A Hybrid Algorithm for Parameter Tuning in Fuzzy Model Identification

Zsolt Csaba Johanyák, Olga Papp

Department of Information Technologies, Faculty of Mechanical Engineering and Automation, Kecskemét College, Izsáki út 10, H-6000 Kecskemét, Hungary
E-mail: {johanyak.csaba, papp.olga}@gamf.kefo.hu

Abstract: Parameter tuning is an important step in automatic fuzzy model identification from sample data. It aims at the determination of quasi-optimal parameter values for fuzzy inference systems using an adequate search technique. In this paper, we introduce a new hybrid search algorithm that uses a variant of the cross-entropy (CE) method for global search purposes and a hill climbing type approach to improve the intermediate results obtained by CE in each iteration stage. The new algorithm was tested against four data sets for benchmark purposes and ensured promising results.

Keywords: cross-entropy; hill climbing; fuzzy rule interpolation; fuzzy model identification

1 Introduction

Fuzzy systems have been successfully applied in a wide range of areas in this century and the previous. Typical fields are controllers (e.g. [32] [33] [35]), expert systems (e.g. [11] [24]), clustering (e.g. [9] [28] [40]), fuzzy modeling (e.g. [18]), management decision support (e.g. [31] [42]), time series estimation (e.g. [14]), etc. The proper functioning of such systems greatly depends on the underlying rule base. Thus, the methods used for its automatic generation and the determination of the rules' optimal parameters become particularly important.

There are several methods for the automatic generation of the rule base from sample data. Generally, they form two main groups. The methods belonging to the first group (e.g. [6] [8] [41]) create the rule base in two steps. Firstly, they define the structure by creating an initial rule base, and next, they look for an optimal parameter set applying a search algorithm. The methods belonging to the second group (e.g. [19] [39]) differ from this approach only in their second step, when they allow the modification of the structure by creating new rules or deleting old ones.

In our previous work [20], we presented a comparative analysis of a global and a local search algorithm for parameter optimization. They were applied in the second step of a rule base generation conforming to the above mentioned first approach. As a result of the analysis, we found that the local search algorithm ensured a significant improvement of the system performance in case of the used benchmark problems. In comparison, the global search method improved the system performance on three out of four benchmark problems; however its running time was remarkably better than the local heuristic's. This prompted us to implement a hybrid approach, where after enhancing some parts of the two algorithms; we combined the quick run of the global search technique with the increased accuracy of the local heuristic.

In this paper, we present this new hybrid algorithm and the results obtained by its application for finding optimal parameters in the case of the same benchmarking problems as used in [20]. The rest of this paper is organized as follows. Section 2 presents the applied global (sec. 2.1) and local (sec. 2.2) search methods as well as the concept of their combination. Section 3 gives a brief review of the applied fuzzy inference technique. Section 4 reports the results of the tests.

2 Parameter Tuning

The starting point is an initial rule base created with an arbitrary method (e.g. based on fuzzy clustering) automatically from sample data or manually by a human expert. Next, by the help of parameter tuning one tries to find such values for the parameters of the rules that ensure a better performance for the fuzzy system. The performance evaluation method we applied is discussed in sec. 2.4. In the following three subsections we present two search techniques and their proposed integration.

2.1 Cross-Entropy Method

The Cross-Entropy (CE) method is a global search algorithm used for solving continuous multi-extremal and discrete optimization problems, such as buffer allocation [2], static simulation models [12], control and navigation [10], reinforcement learning [27] and others. Its original version was proposed by Rubinstein [34]. The method does not use the local neighborhood structure, instead it works as a black-box and looks for the optimal parameter values using an iterative approach.

Suppose we want to find the best parameter vector p for which our black box yields a performance index $PI(p)$. This parameter (p) should be between a given lower bound (lb) and upper bound (ub). Starting with the first iteration, an initial

probability parameter vector (pr_0) is optimized for each parameter, for example pr_0={0.5, 0.5, ... 0.5}.

In each iteration step i, $S(p_1, p_2,..., p_S)$ samples are generated according to the latest pr_{i-1} probability vector values. Performance index values are calculated for each generated sample, and according to its values the samples are ordered increasingly. After ordering the samples, one of them is chosen according to a parameter q for comparison. The sample with the performance index $g_i = PI_{[1-q]N}$ is chosen. Using g_i the new probability parameter, values are determined by

$$pr_i = \frac{\sum_i I(PI(p_i) \geq g_i) I(p_i \geq lb_i) I(p_i \leq ub_i)}{\sum_i I(PI(p_i) \geq g_i)}, \quad (1)$$

where I is an indicator function which returns *1* if the condition in its parenthesis is true, and *0* otherwise.

The algorithm generates a series of performance index values g_i which get smaller with each iteration, approaching the desired minimum.

The number of the iteration cycles (n_{iCE}), the number of generated samples for each iteration (S), and the optimization parameter q are parameters of the method.

2.2 Hill Climbing Type Local Search

The local search algorithm presented in this subsection is a modified version of the algorithm used by the ACP [16] rule base generation method. It searches for better parameter values through several iterations by applying a hill climbing type approach. The number of iteration cycles (n_{iHC}) is a parameter of the method.

In each cycle all parameters (in all antecedent and consequent dimensions for all fuzzy sets) are examined one-by-one. In the case of each parameter $2 \cdot n_p$ new values are calculated (see Fig. 1) and the fuzzy system is evaluated against the training data set for each new parameter value. Finally, that parameter value is kept from the $2 \cdot n_p + 1$ ($2 \cdot n_p$ new and the original one) candidates that ensures the best system performance. The new parameter values are calculated from the original one by increasing/decreasing its value as follows

$$\begin{aligned} p_i^k &= p_0^k + i \cdot s, i = \overline{1, n_p}, \\ p_i^k &= p_0^k - (i - n_p) \cdot s, i = \overline{n_p + 1, 2 \cdot n_p}, \end{aligned} \quad (2)$$

where p_0^k is the original value of the kth parameter of a fuzzy set, s is the actual step, and n_p is a parameter of the method. Owing to the possible different ranges of the partitions in different dimensions, the step size is calculated by

$$s = c_s \cdot r \,, \tag{3}$$

where r is the range of the actual partition defined by its upper (r_2) and lower (r_1) bounds,

$$r = r_2 - r_1 \,, \tag{4}$$

and $c_s \in [0, 1]$ is the step coefficient, which is also a parameter of the method.

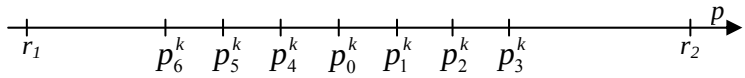

Figure 1
Original and new values of a fuzzy set's *k*th parameter in case of n_p=3

After calculating the new parameter values, some constraints are applied to preserve the validity and interpretability of the resulting fuzzy sets. These constraints strongly depend on the used membership function types and the parameterization technique. Further on we will present the constraints for the case of piece-wise linear membership functions and break-point type parameterization.

1. The new (*i*th) parameter value must remain inside its neighbors.
   - If the new value of the actual (*k*th) parameter is smaller than the previous parameter, it will be increased to that parameter's value

     $$p_i^k = \max\left(p_i^{k-1}, p_i^k\right), k = \overline{2, n_s}, i = \overline{1,2 \cdot n_p} \,, \tag{5}$$

     where n_s is the number of a fuzzy set's parameters.
   - If the new value is greater than the next parameter it will be reduced to that parameter's value

     $$p_i^k = \min\left(p_i^k, p_i^{k+1}\right), k = \overline{1, n_s - 1}, i = \overline{1,2 \cdot n_p} \,. \tag{6}$$

2. The set must remain at least partially inside the range.
   - The first parameter must always be smaller or equal to the upper bound of the range of the current linguistic variable (r_2)

     $$p_i^1 = \min\left(p_i^1, r_2\right), i = \overline{1,2 \cdot n_p} \,. \tag{7}$$

   - The last parameter must always be greater or equal to the lower bound of the range of the current linguistic variable (r_1)

     $$p_i^{n_s} = \max\left(r_1, p_i^{n_s}\right), i = \overline{1,2 \cdot n_p} \,. \tag{8}$$

Owing to the above-mentioned corrections, two or more new parameter values could result identical. Therefore, the duplicate values are removed from the parameter vector *p*.

Another feature of the algorithm is that the step coefficient c_s is decreased when the amelioration of the performance index in the course of two consecutive iteration cycles is smaller than the threshold value (dPI_{tr})

$$c_s = c_s \cdot c_d, c_d \in [0,1], \tag{9}$$

where c_d is the decrement coefficient. Its value, as well as the value of dPI_{tr}, are parameters of the algorithm.

2.3 The Hybrid Approach

The basic idea of the hybrid approach is that the local search method is integrated with the global technique as follows. The parameter tuning is started with five steps of the global search method presented in sec. 2.1, where after selection of the samples $\{p_i \mid PI(p_i) \geq g_i\}$ for each selected sample, a local search is launched to find better parameter values in the neighborhood of the initial values determined by the previous step of the CE method. The local search is performed by executing one, two, respectively three steps as indicated in sec. 2.2. The local search results in for each x_i a new p_i^* value set with $PI(p_i^*) \geq PI(p_i)$ performances. Next, the new p_i^* samples are used for the calculation of the probability parameters in (1).

After each parameter modification and system evaluation, the whole parameter set (fuzzy system) and its performance measure against the training data set are saved. After finishing the tuning process, all saved parameter sets are tested against the test data set (PI_{te}) as well. The variation of PI_{tr} and PI_{te} give a good picture about the tuning process, indicating clearly in most of the cases the phenomenon of parameter overfitting to the train data.

For example, supposing an error related performance index which is of type “the smaller the better”, Fig. 2 illustrates the variation of the performance indexes in the function of the number of system evaluations.

In order to minimize the overfitting effect and get a system performing well on the entire input space, an overall system performance (PI_{ov}) is calculated, which takes into consideration both the training and the test data sets. Finally, that parameter set is chosen as the best one that ensures the best PI_{ov} value (indicated by an arrow in Fig. 2).

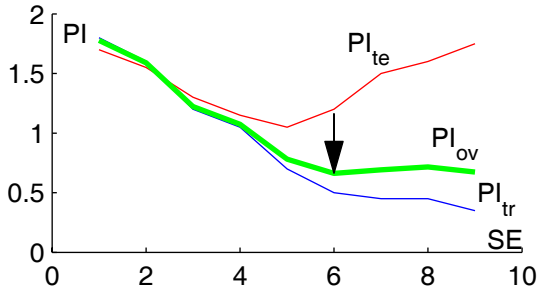

Figure 2
Variation of the performance index in case of the train and test data and the overall performance index in function of the number of system evaluations

2.4 Performance Evaluation

The performance index (*PI*) expresses the quality of the approximation ensured by the fuzzy system using a number that aggregates and evaluates the differences between the prescribed output values and the output values calculated by the fuzzy system. We used as the performance index of the resulting fuzzy systems the root mean squared error, expressed in percentage, of the output variable's range. It was chosen because it facilitates the interpretation of the error and its benchmarking against the width of the variation interval of the output. It is calculated by

$$PI = \frac{1}{r} \cdot \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}} \cdot 100 \,[\%], \tag{10}$$

where n is the number of data points in the sample, y_i is the ith output value from the sample, and $\hat{y}_i$ is the ith output value calculated by the fuzzy system.

The overall performance indicator (PI_{ov}) of a fuzzy system takes into consideration the performance against both the training (PI_{tr}) and the test (PI_{te}) data sets in a weighted manner, where the weighting expresses the measure of the whole data set's coverage by the two samples. It is calculated by

$$PI_{ov} = \frac{n_{tr} \cdot PI_{tr} + n_{te} \cdot PI_{te}}{n_{tr} + n_{te}} \,[\%], \tag{11}$$

where n_{tr} and n_{te} are the number of data points in the training and test data sets, respectively.

3 Fuzzy Inference by FRISUV

The tuning of the fuzzy sets' parameters can produce a sparse rule base when the modification of the supports is enabled in course of the tuning. A rule base is characterized as sparse when there is at least one possible observation for which none of the rule's activation degree is greater than zero. The activation degree of a rule R_i [37] for an n-dimensional observation A^* is

$$\varpi_{h,t}(R_i) = s\left(t\left(A_{i1}, A_1^*\right), \ldots, t\left(A_{in}, A_n^*\right)\right), i = \overline{1, n_R}, \tag{12}$$

where s is an arbitrary s-norm, t is an arbitrary t-norm, A_{ij} is the antecedent set in the jth dimension of the ith rule, and n_R is the number of rules.

The traditional compositional fuzzy inference methods (e.g. Mamdani [25], Takagi-Sugeno [36], etc.) require a full coverage of the input space by rule antecedents. This demand cannot be fulfilled in sparse rule bases. The recognition of this shortcoming led to the emergence of inference techniques based on fuzzy rule interpolation (e.g. [4] [7] [13] [15] [17] [21] [22] [23] [26] [29]).

In the course of the experiments aimed at testing the new tuning method, the FRISUV [15] inference method was used, owing to its low computational complexity. The key idea of the fuzzy rule interpolation based on subsethood values is that it measures the similarity between the current observation and the rule antecedents, taking into consideration two factors: the shape similarity and the relative distance.

The shape similarity between the observation and the rule antecedent sets is calculated separately in each antecedent dimension by the means of the fuzzy subsethood value. First, the examined antecedent set is shifted into the position of the observation. Here the reference point of the fuzzy set is used for the definition of its position and for the calculation of distances between sets. The fuzzy subsethood value in case of the ith rule and the jth dimension is

$$FSV_{ij} = \frac{\sum_{x \in X_j} \mu_{A_j^* \cap A_{ij}}(x)}{\sum_{x \in X_j} \mu_{A_{ij}}(x)}, \tag{13}$$

where $\cap$ is an arbitrary t-norm, and X_j is the jth dimension of the input universe of discourse. The individual FSVs are aggregated by an average calculation

$$FSV_i = \frac{\sum_{j=1}^{n} FSV_{ij}}{n}. \tag{14}$$

The second aspect of the applied similarity measure is determined based on the Euclidean distance between the two points of the antecedent space defined by the

reference points of the fuzzy sets that describe the current observation and the reference points of the fuzzy sets that form the antecedent part of the current rule. It is a relative distance, defined by

$$d_i = \sqrt{\frac{\sum_{j=1}^{n}\left(RP(A_j^*) - RP\left(A_{ij}\right)\right)^2}{\sum_{j=1}^{n}\left(x_{j\max} - x_{j\min}\right)^2}}, \tag{15}$$

where *RP(.)* denotes the reference point of a fuzzy set, and x_{jmin} and x_{jmax} are the lower and upper bounds in the *j*th antecedent dimension, respectively. Finally, the similarity measure will be

$$S_i = \frac{FSV_i + 1 - d_i}{2}. \tag{16}$$

FRISUV calculates the position of the conclusion adapting the Shepard crisp interpolation [38]

$$RP\left(B^*\right) = \begin{cases} RP\left(B_i\right) & if\ S_i = 1, \\ \dfrac{\sum_{i=1}^{n_R}\dfrac{1}{1-S_i}\cdot RP\left(B_i\right)}{\sum_{i=1}^{n_R}\dfrac{1}{1-S_i}} & otherwise. \end{cases} \tag{17}$$

The method demands that all the sets of the consequent partition have the same shape. Thus the membership function of the conclusion will also share this feature.

4 Results

We performed tests of the hybrid algorithm on four benchmark problems. Three of them were real life problems, namely ground level ozone prediction [30], petrophysical properties prediction [41], yield strength prediction [1] [3], and the fourth was a synthetic function approximation problem. Testing was performed by executing one, two or three local search steps (n_p) after each five global search steps. Table 1 presents the test results. The number of data points (cardinality of the data samples) are summarized in Table 2. The overall performance indicator (PI_{ov}) values are contained in Table 3.

Table 1
Performance of the Systems Tuned by the CE Method compared to the Hybrid Method with one, two respectively three local search steps after each five global search steps

Dataset	CE Method		Hybrid Method with n_p=1		Hybrid Method with n_p=2		Hybrid Method with n_p=3	
	Train	Test	Train	Test	Train	Test	Train	Test
Ozone	14.6531	13.2386	14.5919	13.1057	14.6395	13.1737	14.4234	12.8965
Yield Strength	38.2629	36.1852	26.2513	15.3209	30.0461	22.0258	31.0267	24.3468
Well	27.4533	28.5658	14.8432	13.6063	14.9870	14.0165	14.4390	13.4913
Synthetic	19.6862	18.2116	19.0711	18.2106	17.5059	15.7902	18.8306	18.1833

Table 2
Number of data points in the training (n_{tr}) and test (n_{te}) data sets

Dataset	n_{tr}	n_{te}
Ozone	224	112
Yield Strength	310	90
Well	71	51
Synthetic	196	81

Table 3
Overall performance indicator (PI_{ov}) values

Dataset	CE Method	Hybrid Method		
		n_p=1	n_p=2	n_p=3
Ozone	14.1816	14.0965	14.1509	13.9144
Yield Strength	37.7954	23.7920	28.2415	29.5237
Well	27.9184	14.3261	14.5813	14.0428
Synthetic	19.2550	18.8104	17.0042	18.6413

The application of the Hybrid Method resulted in improvements compared to the usage of the CE method on all datasets. Examining the improvements separately for the case of train and test data samples we can summarize the followings.

In case of the train data samples the least improvement (0.09%) was encountered in case of the ozone data set and n_p=2, while n_p=3 in case of the well data set ensured the best improvement (47.41%). Although in two out of four cases n_p=3 led to a better result, surprisingly the average improvement (20.22%) was observed by n_p=1.

In the case of the test data samples, the improvement varied between 0.01% (synthetic data set and n_p=1) and 57.66% (yield strength data set and n_p=1). In the case of all the samples, the greatest improvement was found by the same local search number as in case of the train data sets. The greatest average improvement (27.76%) was observed by n_p=1.

Evaluating the results based on the overall performance indicator (PI_{ov}), we found a bit narrower variation interval for the improvement([0.60, 49.70]) with an overall average improvement of 20.85%. The greatest variation of PI_{ov}'s improvement due to n_p was 15.17%, in the case of the yield strength sample.

Conclusions

The test results show clearly that the Hybrid Method has great potential in parameter tuning, and the number of local search steps can have a significant influence on the achieved results.

Further research will concentrate on further adjusting the parameters of the presented method and examining the relation between some features of the modeled phenomena and the achieved improvement measure with the help of the Hybrid Method.

Acknowledgment

This research was partly supported by the Hungarian National Scientific Research Fund (Grant No. OTKA K77809) the Normative Application of R & D by Kecskemét College, GAMF Faculty (Grant No. 1KU31).

References

[1] Ádámné, A. M., Belina K.: Effect of Multiwall Nanotube on the Properties of Polypropylenes. Int. J. of Mater. Form., Vol. 1, No 1, 2008, pp. 591-594

[2] Alon, G., Kroese, D. P., Raviv T., Rubinstein, R. Y.: Application of the Cross-Entropy Method to the Buffer Allocation Problem in a Simulation-Based Environment. Ann. of Oper. Res., 2005

[3] Ádámné, A. M., Belina, K.: Investigation of PP and PC Carbon Nanotube Composites. 6th International Conference of PhD Students, Miskolc, 12-18 August 2007, pp. 1-6

[4] Baranyi. P., Kóczy, L. T., Gedeon. T. D.: A Generalized Concept for Fuzzy Rule Interpolation. in IEEE Trans. on Fuzzy Syst., Vol. 12, No. 6, 2004, pp 820-837

[5] Bezdek, J. C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York, 1981

[6] Botzheim, J., Hámori, B., Kóczy, L. T.: Extracting Trapezoidal Membership Functions of a Fuzzy Rule System by Bacterial Algorithm, 7th Fuzzy Days, Dortmund, Springer-Verlag, 2001, pp. 218-227

[7] Chen, S. M., Ko, Y. K.: Fuzzy Interpolative Reasoning for Sparse Fuzzy Rule-based Systems Based on α-cuts and Transformations Techniques, IEEE Trans. on Fuzzy Syst, Vol. 16, No. 6, 2008, pp. 1626-1648

[8] Chong, A., Gedeon, T. D., Kóczy L. T.: Projection-based Method for Sparse Fuzzy System Generation. in Proceedings of the 2nd WSEAS International Conference on Scientific Computation and Soft Computing, Crete, Greece, 2002, pp. 321-325

[9] Devasenapati, S. B., Ramachandran, K. I.: Hybrid Fuzzy Model-based Expert System for Misfire Detection in Automobile Engines, International Journal of Artificial Intelligence, Vol. 7, No. A11, 2011, pp. 47-62

[10] Helvik, B. E., Wittner, O.: Using the Cross-Entropy Method to Guide / Govern Mobile Agent's Path Finding in Networks, Lect. Notes in Comp. Sci., 2164/2001, pp. 255-268

[11] Hládek, D., Vaščák, J., Sinčák, P.: Hierarchical Fuzzy Inference System for Robotic Pursuit Evasion Task. in Proceedings of SAMI 2008, 6th International Symposium on Applied Machine Intelligence and Informatics, January 21-22, 2008, Herľany, Slovakia, pp. 273-277

[12] Homem-de-Mello, T., Rubinstein, R. Y.: Estimation of Rare Event Probabilities using Cross-Entropy, WSC 1, 2002, pp. 310-319

[13] Huang, Z. H., Shen, Q.: Fuzzy Interpolation with Generalized Representative Values. In Proceedings of the UK Workshop on Computational Intelligence, 2004, pp. 161-171

[14] Joelianto, E., Widiyantoro, S., Ichsan, M.: Time Series Estimation on Earthquake Events using ANFIS with Mapping Function, International Journal of Artificial Intelligence, Vol. 3, No. A09, 2009, pp. 37-63

[15] Johanyák, Zs. Cs.: Fuzzy Rule Interpolation Based on Subsethood Values. in Proceedings of 2010 IEEE Interenational Conference on Systems Man. and Cybernetics (SMC 2010) 2010, pp. 2387-2393

[16] Johanyák, Zs. Cs.: Sparse Fuzzy Model Identification Matlab Toolbox - RuleMaker Toolbox. IEEE 6th International Conference on Computational Cybernetics, November 27-29, 2008, Stara Lesná, Slovakia, pp. 69-74

[17] Johanyák, Zs. Cs.: Performance Improvement of the Fuzzy Rule Interpolation Method LESFRI, in Proceeding of the 12th IEEE International Symposium on Computational Intelligence and Informatics, Budapest, Hungary, November 21-22, 2011, pp. 271-276

[18] Johanyák, Zs. Cs., Ádámné, A.M.: Mechanical Properties Prediction of Thermoplastic Composites using Fuzzy Models. Scientific Bulletin of "Politehnica" University of Timisoara. Romania. Transactions on Automatic Control and Computer Science, Vol: 54(68), No: 4/2009, pp. 185-190

[19] Johanyák, Zs. Cs., Kovács, S.: Sparse Fuzzy System Generation by Rule Base Extension. in Proceedings of the 11th IEEE International Conference of Intelligent Engineering Systems (IEEE INES 2007) Budapest, Hungary, pp. 99-104

[20] Johanyák, Zs. Cs., Papp, O.: Comparative Analysis of Two Fuzzy Rule Base Optimization Methods. 6th IEEE International Symposium on Applied

Computational Intelligence and Informatics (SACI 2011) 19-21 May 2011, Timisoara, Romania, pp. 235-240

[21] Kóczy, L. T., Hirota, K.: Approximate Reasoning by Linear Rule Interpolation and General Approximation. in International Journal of Approximative Reasoning, Vol. 9, 1993, pp. 197-225

[22] Kovács, L.: Rule Approximation in Metric Spaces. Proceedings of 8th IEEE International Symposium on Applied Machine Intelligence and Informatics SAMI 2010, Herl'any, Slovakia, 2010, pp. 49-52

[23] Kovács, S.: Extending the Fuzzy Rule Interpolation "FIVE" by Fuzzy Observation. Advances in Soft Computing. Computational Intelligence, Theory and Applications, Bernd Reusch (Ed.) Springer Germany, 2006, pp. 485-497

[24] Kovács, S., Kóczy, L. T.: Application of Interpolation-based Fuzzy Logic Reasoning in Behaviour-based Control Structures. Proceedings of the FUZZIEEE. IEEE International Conference on Fuzzy Systems, 25-29 July 2004, Budapest, Hungary, p. 6

[25] Mamdani, E. H., Assilian, S.: An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller, in International Journal of Man Machine Studies, Vol. 7, 1975, pp. 1-13

[26] Detyniecki, M., Marsala, C., Rifqi, M.: Double-Linear Fuzzy Interpolation Method, IEEE International Conference on Fuzzy Systems (FUZZ 2011), Taipei, Taiwan, Jun 27-30, 2011, pp. 455-462

[27] Menache, I., Mannor, S., Shimkin, N.: Basis Function Adaptation in Temporal Difference Reinforcement Learning, Ann. of Oper. Res., 134, 2005, pp. 215-238

[28] Palanisamy, C., Selvan, S.: Wavelet-based Fuzzy Clustering of Higher Dimensional Data, International Journal of Artificial Intelligence, Vol. 2, No. S09, 2009, pp: 27-36

[29] Perfilieva, I., Wrublova, M. Hodakova, P.: Fuzzy Interpolation According to Fuzzy and Classical Conditions, Acta Polytechnica Hungarica, Vol. 7, Issue 4, Special Issue: SI, 2010, pp. 39-55

[30] Pires, J. C. M., Martins F. G., Pereira M. C., Alvim-Ferraz M. C. M.: Prediction of Ground-Level Ozone Concentrations through Statistical Models. in Proceedings of IJCCI 2009 - International Joint Conference on Computational Intelligence, 5-7 October 2009 Funchal-Madeira, Portugal, pp. 551-554

[31] Portik, T., Pokorádi, L.: Possibility of Use of Fuzzy Logic in Management. 16th Building Services. Mechanical and Building Industry days" International Conference, 14-15 October 2010, Debrecen, Hungary, pp. 353-360

[32] Precup, R. E., Doboli, S., Preitl, S.: Stability Analysis and Development of a Class of Fuzzy Systems, Eng. Appl. of Artif. Int., Vol. 13, No. 3, June 2000, pp. 237-247

[33] Precup, R.-E., Preitl, S.: Optimisation Criteria in Development of Fuzzy Controllers with Dynamics. Eng. Appl. of Artif. Intell., Vol. 17, No. 6, 2004, pp. 661-674

[34] Rubinstein, R. Y.: The Cross-Entropy Method for Combinatorial and Continuous Optimization, Methodol. and Comput. in Appl. Probab, 1999

[35] Škrjanc, I., Blažič, S., Agamennoni, O. E.: Identification of Dynamical Systems with a Robust Interval Fuzzy Model, Automatica, 2005, Vol. 41, No. 2, pp. 327-332

[36] Takagi, T. and Sugeno, M.: Fuzzy Identification of Systems and its Applications to Modeling and Control, in IEEE Transactions on System, Man and Cybernetics, Vol. 15, 1985, pp. 116-132

[37] Tikk, D., Johanyák, Zs. Cs., Kovács, S., Wong, K. W.: Fuzzy Rule Interpolation and Extrapolation Techniques: Criteria and Evaluation Guidelines, Journal of Advanced Computational Intelligence and Intelligent Informatics, ISSN 1343-0130, Vol. 15, No. 3, 2011, pp. 254-263

[38] Shepard, D.: A Two Dimensional Interpolation Function for Irregularly Spaced Data. In Proceedings of the 23rd ACM International Conference, 1968, New York, USA, pp. 517-524

[39] Vincze, D., Kovács, S.: Incremental Rule Base Creation with Fuzzy Rule Interpolation-based Q-Learning. Studies in Computational Intelligence - Computational Intelligence in Engineering, Vol. 313, 2010, pp. 191-203

[40] Wang, W., Zhang, Y.: On Cluster Validity Indices. Fuzzy Sets and Systems, 158, 2007, pp. 2095-2117

[41] Wong, K. W., Gedeon, T. D.: Petrophysical Properties Prediction Using Self-generating Fuzzy Rules Inference System with Modified Alpha-Cut-based Fuzzy Interpolation. Proceedings of the Seventh International Conference of Neural Information Processing ICONIP 2000, November 2000, Korea, pp. 1088-1092

[42] Zemková, B., Talašová, J.: Fuzzy Sets in HR Management. Acta Polytechnica Hungarica, Vol. 8, No. 3, 2011, pp. 113-124