuzzy model (a population is represented by a set of particles). After every generation, we calculate the fitness function of each fuzzy model. Then, we update positions of particles through PSO algorithm until we follow the best fuzzy model.

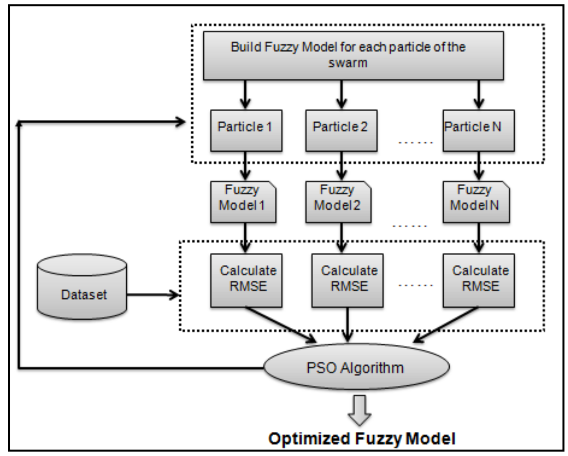

Fig. 2. Fuzzy Model Identification through PSO

The main idea here is to generate a FIS from each particle of the swarm. The FIS structure is an object containing all the information about the FIS, i.e. definitions of membership function, rule base, names of variables, etc. An FIS has a structure that can be easily modified. This flexibility has been exploited for optimization of the parameters and rule base using PSO encoding mechanism.

C. *Encoding mechanism of the fuzzy system*

One of the most important steps is to provide an efficient encoding method. In most cases, triangular and Gaussian functions are usually used as fuzzy basis functions. However, we use the Beta function due to its large flexibility. This latter shows a better performance against the other type of functions.

Seen that the type of the membership functions used is the Beta function, four adjustable parameters must be considered in the learning and tuning process (the center c, the width σ and the form parameters p and q). Subtractive clustering algorithm tunes only centers of membership functions c, and uses constant values of σ, p and q. Consequently, the widths, the form parameters and the consequent values of rules are encoded into particles to be tuned via PSO as shown in Figure 3.

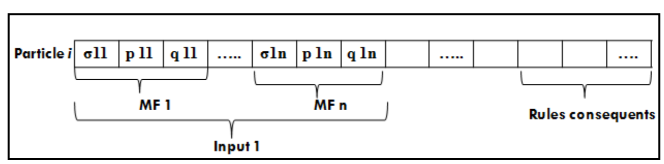

Fig. 3. Representation of a fuzzy model by a particle

To summarize, each particle of the swarm represents a fuzzy model whose membership function parameters and rule consequents are optimized through PSO algorithm to follow the best fuzzy model.

D. Fitness Function

The difference between the calculated output and the actual output (as given in the dataset) defines the error. In this work, the fitness of an individual is determined by calculating a biased Root Mean Squared Error (RMSE) of the fuzzy model:

$$Fit(i) = \sqrt{\frac{1}{P}\sum_{j=1}^{p}(y_t^j - y_{out}^j)^2} \quad (7)$$

where p is the total number of samples, y_t^j and y_{out}^j are respectively observed output and predicted output. $Fit(i)$ defines the fitness function value of the i^{th} individual.

E. The centralized evolving algorithm

The process for the identification of fuzzy model using the centralized PSO is represented as pseudo–code by the following steps:

Step 1.
Generate an initial random population of candidate solutions.

Step 2.
Apply the learning process (subtractive clustering technique) to derive the initial structure of the fuzzy model as shown in section *A*. In this step, we will determine the number of Beta MFs and rules.

Step 3.
Tune the Beta MFs parameters (σ, p, q parameters) and the previously obtained rule base by applying the centralized tuning process through PSO algorithm (section *B*).

Step 4.
Evaluate the fitness value (RMSE) of each particle.
Update individual and global best positions (Gbest and Pbest) by comparing the newly evaluated fitness against the previous individual and global best fitness.

Step 5.
For every particle:

- Calculate its velocity according to equation (5).
- Update its position according to equation (6)

Step 6.
Increment the generation number (g = g + 1).

Step 7.
If g = G_{max} then exit, otherwise go to step 3.

Step 8.
The obtained best particle will be used to extract the final optimized fuzzy model.

V. MULTI-AGENT ARCHITECTURE

A multi agent system is a set of entities that interact together to solve a problem or to reach a goal. In this work, the multi-agent approach is introduced in the level of the tuning process to improve the optimization performance of PSO algorithm. We attempt to develop an intelligent decision-making model based on multiple cooperative and negotiator agents to obtain the desired optimized fuzzy modeling system. Agents offer several potential advantages over the traditional centralized approach. Workers agents are not meant to work in isolation; they rather cooperate and are usually in a competitive interaction, and this allows a dynamic adaptation of their behavior. As a result, agents can re-estimate their decisions if needed during the negotiation process to avoid false convergence behavior. The main idea here is to affect to every agent its own PSO to be executed in order to generate its best solution (best particle) as shown in Figure 4. The initial population of each agent is randomly generated. In this context, we can introduce two types of communications:

- *Agent-Agent communication*

When executing its own PSO and after every generation, each agent worker sends its global best particle (g_{best}) and the correspondent fitness value to the other agents. On the other hand, each worker will receive this g_{best} and will accept it for the next generation if and only if it yields a reduction in the value of the objective function. Otherwise, it will reject it.

This step of negotiation is expected to provide better convergence behavior of agents and also leads to a considerable reduction in convergence time.

- *Agent-Coordinator communication*

In the next step, when all agents have completed the execution of their PSO algorithms, the final best solution of each agent will be sent to the coordinator agent. This latter will form a new population by collecting all the best particles previously generated and then executes again a PSO algorithm. As a result, a Global best solution (G_{best}) is obtained and then injected in the population of the least efficient agent (having the highest value of fitness function). This injection aims to repeat the same concept of parallelism until a satisfactory solution is found or a maximum global iteration number is reached.

To end, at each iteration, the new populations of agents will be formed as follows: the population of the least confident agent will contain the global best particle of the whole system G_{best} and the other particles will be liquidated at random. And for the other agents, their new populations will contain their own best particles and also random particles.

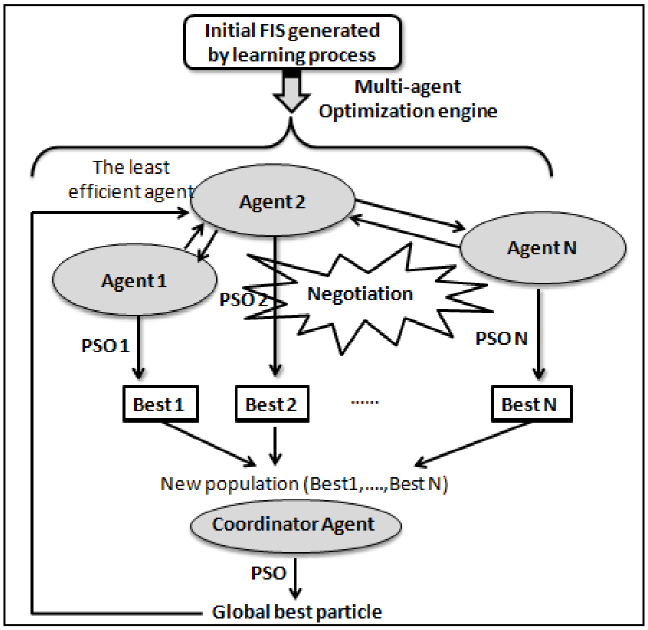

Fig. 4. General concept of fuzzy multi-agent tuning process

The aim of using a multi-agent architecture is to overcome the problems of premature convergence and slow of the classical centralized approach.

VI. Experimental Results

For the purpose of illustration and validation of the approach, the performance of the proposed method is evaluated for time series prediction datasets. A number of well-known benchmark problems such as "Mackey-Glass chaotic", "Jenkins–Box" and "sunspot number" were employed. The best-suited sets of parameters used in the experimentation study are listed in table 1. We ran the simulation 10 times and then we averaged the results. The best solution in each case was marked in bold.

TABLE I. List of Intial Parameters

PSO	
Parameter	***Initial value***
Population size (*NP*)	20
c_1	0.2
c_2	0.6
subtractive clustering	
Parameter	***Initial value***
cluster radius (MG problem)	0.8
cluster radius (BJ problem)	0.6
cluster radius (sunspot number problem)	0.7

The tests results obtained are used to evaluate the effectiveness of the proposed method with respect to two measures of performance which are solution quality (through RMSE values) and convergence speed (through the number of Function Evaluations FEs marked). Accuracy and efficiency of the proposed methodology was demonstrated through comparisons with other available fuzzy/neural learning approaches developed in recent years.

A. Mackey–Glass time series prediction

The Mackey-Glass series (MG) [26] is one of the well established benchmark problems which have been intensively studied in several previous works. This time series is based on the following Mackey–Glass differential equation:

$$\frac{d(x(t))}{dt} = \frac{ax(t-\tau)}{1+x^c(t-\tau)} - bx(t) \qquad (8)$$

Parameters are selected as follows: a = 0.2, b = 0.1, c=10 and $\tau \geq 17$. The aim is to predict the time series at point $x(t+6)$. The inputs variables are respectively $x(t), x(t-6$, $xt-12$ and $xt-18$. In this study, 500 samples were selected for training and the next 500 samples were used for validating the identified model.

The obtained RMSE values for the training data and test data respectively are 1.7902e-016 and 1.6883e-016. The number of Function Evaluations FEs used to find the optimum fuzzy model is equal to 56 which affirms the considerable reduction in convergence time. Figure 5 and 6 draw the actual and the forecast outputs for training and testing data.

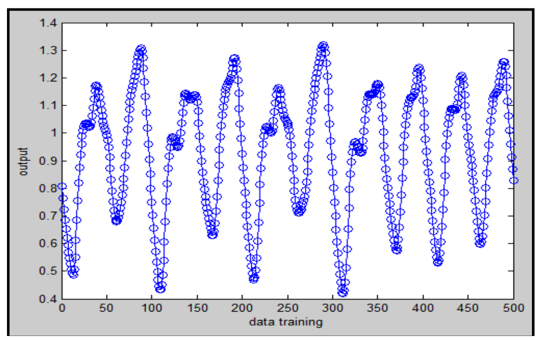

Fig. 5. The actual output (solid line) and the forecast output (circles) for training data in the case of Mackey-Glass

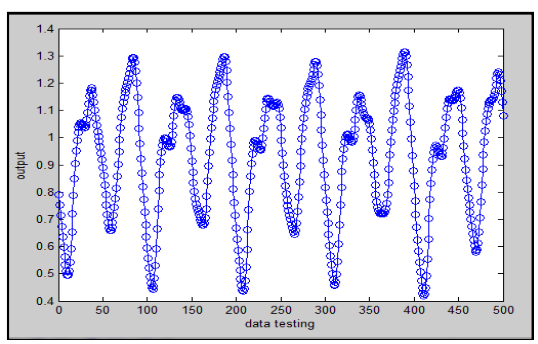

Fig. 6. The The actual output (solid line) and the forecast output (circles) for testing data in the case of Mackey-Glass

The experimental results indicate that the proposed system performs well in terms of accuracy and convergence speed. For a fair experimentation study, the results of the proposed method were compared with some of the previous studies.

The performance comparison to other works is given in Table 2. As observed, the prediction performance and accuracy by the proposed approach is much better than other compared approaches. But on the other hand, the number of FEs of those methods isn't mentioned.

TABLE II. COMPARISON RESULTS FOR THE PREDICTION OF MACKEY-GLASS TIME-SERIES

Method	Training error (RMSE)	Testing error (RMSE)
ADANN-GA [27]	0.055	-
ADANN-DE [27]	0.025	-
ADANN-EDA [27]	0.012	-
HCMSPSO [28]	0.0095	0.0208
HMDDE–BBFNN [29]	0.0094	0.0170
G-FNN [30]	0.0063	0.0056
HL-NFS [31]	0.0014	0.0013
NARMA [32]	0.000638	0.000627
LQFNM [33]	0.000543	-
FIS with non- uniform embedding [34]	0.0003497	-
FBONT_EIP& HBFOA [15]	5.3430e-10	1.8630e-09
The proposed algorithm	**1.7902e-016**	**1.6883e-016**

B. Box and Jenkins' Gas Furnace Problem

The gas furnace data (BJ) [35] is also used as a benchmark problem. This time series problem is related to a combustion process of a methane-air mixture where CO2 concentration in outlet gas is used as output, y(t), and the input gas flow into the furnace is u(t). From the 296 observations taken from a laboratory furnace [39], the first 200 were used for training and the remaining 96 were exploited for testing the proposed method. We aim to predict y(t) in terms of y(t-1) and u(t-4).

After doing 98 number of Function Evaluations (FEs = 98), the optimal fuzzy model was generated with the RMSE 0.0049. For validation of data set, the RMSE value is 0.0125. Figures 7 and 8 show the actual and the predicted time series for training and testing data. Table 3 shows the comparison of test results of several models for Box–Jenkins data prediction problem. From this Table, it is clear that our method is having less training and testing errors in comparison to the other contributions.

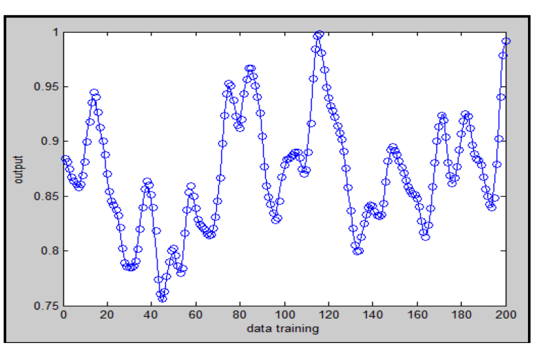

Fig. 7. The actual output (solid line) and the forecast output (circles) for training data in the case of Box and Jenkins

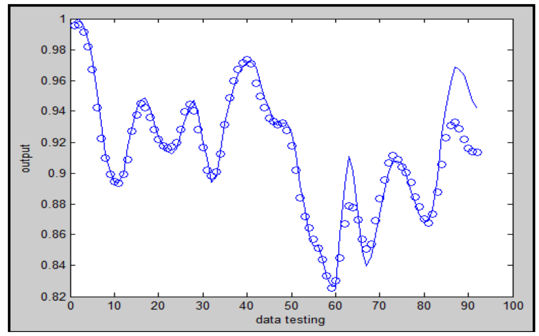

Fig. 8. The actual output (solid line) and the forecast output (circles) for testing data in the case of Box and Jenkins

TABLE III. COMPARISON RESULTS FOR THE PREDICTION OF BOX AND JENKINS TIME-SERIES

Method	Training error (RMSE)	Testing error (RMSE)
Subtractive clustering [36]	0.6134	-
KDE-based clustering [36]	0.6112	-
HMDDE–BBFNN [29]	0.2411	0.3745
HPSO [37]	0.2258	0.3876
COPSO-MSN [38]	0.2151	0.3416
ANFIS [38]	0.0374	0.0640
eTS [39]	-	0.04904
The proposed algorithm	**0.0049**	**0.0125**

C. Prediction of sunspot number time series

The sunspot number time series corresponds to the annual average number of sunspots. This dataset represents the yearly average relative number of sunspot observed [40].

The $y(t-4), y(t-3), y(t-2)$ and $y(t-1)$ are used as inputs to the fuzzy system for the prediction of the output y(t). Data points between 1700 and 1920 are used for system training. Then, two additional sets of data are employed for the test set. The first one is from 1921 to 1955 and the second is from 1956 to 1979.

After performing 78 Function Evaluations (FEs = 78), an optimal fuzzy model was obtained with RMSE 2.1316e-016. The RMSE value for the first data set validation is 5.6314e-005 and for the second data set validation is 3.1931e-005. From simulations and results, it's remarkable that our proposed method uses a reduced number of FEs to reach high accuracy solutions.

Figures 9, 10 and 11 illustrate the observed and the fitted time series for training and testing data (containing the two test cases). As presented in Table 4, a comparison is performed over a diverse collection of techniques. It is shown that the method proposed is competitive and robust in terms of accuracy and speed.

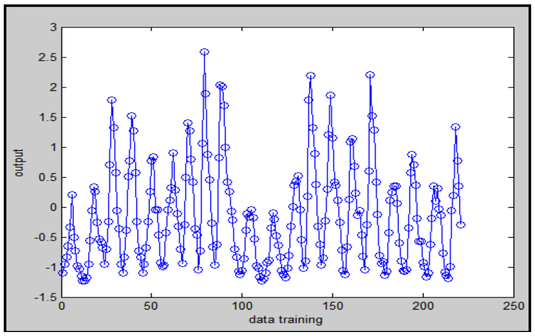

Fig. 9. The actual output (solid line) and the forecast output (circles) for training data in the case of sunspot number time series

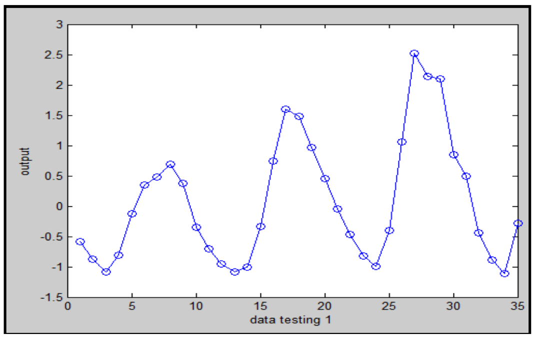

Fig. 10. The actual output (solid line) and the forecast output (circles) for the first testing data in the case of sunspot number time series

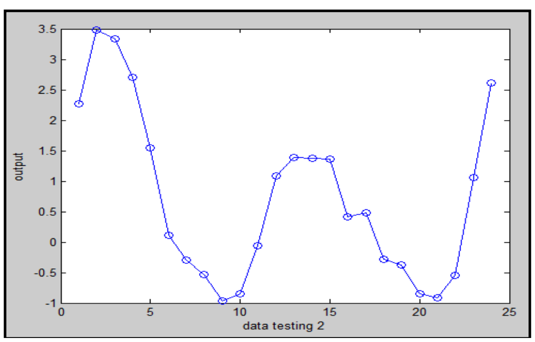

Fig. 11. The actual output (solid line) and the forecast output (circles) for the second testing data in the case of sunspot number time series

TABLE IV. COMPARISON RESULTS FOR THE PREDICTION OF SUNSPOT NUMBER TIME SERIES

Method	RMSE Training	RMSE Testing 1	RMSE Testing 2
FWNN-S [41]	0.0895	0.1093	0.1510
RFNN [42]	-	0.074	0.21
ABC-BBFNN [43]	0.0012	0.0018	0.0044
The proposed algorithm	**2.1316e-016**	**5.6314e-005**	**3.1931e-005**

VII. CONCLUSION

In this paper, an automated design of an initial Beta fuzzy system is first proposed by using subtractive clustering algorithm. Then, a tuning process is applied including a multi-agent Particle Swarm Optimization algorithm in order to tune the shapes of membership functions and rule consequents. This new hybridization aims to gather and make use of the clustering ability in extracting the knowledge base from data, the PSO ability to exchange social information, the high flexibility of Beta function and the robustness and convergence speed of the multi-agent architecture. The experimental results seem to be very encouraging as compared with other contributions and demonstrate the effectiveness of such an approach.

ACKNOWLEDGMENT

The authors would like to acknowledge the financial support of this work by grants from General Direction of Scientific Research (DGRST), Tunisia, under the ARUB program. This work was also supported in the framework of the IT4 Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 by operational programme ‘Research and Development for Innovations’ funded by the Structural Funds of the European Union and state budget of the Czech Republic, EU.

REFERENCES

[1] L. A. Zadeh, “Fuzzy sets”, Information and Control, vol. 8, (1965).

[2] C. Setzkorn and R. C. Paton, “On the use of multi-objective evolutionary algorithms for the induction of fuzzy classification rule systems,” Biosyst., vol. 81, no. 2, pp. 101–112, 2005.

[3] P. Ducange, B. Lazzerini, and F. Marcelloni, “Multi-objective genetic fuzzy classifiers for imbalanced and cost-sensitive datasets,” Soft Comput., vol. 14, no. 7, pp. 713–728, 2010.

[4] R. Alcal´a, Y. Nojima, F. Herrera, and H. Ishibuchi, “Multiobjective genetic fuzzy rule selection of single granularity-based fuzzy classification rules and its interaction with the lateral tuning of membership functions,” Soft Comput., vol. 15, no. 12, pp. 2303–2318, 2011.

[5] Y. Nojima and H. Ishibuchi, “Incorporation of user preference intomultiobjective genetic fuzzy rule selection for pattern classification problems,” Artif. Life Rob., vol. 14, pp. 418–421, 2009.

[6] P. Pulkkinen and H. Koivisto, “A dynamically constrained multiobjective genetic fuzzy system for regression problems,” IEEE Trans. Fuzzy Syst., 18(1), pp. 161–177, Feb. 2010.

[7] R. Alcal´a, M. J. Gacto, and F. Herrera, “A fast and scalable multiobjective genetic fuzzy system for linguistic fuzzy modeling in highdimensional regression problems,” IEEE Trans. Fuzzy Syst., vol. 19, no. 4, pp. 666–681, Aug. 2011.

[8] Y. Jarraya, S. Bouaziz, Adel M. Alimi, A. Abraham, “Fuzzy Modeling System based on Hybrid Evolutionary Approach”, 13th International Conference on Hybrid Intelligent Systems (HIS), December 4-6, 2013, Yassmine Hammamet – Tunisia.

[9] R. Alcal´a, J. Alcal´a-Fdez, M. J. Gacto, and F. Herrera, “Improving fuzzy logic controllers obtained by experts: A case study in HVAC systems,” Appl. Intell., vol. 31, pp. 15–30, 2009.

[10] F. Hoffmann, “Evolutionary algorithms for fuzzy control system design,” Proc. IEEE, 89(9), pp. 1318–1333, Sep. 2001.

[11] M. J. Gacto, R. Alcal´a, and F. Herrera, "A multi-objective evolutionary algorithm for an effective tuning of fuzzy logic controllers in heating, ventilating and air conditioning systems," Appl. Intell., vol. 36, no. 2, pp. 330–347, 2012.

[12] J. Casillas and F. J. Martinez-Lopez, "Mining uncertain data with multiobjective genetic fuzzy systems to be applied in consumer behaviour modelling," Expert Syst. Appl., vol. 36, no. 2, Part 1, pp. 1645–1659, 2009.

[13] R. Alhajj and M. Kaya, "Multi-objective genetic algorithms based automated clustering for fuzzy association rules mining," J. Intell. Inf. Syst., vol. 31, pp. 243–264, 2008.

[14] S. Bouaziz, A.M. Alimi and A. Abraham, "Extended Immune Programming and Opposite-based PSO for Evolving Flexible Beta Basis Function Neural Tree", IEEE International Conference on Cybernetics, pp. 13 –18, Lausanne Switzerland, 13-15 June 2013.

[15] S. Bouaziz, A.M. Alimi and A. Abraham, "Evolving Flexible Beta Basis Function Neural Tree for nonlinear systems", International Joint Conference on Neural Networks, Dallas Texas, 4-9 August 2013.

[16] M. Ammar, S. Bouaziz, A.M. Alimi, A. Abraham, "Hybrid Harmony Search algorithm for Global Optimization", Fifth World Congress on Nature and Biologically Inspired Computing, pp. 69–75, Fargo-USA, 12-14 August 2013.

[17] Y. Chen, B. Yang, A. Abraham, and L. Peng, "Automatic design of hierarchical Takagi-Sugeno type fuzzy systems using evolutionary algorithms," IEEE Transactions on Fuzzy Systems, vol. 15, no. 3, pp. 385–397, 2007.

[18] Cordon O, Herrera F, Hoffmann F, Magdalena L (2001) ,'"Genetic fuzzy systems-evolutionary tuning and learning of fuzzy knowledge bases". Advances in fuzzy systems-applications and theory, World Scientific, vol 19, p 462.

[19] Y. Jarraya, S. Bouaziz, A.M. Alimi, A. Abraham, "The Adaptive Chemotactic Foraging with Differential Evolution algorithm", Fifth World Congress on Nature and Biologically Inspired Computing, Fargo-USA, pp. 63–68, 2013.

[20] Y. Jarraya, S. Bouaziz, A.M. Alimi, A. Abraham, "A Hybrid Computational Chemotaxis in Bacterial Foraging Optimization Algorithm for Global Numerical Optimization", IEEE International Conference on Cybernetics, Lausanne Switzerland, pp. 213 –218, 2013.

[21] Adel M. Alimi, "The Beta Fuzzy System: Approximation of Standard Membership Functions", Proc. 17eme Journees Tunisiennes d'Electrotechnique et d'Automatique: JTEA'97,Nabeul, Tunisia, Nov., vol. 1, pp. 108-112, 1997.

[22] Adel M. Alimi, "Beta fuzzy basis functions for the design of universal robust neuro-fuzzy controllers", In: Proceeding of S´eminaire sur la Commande Robuste ses Applications: SCRA'97, Nabeul, Tunisia, pp C1–C5, 1997.

[23] Adel M. Alimi, "The Beta System: Toward a Change in Our Use of Neuro-Fuzzy Systems", In-ternational Journal of Management, Invited Paper, June, pp. 15-19, 2000.

[24] J. Kennedy, R.C. Eberhart, "Particle swarm optimization", Proceedings of the IEEE International Conference on Neural Networks. IEEE Press, Piscataway, NJ, vol. 4, pp. 1942–1948, 1995.

[25] S. Chiu, "Fuzzy Model Identification Based on Cluster Estimation,"Journal of Intelligent & Fuzzy Systems, Vol. 2, No. 3, Spet. 1994.

[26] M.C. Mackey, L. Glass, "Oscillation and chaos in physiological control systems", Science 197 (1977) 287–289.

[27] J. P. Donate , X. Li, G. G. Sanchez, A. S. de Miguel « Time series forecasting by evolving artificial neural networks with genetic algorithms, differential evolution and estimation of distribution algorithm », journal of Neural Computing and Applications, 2013.

[28] C.F. Juang, C.M. Hsiao, C.H. Hsu, "Hierarchical Cluster-Based Multispecies Particle-Swarm optimization for Fuzzy-System Optimization", IEEE Transactions on Fuzzy Systems, vol. 18(1), pp. 14-26, 2010.

[29] H. Dhahri, Adel M. Alimi, A. Abraham, "Hierarchical multi-dimensional differential evolution for the design of beta basis function neural network", neurocomputing, vol. 79, pp.131-140, 2012.

[30] Y. Gao, M. Joo Er, "Soft cumputing approach for time series prediction", 16th IFAC World Congress in Prague, Czech Republic, 2005.

[31] C. Li, J. Hu, T. Chiang, T. Wu, "Computational Intelligence Hybrid Learning Approach to Time Series Forecasting", World Academy of Science, Engineering and Technology, 2010.

[32] C. G. da Silva, "Time series forecasting with a non-linear model and the scatter search meta-heuristic", Fifth Internation Conference on Machine Learning and Cybernetics, 2008.

[33] M. J. Mahjoob, M. Abdollahzade, R. Zarringhalam, A. Kalhor, "Chaotic time series forecasting using locally quadratic fuzzy neural models", FS'08 Proceedings of the 9th WSEAS International Conference on Fuzzy Systems, 2008.

[34] K. Lukoseviciute, M. Ragulskis, "Evolutionary algorithms for the selection of time lags for time series forecasting by fuzzy inference systems". Journal of Neurocomputing 73 (2010) 2077–2088, 2010.

[35] G.E.P. Box, G.M. Jenkins, "Time Series Analysis--Forecasting and Control", Holden Day, San Francisco, CA, 1976.

[36] S. Łukasik, P. Kowalski, M. Charytanowicz and P. Kulczycki, "Fuzzy Model Identification Using Kernel-Density-Based Clustering", in Proceedings of the Sixth International Workshop on Intuitionistic Fuzzy Sets and Generalized Nets, Warsaw, 2007.

[37] H. Dhahri, Adel M. Alimi, A. Abraham, "Hierarchical Particle Swarm Optimization for the Design of Beta Basis FunctionNeural Network", Intelligent Informatics, AISC 182, pp. 193-205. Springer-Verlag Berlin Heidelberg 2013.

[38] B.Samanta, "Prediction of chaotic time series using computational intelligence", Expert Systems with Applications: An International Journal, Volume 38 Issue 9, September, 2011, Pages 11406-11411

[39] P. Angelov, D. Filev, "Simpl-eTS: a simplified method for learning evolving Takagi-Sugeno fuzzy models," IEEE International Conference on Fuzzy Systems, pp: 1068-1073, 2005.

[40] A. J. Izeman , J.R. Wolf and the Zurich, "sunspot relative numbers", The Mathematical Intelligence, vol.7, no.1, pp. 27-33, 1985.

[41] S. Yilmaz, Y. Oysal, "Fuzzy wavelet neural network models for prediction and identification of dynamical systems", IEEE Transactions on Neural Networks, 2010: 1599~1609.

[42] R.A. Aliev, B.G. Guirimov, R.R. Aliev, "Evolutionary algorithm-based learning of fuzzy neural networks", Part 2: Recurrent fuzzy neural networks, Fuzzy Sets and Systems, vol. 160 (17), 2009.

[43] H. Dhahri, Adel M. Alimi, A. Abraham, "Designing Beta Basis Function Neural Network for Optimization Using Artificial Bee Colony (ABC)". WCCI 2012 IEEE World Congress on Computational Intelligence June, 10-15, 2012 - Brisbane, Australia.