New Method for Design of Fuzzy Systems for Nonlinear Modelling Using Different Criteria of Interpretability

Krystian Łapa[1], Krzysztof Cpałka[1], and Lipo Wang[2]

[1] Częstochowa University of Technology,
Institute of Computational Intelligence, Poland
{krystian.lapa,krzysztof.cpalka}@iisi.pcz.pl
[2] Nanyang Technological University,
School of Electrical & Electronic Engineering, Singapore
elpwang@ntu.edu.sg

Abstract. In this paper a new method for designing neuro-fuzzy systems for nonlinear modelling is proposed. This method contains a complex weighted fitness function with interpretability criteria and new enhanced tuning process for selecting parameters and structure of the system based on a hybrid population-based algorithm (composed of evolutionary strategy, genetic algorithm and bees algorithm). To evaluate this method, we used a well-known dynamic nonlinear modelling problem.

1 Introduction

The analysis of technical issues aims at finding and understanding the essence of the problem, it tries to create a model. The reason for this is the willingness to ensure predictability, which guarantees safety, decreases costs and ensures control. In the literature the following approaches to modelling are considered: **(a) White-box model.** This approach uses phenomenological (theoretical) description of physical phenomena. For more details, see e.g. [9], [40]. **(b) Black-box model.** In this approach the behaviour of the object is recreated on the basis of observations of cause and effect of dependencies. For more details, see e.g. [19]-[20], [25]-[26], [35]-[36], [51], [56]. **(c) Grey-box model.** This approach is based on model structure derived from some laws and parameters tuned to the data defining behaviour of the object. These methods include, among others, multivariable non-stationary systems, hybrid solutions and systems of computational intelligence such as fuzzy systems, neuro-fuzzy systems etc. For more details, see e.g. [6], [18], [21], [32]-[34], [40]-[41], [43]-[44].

There is still a search for such nonlinear modelling methods which will be characterized by a good accuracy and possibility to interpret the knowledge accumulated within it. The interpretability issue in the context of nonlinear modelling is much harder than in case of classification (in the system the exact value of the output signal is important). Each limitation put upon system structure (used to increase the interpretability) has pronounced negative effect

L. Rutkowski et al. (Eds.): ICAISC 2014, Part I, LNAI 8467, pp. 217–232, 2014.

on the accuracy (and vice versa). In the literature there are different approaches to increase the interpretability of fuzzy systems. It can be noted that these approaches are mainly based on a suitable structure of the fuzzy system (e.g. [1]-[5], [10]-[13], [22]-[23], [27], [48], [52]-[55]) or on the use of specific training algorithm (e.g. methods in the field of multiobjective optimization or evolutionary optimization) (see e.g. [14], [17], [28]-[29], [38], [57]). In this paper we propose a new method for designing neuro-fuzzy systems for nonlinear modelling (see e.g. [30]). This method can be described as follows: (1) the parameters and also the structure of the neuro-fuzzy systems are obtained in the learning process, (2) the learning process takes into consideration accuracy of the system, complexity of the system, and interpretability criteria, (3) presented criteria allow to obtain clear and well-spread semantic of the rules of the system, (4) learning process was based on hybrid population algorithm composed of evolutionary strategy (μ, λ) (see e.g. [15]) and bees algorithm (see e.g. [37]). To evaluate performance of our method we used well-known dynamic nonlinear modelling problem - Van der Pol oscillator problem.

This paper is organized into 5 sections. Section 2 contains description of the fuzzy system for nonlinear modelling. Description of the new method for designing our system is given in Section 3. Simulation results are presented in Section 4. Conclusions are drawn in Section 5.

2 Description of the Fuzzy System for Nonlinear Modelling

In our previous works we considered a new class of the neuro-fuzzy systems (see e.g. [24], [47], [49]-[50]) - the flexible neuro-fuzzy systems (see [42], [59]-[62]). Those systems have very high accuracy in the field on classification and approximation problems. We consider multi-input, multi-output neuro-fuzzy system mapping $\mathbf{X} \to \mathbf{Y}$, where $\mathbf{X} \subset \mathbf{R}^n$ and $\mathbf{Y} \subset \mathbf{R}^m$. The flexible fuzzy rule base consists of a collection of N fuzzy IF-THEN rules in the form

$$R^k : \left[\left(\begin{array}{c} \text{IF } \left(\bar{x}_1 \, \text{is} \, A_1^k\right) \left| w_{k,1}^A \right. \text{AND} \ldots \text{AND} \left(\bar{x}_n \, \text{is} \, A_n^k\right) \left| w_{k,n}^A \right. \\ \text{THEN } \left(y_1 \, \text{is} \, B_1^k\right), \ldots, \left(y_m \, \text{is} \, B_m^k\right) \end{array} \right) \left| w_k^{\text{rule}} \right. \right], \tag{1}$$

where $\bar{\mathbf{x}} = [\bar{x}_1, \ldots, \bar{x}_n] \in \mathbf{X}$, $\mathbf{y} = [y_1, \ldots, y_m] \in \mathbf{Y}$, $A_1^k, \ldots, A_n^k$ are fuzzy sets characterized by membership functions $\mu_{A_i^k}(x_i)$, $i = 1, \ldots, n$, $k = 1, \ldots, N$, $B_1^k, \ldots, B_m^k$ are fuzzy sets characterized by membership functions $\mu_{B_j^k}(y_j)$, $j = 1, \ldots, m$, $k = 1, \ldots, N$, $w_{k,i}^A \in [0, 1]$, $i = 1, \ldots, n$, $k = 1, \ldots, N$, are weights of antecedents, $w_k^{\text{rule}} \in [0, 1]$, $k = 1, \ldots, N$, are weights of rules. In Mamdani approach output signal $\bar{y}_j$, $j = 1, \ldots, m$ of the neuro-fuzzy system is described by the formula (for more details see our previous papers, e.g. [45]-[46])

$$\bar{y}_j = \frac{\sum\limits_{r=1}^{R} \bar{y}^{\mathrm{def}}_{j,r} \cdot \mathop{S^*}\limits_{k=1}^{N} \left\{ T \left\{ \mathop{T^*}\limits_{i=1}^{n} \left\{ \mu_{A_i^k}\left(\bar{x}_i\right); w^{A}_{k,i} \right\}, \mu_{B_j^k}\left(\bar{y}^{\mathrm{def}}_{j,r}\right) \right\}; w^{\mathrm{rule}}_{k} \right\}}{\sum\limits_{r=1}^{R} \mathop{S^*}\limits_{k=1}^{N} \left\{ T \left\{ \mathop{T^*}\limits_{i=1}^{n} \left\{ \mu_{A_i^k}\left(\bar{x}_i\right); w^{A}_{k,i} \right\}, \mu_{B_j^k}\left(\bar{y}^{\mathrm{def}}_{j,r}\right) \right\}; w^{\mathrm{rule}}_{k} \right\}}, \tag{2}$$

where $\bar{y}^{\mathrm{def}}_{j,r}$, $j = 1, \ldots, m$, $r = 1, \ldots, R$, are discretization points, R is a number of discretization points.

In the next section a new learning algorithm for evolution of flexible neuro-fuzzy system (2) is proposed. The aim of the algorithm is the selection of the parameters and structure of the neuro-fuzzy system for nonlinear modelling described by equation (2) with the accuracy and interpretability taken into consideration. In the process of evolution (evolution of parameters) we will find all parameters of the neuro-fuzzy system (2). Moreover, in the process of evolution (evolution of the structure) we will find number of inputs n, number of rules N, number of antecedents and consequents (number of fuzzy sets) and number of discretization points R.

In the next section we are going to show the use of a new population based algorithm used to select the structure and parameters of system (2) with the accuracy and interpretability taken into consideration.

3 Description of the New Evolutionary Approach to Choice of the System Structure and Parameters for Nonlinear Modelling

As mentioned before, for selection of structure and parameters of system (2) we have proposed a new evolutionary algorithm. The algorithm is based on the Pittsburgh approach ([31], [42]), on the evolutionary strategy (μ, λ) for selecting parameters of system (2), on the classical genetic algorithm for choosing structure of system (2) and on the bees algorithm for fixing parameters of reduced systems (2). The evolutionary strategy (μ, λ) starts with a random generation of the initial parents population $\mathbf{P}$ containing μ individuals. Next, a temporary population $\mathbf{T}$ is created by means of reproduction, whose population contains λ individuals, while $\lambda \geq \mu$. Reproduction consists in a multiple random selection of λ individuals out of the population $\mathbf{P}$ (multiple sampling) and placing the selected ones in temporary population $\mathbf{T}$. Individuals of the population $\mathbf{T}$ undergo crossover and mutation operations as a result of which an offspring population $\mathbf{O}$ is created, which also has size λ. The purpose of the repair procedure of the population $\mathbf{O}$ is to correct the parameters if they reach inadmissible values. The new population $\mathbf{P}$ containing μ individuals is selected only out of the best λ individuals of the population $\mathbf{O}$. The bees algorithm mimics the food foraging behaviour of honey bee colonies and it is used to tuning parameters of system (2). The aim of using this algorithm is to tune the parameters of the systems with recently reduced structure and to repair damaged accuracy. The behaviour of the bees can be described as follows: (1) For every μ population chromosomes of **P** and μ chromosomes are generated (scout bees), (2) For every chromosome of **B** a search territory area is calculated (as an area of solution explorations coded in the population **P**). For every iteration of the algorithm the area of exploration is decreased, (3) After this modification, chromosomes from **B** are repaired and evaluated (analogically to evolutionary strategy (μ, λ)), (4) In the last step one solution (with best fitness function value) is picked from each group of scout bees and moved into the population **P**. More details about bees algorithm can be seen in [37].

3.1 Coding of Parameters and Structure

The parameters of system (2) were coded in the following chromosome (Pittsburgh approach)

$$\mathbf{X}_{ch}^{\text{par}} = \left\{ \begin{array}{c} \bar{x}_{1,1}^{A}, \sigma_{1,1}^{A}, \ldots, \bar{x}_{n,1}^{A}, \sigma_{n,1}^{A}, \ldots \\ \bar{x}_{1,Nmax}^{A}, \sigma_{1,Nmax}^{A}, \ldots, \bar{x}_{n,Nmax}^{A}, \sigma_{n,Nmax}^{A}, \\ \bar{y}_{1,1}^{B}, \sigma_{1,1}^{B}, \ldots, \bar{y}_{m,1}^{B}, \sigma_{m,1}^{B}, \ldots \\ \bar{y}_{1,Nmax}^{B}, \sigma_{1,Nmax}^{B}, \ldots, \bar{y}_{m,Nmax}^{B}, \sigma_{m,Nmax}^{B}, \\ w_{1,1}^{A}, \ldots, w_{n,1}^{A}, \ldots, w_{1,Nmax}^{A}, \ldots, w_{n,Nmax}^{A}, \\ w_{1}^{\text{rule}}, \ldots, w_{Nmax}^{\text{rule}}, \\ \bar{y}_{1,1}^{\text{def}}, \ldots, \bar{y}_{1,Rmax}^{\text{def}}, \ldots, \bar{y}_{m,1}^{\text{def}}, \ldots, \bar{y}_{m,Rmax}^{\text{def}} \end{array} \right\} = \left\{ X_{ch,1}^{\text{par}}, \ldots, X_{ch,L}^{\text{par}} \right\}, \tag{3}$$

where $L = Nmax \cdot (3 \cdot n + 2 \cdot m + 1) + Rmax$, $ch = 1, \ldots, \mu$ for the parent population or $ch = 1, \ldots, \lambda$ for the temporary population, $Nmax$ is the maximum number of rules, $Rmax$ is the maximum number of discretization points. The maximum number of rules $Nmax$ should be selected individually to the problem from the range $[1, Nmax]$. Analogously, the maximum number of discretization points $Rmax$ should also be selected to the problem individually from the range $[1, Rmax]$ ([7]). The purpose of the algorithm is also to select the number of antecedents (from the range $[1, n]$) and consequents (from the range $[1, m]$) within each rule from rule base. The reduction of the system is done with the use of additional chromosome $\mathbf{X}_{ch}^{\text{red}}$. Its genes take binary values and indicate which rules, antecedents, consequents, inputs, and discretization points are selected. The chromosome $\mathbf{X}_{ch}^{\text{red}}$ is given by

$$\mathbf{X}_{ch}^{\text{red}} = \left\{ \begin{array}{c} x_1, \ldots, x_n, \\ A_1^1, ..., A_n^1, ..., A_1^{Nmax}, ..., A_n^{Nmax}, \\ B_1^1, ..., B_m^1, ..., B_1^{Nmax}, ..., B_m^{Nmax}, \\ \text{rule}_1, ..., \text{rule}_{Nmax}, \\ \bar{y}_{1,1}^{\text{def}}, ..., \bar{y}_{1,Rmax}^{\text{def}}, ..., \bar{y}_{m,1}^{\text{def}}, ..., \bar{y}_{m,Rmax}^{\text{def}} \end{array} \right\} = \left\{ X_{ch,1}^{\text{red}}, ..., X_{ch,L^{\text{red}}}^{\text{red}} \right\}, \tag{4}$$

where $L^{\text{red}} = Nmax \cdot (n + m + 1) + n + m \cdot Rmax$ is the length of the chromosome $\mathbf{X}_{ch}^{\text{red}}$, $ch = 1, \ldots, \mu$, for the parent population or $ch = 1, \ldots, \lambda$, for the temporary population. Its genes indicate which rules (rule_k, $k = 1, \ldots, Nmax$), antecedents (A_i^k, $i = 1, \ldots, n$, $k = 1, \ldots, Nmax$), consequents ($B_j^k, j = 1, ..., m, k = 1, ..., Nmax$), inputs ($\bar{x}_i$, $i = 1, \ldots, n$), and discretization points ($\bar{y}^r$, $r = 1, \ldots, Rmax$) are taken to the system. We can easily notice that the number of inputs used in the system encoded in the chromosome *ch* can be determined as follows

$$n_{ch} = \sum_{i=1}^{n} \mathbf{X}_{ch}^{\text{red}} \{x_i\}, \tag{5}$$

where $\mathbf{X}_{ch}^{\text{red}} \{x_i\}$ means gene of the chromosome $\mathbf{X}_{ch}^{\text{red}}$ associated with the input x_i (as previously mentioned, if the value of the gene is 1, the associated input is taken into account during work of the system). The number of rules (N_{ch}) used in the system encoded in the chromosome *ch* may be determined analogously. Implementation of the strategy (μ, λ) uses an additional chromosome

$$\sigma_{ch}^{\text{par}} = \left(\sigma_{ch,1}^{\text{par}}, \ldots, \sigma_{ch,L}^{\text{par}} \right), \tag{6}$$

where $ch = 1, \ldots, \mu$ for the parent population or $ch = 1, \ldots, \lambda$ for the temporary population. This allows the implementation of the mechanism of self-adaptive range of mutation. At the beginning of the operation of evolutionary strategy the range is large, while during the convergence its gradual reduction is observed. This results in a smooth transition from exploration (occurring at the beginning of the algorithm) to exploitation of the promising areas.

3.2 Evolution of Parameters and Structure

This hybrid population-based method allows for tuning both structure and parameters of system (2) with interpretability criteria. It is worth mentioning that: **(a)** An evolutionary strategy (μ, λ) was used for tuning the parameters of system (2). It processes chromosomes $\mathbf{X}_{ch}^{\text{par}}$ i σ_{ch}^{par} from the population **P**, **T** and **O**. The details about crossover and mutation operators from this strategy can be found in [42]. **(b)** For the structure evolution of system (2), a classic genetic algorithm was chosen. It processes chromosomes $\mathbf{X}_{ch}^{\text{red}}$ from the population **P**, **T** and **O**. The details about crossover and mutation operators from this strategy can be found in [31]. It is important to mention that genetic algorithm works together with evolutionary strategy (μ, λ), and it allows to reduce any element of the system structure, such like antecedence, consequences, inputs, rules and discretization points. **(c)** For tuning parameters of system (2), a bees algorithm was additionally used. It processes chromosomes $\mathbf{X}_{ch}^{\text{par}}$ from the population **B**. The purpose of use the bees algorithm is to search neighbourhood around chromosomes from population **B** (chromosomes with reduced structure of the system) and replace them with fitter solutions. The details about bees algorithm can be found in [37]. **(d)** The important mechanism of our method is a process of evaluation of the chromosomes from the populations **P**, **T**, **O** and **B** described in Section 3.3. It takes into consideration an accuracy-interpretability trade-off and allows to obtain a balanced dependent from weights of the fitness function components solutions (see e.g. [16], [58]).

3.3 Chromosome Population Evaluation

Each individual $\mathbf{X}_{ch}$ of the parental and temporary populations is represented by sequence of chromosomes $\langle \mathbf{X}_{ch}^{\text{par}}, \sigma_{ch}^{\text{par}}, \mathbf{X}_{ch}^{\text{red}} \rangle$, given by formulas (3), (4) and (6). The genes of the two first chromosomes take real values, whereas the genes of the last chromosome takes integer values from the set $\{0,1\}$. The system aims to minimize the following fitness function

$$\text{ff}\left(\mathbf{X}_{ch}\right) = T^{*}\left\{ \begin{array}{c} \text{ffaccuracy}\left(\mathbf{X}_{ch}\right), \text{ffcomplexity}\left(\mathbf{X}_{ch}\right), \text{ffinterpretability}\left(\mathbf{X}_{ch}\right); \\ w_{\text{ffaccuracy}}, w_{\text{ffcomplexity}}, w_{\text{ffinterpretability}} \end{array} \right\}, \tag{7}$$

where $T^{*}\{\cdot\}$ is the algebraic weighted t-norm (see e.g. [8]), $w_{\text{ffaccuracy}} \in (0,1]$ denoted weight of the component $\text{ffaccuracy}\left(\mathbf{X}_{ch}\right)$ etc. The individual components of the $\text{ff}\left(\mathbf{X}_{ch}\right)$ are defined as follows:

The component $\text{ffaccuracy}\left(\mathbf{X}_{ch}\right)$ determines the accuracy of system (2) i.e. average normalized system error for all outputs and all data from learning sequence

$$\text{ffaccuracy}\left(\mathbf{X}_{ch}\right) = \frac{1}{m_{ch}} \sum_{j=1}^{m_{ch}} \frac{\frac{1}{Z} \sum\limits_{z=1}^{Z} \left| d_{z,j} - \bar{y}_{z,j} \right|}{\max\limits_{z=1,\ldots,Z} \left\{ d_{z,j} \right\} - \min\limits_{z=1,\ldots,Z} \left\{ d_{z,j} \right\}}, \tag{8}$$

where m_{ch} is a number of outputs encoded in the chromosome ch, Z is the number of samples of learning sequence, $d_{z,j}$ is desired value of output signal $j = 1, ..., m$ for input vector $z = 1, ..., Z$, $\bar{y}_{z,j}$ is real value of the output signal $j = 1, ..., m$ for input vector $z = 1, ..., Z$. The purpose of the normalization of the component $\text{ffaccuracy}\left(\mathbf{X}_{ch}\right)$ was to ensure an influence on every component of the function (7).

The component $\text{ffcomplexity}\left(\mathbf{X}_{ch}\right)$ determines complexity of system (2) i.e. a number of reduced elements of the system (rules, antecedents- input fuzzy sets, consequents- output fuzzy sets, inputs, and discretization points) in relation to length of the chromosome $\mathbf{X}_{\text{ch}}^{\text{red}}$ (it allows to increase complexity-based interpretability)

$$\text{ffcomplexity}\left(\mathbf{X}_{ch}\right) = \frac{\left(\begin{array}{c} \sum\limits_{i=1}^{n} \mathbf{X}_{ch}^{\text{red}}\left\{x_i\right\} \cdot \sum\limits_{k=1}^{Nmax} \mathbf{X}_{ch}^{\text{red}}\left\{\text{rule}_k\right\} \cdot \mathbf{X}_{ch}^{\text{red}}\left\{A_i^k\right\} + \\ + \sum\limits_{j=1}^{m} \sum\limits_{k=1}^{Nmax} \mathbf{X}_{ch}^{\text{red}}\left\{\text{rule}_k\right\} \cdot \mathbf{X}_{ch}^{\text{red}}\left\{B_j^k\right\} + \\ + \sum\limits_{j=1}^{m} \sum\limits_{r=1}^{Rmax} \mathbf{X}_{ch}^{\text{red}}\left\{\bar{y}_{m,r}^{\text{def}}\right\} \end{array} \right)}{N_{ch} \cdot \left(n_{ch} + m\right) + m \cdot Rmax}, \tag{9}$$

where n is a number of inputs, m is a number of outputs, $Rmax$ is maximum number of discretization points, $Nmax$ is maximum number of rules, $\mathbf{X}_{ch}^{\text{red}}\left\{x_i\right\}$ means a gene of the chromosome $\mathbf{X}_{ch}^{\text{red}}$ associated with the input x_i, etc.

The component $\text{ffinterpretability}\left(\mathbf{X}_{ch}\right)$ determines the semantic interpretability of system (2) encoded in the tested chromosome (it allows to increase semantic-based interpretability)

$$\text{ffinterpretability}\left(\mathbf{X}_{ch}\right) = T^{*}\left\{\begin{array}{c} \text{ffint}_{A}\left(\mathbf{X}_{ch}\right), \text{ffint}_{B}\left(\mathbf{X}_{ch}\right), \text{ffint}_{C}\left(\mathbf{X}_{ch}\right), \\ \text{ffint}_{D}\left(\mathbf{X}_{ch}\right), \text{ffint}_{E}\left(\mathbf{X}_{ch}\right), \text{ffint}_{F}\left(\mathbf{X}_{ch}\right); \\ w_{\text{ffintA}}, w_{\text{ffintB}}, w_{\text{ffintC}}, w_{\text{ffintD}}, w_{\text{ffintE}}, w_{\text{ffintF}} \end{array}\right\}, \tag{10}$$

where $w_{\text{ffintA}} \in (0,1]$ denotes weight of the component $\text{ffint}_{A}\left(\mathbf{X}_{ch}\right)$, etc. The individual components of the formula (10) are defined as follows:

(a) The component $\text{ffint}_{A}\left(\mathbf{X}_{ch}\right)$ minimizes number of rules fired at the same time in system (2) for the fuzzy sets

$$\text{ffint}_{A}\left(\mathbf{X}_{ch}\right) = 1 - \frac{1}{Z}\sum_{z=1}^{Z} \frac{\left(\max\limits_{k=1,\ldots,Nmax}\left\{\mathbf{X}_{ch}^{\text{red}}\left\{\text{rule}_{k}\right\} \cdot \tau_{k}\left(\bar{\mathbf{x}}_{z}\right)\right\}\right)^{2}}{\sum\limits_{k=1}^{Nmax} \mathbf{X}_{ch}^{\text{red}}\left\{\text{rule}_{k}\right\} \cdot \tau_{k}\left(\bar{\mathbf{x}}_{z}\right)}, \tag{11}$$

where $\tau_{k}\left(\bar{\mathbf{x}}_{z}\right)$ is the flexible firing strength of the k-th rule, $\bar{\mathbf{x}}_{z}$ is a vector of input signals learning sequence $(z = 1, ..., Z)$.

(b) The component $\text{ffint}_{B}\left(\mathbf{X}_{ch}\right)$ maximizes the fit to the training data of input fuzzy sets of system (2) encoded in the tested chromosome

$$\text{ffint}_{B}\left(\mathbf{X}_{ch}\right) = \frac{\sum\limits_{z=1}^{Z}\sum\limits_{i=1}^{n} \mathbf{X}_{ch}^{\text{red}}\left\{x_{i}\right\} \cdot \left(1 - \max\limits_{k=1,\ldots,Nmax}\left\{\mathbf{X}_{ch}^{\text{red}}\left\{\text{rule}_{k}\right\} \cdot \mu_{A_{i}^{k}}\left(\bar{x}_{z,i}\right)\right\}\right)}{Z \cdot n_{ch}}, \tag{12}$$

where $\mu_{A_{i}^{k}}\left(\bar{x}_{z,i}\right)$ is a membership function of the input fuzzy set A_{i}^{k}, $\bar{x}_{z,i}$ is a real value of the input signal $i = 1, ..., n$ of the input vector $\bar{\mathbf{x}}_{z}$, $z = 1, ..., Z$.

(c) The component $\text{ffint}_{C}\left(\mathbf{X}_{ch}\right)$ reduces the overlapping of the input and output fuzzy sets of system (2) encoded in the tested chromosome

$$\text{ffint}_{C}\left(\mathbf{X}_{ch}\right) = \frac{1}{4} \cdot \left(\frac{\sum\limits_{i=1}^{n_{ch}} \sum\limits_{k=1}^{\text{noifs}(i)-1} \left(\begin{array}{l} \left| c_{\text{ffintc}} - \exp\left(-\left(\frac{\mathbf{X}_{ch}^{\text{supp}}\left\{\bar{x}_{i,k}^{A}\right\} - \mathbf{X}_{ch}^{\text{supp}}\left\{\bar{x}_{i,k+1}^{A}\right\}}{\mathbf{X}_{ch}^{\text{supp}}\left\{\sigma_{i,k}^{A}\right\} + \mathbf{X}_{ch}^{\text{supp}}\left\{\sigma_{i,k+1}^{A}\right\}}\right)^{2}\right)\right| + \\ + \left| - \exp\left(-\left(\frac{\mathbf{X}_{ch}^{\text{supp}}\left\{\bar{x}_{i,k}^{A}\right\} - \mathbf{X}_{ch}^{\text{supp}}\left\{\bar{x}_{i,k+1}^{A}\right\}}{\mathbf{X}_{ch}^{\text{supp}}\left\{\sigma_{i,k}^{A}\right\} - \mathbf{X}_{ch}^{\text{supp}}\left\{\sigma_{i,k+1}^{A}\right\}}\right)^{2}\right)\right| \end{array} \right)}{\sum\limits_{i=1}^{n_{ch}} \left(\text{noifs}(i) - 1\right)} + \right. \\ \left. + \frac{\sum\limits_{j=1}^{m} \sum\limits_{k=1}^{\text{noofs}(j)-1} \left(\begin{array}{l} \left| c_{\text{ffintc}} - \exp\left(-\left(\frac{\mathbf{X}_{ch}^{\text{supp}}\left\{\bar{y}_{j,k}^{B}\right\} - \mathbf{X}_{ch}^{\text{supp}}\left\{\bar{y}_{j,k+1}^{B}\right\}}{\mathbf{X}_{ch}^{\text{supp}}\left\{\sigma_{j,k}^{B}\right\} + \mathbf{X}_{ch}^{\text{supp}}\left\{\sigma_{j,k+1}^{B}\right\}}\right)^{2}\right)\right| + \\ + \left| - \exp\left(-\left(\frac{\mathbf{X}_{ch}^{\text{supp}}\left\{\bar{y}_{j,k}^{B}\right\} - \mathbf{X}_{ch}^{\text{supp}}\left\{\bar{y}_{j,k+1}^{B}\right\}}{\mathbf{X}_{ch}^{\text{supp}}\left\{\sigma_{j,k}^{B}\right\} - \mathbf{X}_{ch}^{\text{supp}}\left\{\sigma_{j,k+1}^{B}\right\}}\right)^{2}\right)\right| \end{array} \right)}{\sum\limits_{j=1}^{m} \left(\text{noofs}(j) - 1\right)} \right), \tag{13}$$

where $\mathbf{X}_{ch}^{\mathrm{supp}}$ stands for additional chromosome with list of non-reduced fuzzy sets

$$\mathbf{X}_{ch}^{\mathrm{supp}} = \left\{ \begin{array}{c} \bar{x}^A_{1,1}, \sigma^A_{1,1}, \bar{x}^A_{1,2}, \sigma^A_{1,2}, \ldots, \\ \bar{x}^A_{n_{ch},1}, \sigma^A_{n_{ch},1}, \bar{x}^A_{n_{ch},2}, \sigma^A_{n_{ch},2}, \ldots, \\ \bar{y}^B_{1,1}, \sigma^B_{1,1}, \bar{y}^B_{2,N_{ch}}, \sigma^B_{2,N_{ch}}, \ldots, \\ \bar{y}^B_{m,N_{ch}}, \sigma^B_{m,N_{ch}}, \bar{y}^B_{2,N_{ch}}, \sigma^B_{2,N_{ch}}, \ldots \end{array} \right\} = \left\{ X^{\mathrm{supp}}_{ch,1}, \ldots, X^{\mathrm{supp}}_{ch,L^{\mathrm{supp}}} \right\}, \quad (14)$$

where $L^{\mathrm{supp}} = 2 \cdot \left(\sum_{i=1}^{n_{ch}} \mathrm{noifs}\,(i) + \sum_{j=1}^{m} \mathrm{noofs}\,(j) \right)$, stands for length of the chromosome $\mathbf{X}_{ch}^{\mathrm{supp}}$, n_{ch} stands for the number of system inputs coded in the chromosome ch (see formula (5)). Moreover, a number of i input fuzzy sets from equation (13) can be reached using function $\mathrm{noifs}\,(i)$ defined as follows

$$\mathrm{noifs}\,(i) = \sum_{k=1}^{N_{ch}} \mathbf{X}_{ch}^{\mathrm{red}} \left\{\mathrm{rule}_k\right\} \cdot \mathbf{X}_{ch}^{\mathrm{red}} \left\{A_i^k\right\}, \quad (15)$$

where N_{ch} stand for number of rules of the system encoded in chromosome ch. Analogically a number of j output fuzzy sets can be calculated.

The lists of parameters encoded in chromosome $\mathbf{X}_{ch}^{\mathrm{supp}}$ does not have specified final elements - their amount depends on the structure of the chromosome $\mathbf{X}_{ch}^{\mathrm{red}}$. It is worth to mention that the lists of parameters are sorted by the centres of the fuzzy sets. Single rows from the $\mathbf{X}_{ch}^{\mathrm{supp}}$ contain parameters connected with specified input and output fuzzy sets. Due to that this approach is different than in case of approach using chromosome $\mathbf{X}_{ch}^{\mathrm{red}}$.

(d) The component $\mathrm{ffint}_D\left(\mathbf{X}_{ch}\right)$ increases the integrity of the shape of the input and output fuzzy sets associated with the inputs and outputs of system (2) encoded in the tested chromosome

$$\mathrm{ffint}_D\left(\mathbf{X}_{ch}\right) = \left(\begin{array}{c} \dfrac{\sum\limits_{i=1}^{n} \mathbf{X}_{ch}^{\mathrm{red}}\{x_i\} \cdot \sqrt{ \sum\limits_{k1=1}^{Nmax} \dfrac{ \mathbf{X}_{ch}^{\mathrm{red}}\{\mathrm{rule}_{k1}\} \cdot \left(\begin{array}{c} \mathbf{X}_{ch}^{\mathrm{par}}\left\{\sigma^A_{i,k1}\right\} + \\ - \dfrac{\sum\limits_{k2=1}^{Nmax} \mathbf{X}_{ch}^{\mathrm{red}}\{\mathrm{rule}_{k2}\} \cdot \mathbf{X}_{ch}^{\mathrm{par}}\left\{\sigma^A_{i,k2}\right\}}{N_{ch}} \end{array} \right)^2 }{N_{ch}} } }{n_{ch}} + \\ + \dfrac{\sum\limits_{j=1}^{m} \sqrt{ \sum\limits_{k1=1}^{Nmax} \dfrac{ \mathbf{X}_{ch}^{\mathrm{red}}\{\mathrm{rule}_{k1}\} \cdot \left(\begin{array}{c} \mathbf{X}_{ch}^{\mathrm{par}}\left\{\sigma^B_{j,k1}\right\} + \\ - \dfrac{\sum\limits_{k2=1}^{Nmax} \mathbf{X}_{ch}^{\mathrm{red}}\{\mathrm{rule}_{k2}\} \cdot \mathbf{X}_{ch}^{\mathrm{par}}\left\{\sigma^B_{j,k2}\right\}}{N_{ch}} \end{array} \right)^2 }{N_{ch}} } }{m} \end{array} \right), \quad (16)$$

where $\mathbf{X}_{ch}^{\mathrm{par}}\left\{\sigma^A_{i,k}\right\}$ stands for a gene of the chromosome $\mathbf{X}_{ch}^{\mathrm{par}}$ associated with the parameter $\sigma^A_{i,k}$ (width of input Gaussian-type fuzzy set A_i^k used in simulations),

$\mathbf{X}_{ch}^{\mathrm{par}}\left\{\sigma_{j,k}^{B}\right\}$ means gene of the chromosome $\mathbf{X}_{ch}^{\mathrm{par}}$ associated with the parameter $\sigma_{j,k}^{B}$.

(e) The component $\mathrm{ffint}_E\left(\mathbf{X}_{ch}\right)$ increases complementarity of the input fuzzy sets of system (2) encoded in the tested chromosome

$$\mathrm{ffint}_E\left(\mathbf{X}_{ch}\right)=\frac{\sum\limits_{z=1}^{Z}\sum\limits_{i=1}^{n}\left(\begin{array}{c}\mathbf{X}_{ch}^{\mathrm{red}}\left\{x_i\right\}\cdot\\ \cdot\max\left(1,\left|1-\sum\limits_{k=1}^{Nmax}\mathbf{X}_{ch}^{\mathrm{red}}\left\{\mathrm{rule}_k\right\}\cdot\mu_{A_i^k}\left(\bar{x}_{z,i}\right)\right|\right)\end{array}\right)}{Z\cdot n_{ch}}. \tag{17}$$

(f) The component $\mathrm{ffint}_F\left(\mathbf{X}_{ch}\right)$ increases readability of the antecedents and weights of rules of system (2) encoded in the tested chromosome

$$\mathrm{ffint}_F\left(\mathbf{X}_{ch}\right)=1-\frac{\left(\begin{array}{c}\sum\limits_{k=1}^{Nmax}\mathbf{X}_{ch}^{\mathrm{red}}\left\{\mathrm{rule}_k\right\}\cdot\left(\frac{\sum\limits_{i=1}^{n}\mathbf{X}_{ch}^{\mathrm{red}}\left\{x_i\right\}\cdot\mu_w\left(w_{i,k}^{A}\right)}{n_{ch}}\right)+\\ +\sum\limits_{k=1}^{Nmax}\mathbf{X}_{ch}^{\mathrm{red}}\left\{\mathrm{rule}_k\right\}\cdot\mu_w\left(w_k^{\mathrm{rule}}\right)\end{array}\right)}{2\cdot N_{ch}}, \tag{18}$$

where $\mu_w\left(w_{i,k}^{A}\right)$ is a function defining congeries around values 0, 0.5 and 1 (in simulations we assumed that $a=0.25$, $b=0.50$ i $c=0.75$). This function is described as follows

$$\mu_w\left(x\right)=\begin{cases}\frac{a-x}{a} & \text{for } x\geq 0 \text{ and } x\leq a\\ \frac{x-a}{b-a} & \text{for } x\geq a \text{ and } x\leq b\\ \frac{c-x}{c-b} & \text{for } x\geq b \text{ and } x\leq c\\ \frac{x-c}{1-c} & \text{for } x\geq c \text{ and } x\leq 1\end{cases}. \tag{19}$$

4 Simulation Results

In our paper we considered the van der Pol oscillator ([63]) which is used in the medicine as the model of the heartbeat. In our simulations three approaches were assumed. In each of them different weights of fitness function (7) were chosen. It is worth to mention that the function (7) is very elastic due to weights, and allows to obtain solutions with different accuracy-interpretability trade-off. Owing to the fact that we choose three specified cases (see Table 1): **(a)** "high accuracy" case, where most important part of the (7) takes responsibility for accuracy of system (2) (see column (a) in Table 1). **(b)** "high interpretability" case, where the most important part of the (7) takes responsibility for interpretability of system (2) (see column (b) in Table 1). **(c)** "good accuracy and good interpretability" (balanced), where both accuracy component and interpretability component weights in the function (7) were set to high values (2) (see column (c) in Table 1). Supplemental properties of our simulations can be

Table 1. Components of fitness function (7) and reduction level of system (2) for different weights of (7) (for the best chromosomes): a) high accuracy case, b) high interpretability case, c) high accuracy and interpretability case

Name of the component	Case (a)	Case (b)	Case (c)
$w_{\text{ffaccuracy}}$	1.00	0.50	0.75
$w_{\text{ffinterpretability}}$	0.50	1.00	0.75
$w_{\text{ffcomplexity}}$	0.50	0.50	0.50
$w_{\text{ffintA}} = w_{\text{ffintB}} = w_{\text{ffintE}}$	0.20	0.20	0.20
$w_{\text{ffintC}} = w_{\text{ffintD}}$	1.00	1.00	1.00
w_{ffintF}	0.50	0.50	0.50
ffaccuracy $(\mathbf{X}_{ch})$	0.0092	0.1531	0.0482
ffinterpretability $(\mathbf{X}_{ch})$	0.3936	0.0001	0.0007
ffcomplexity $(\mathbf{X}_{ch})$	0.5526	0.5789	0.9211
ff $(\mathbf{X}_{ch})$	0.0050	0.0001	0.0688
$RMSE$	0.2212	0.7883	0.5218
Name of the reduced elements	Case (a)	Case (b)	Case (c)
inputs	0/2	0/2	0/2
antecedents	2/6	2/6	3/6
consequences	7/12	6/12	7/12
rules	1/3	1/3	1/3
discretization points	12/20	14/20	13/20

summed up as follows: **(1)** For modelling dynamic objects the method presented in our previous work [39] was used. In this method, every output of system (2) generates one element of variable state matrix. **(2)** For system (2) a Gaussian functions with algebraic triangular norms were used. **(3)** The following properties of evolutionary algorithm was assumed: the number of chromosomes in the population was set to 100, the algorithm performs 10 000 steps (generations), the crossover probability was set as $p_c = 0.8$, the mutation probability was set as $p_m = 0.2$, the mutation intensity was set as $\sigma = 0.3$. **(4)** In interpretability component (10) of the fitness function (7) following weights was set: ffint$_C$ $(\mathbf{X}_{ch})$ $(w_{\text{ffintC}} = 1.0)$, ffint$_D$ $(\mathbf{X}_{ch})$ $(w_{\text{ffintD}} = 1.0)$ i ffint$_F$ $(\mathbf{X}_{ch})$ $(w_{\text{ffintF}} = 0.5)$. Weights of remaining components of function (10) was set as 0.2.

The conclusions from simulations can be summarized as follows: **(1)** Example (a) allowed to obtain a system with very high accuracy and quite acceptable interpretability. **(2)** Example (b) and (c) allowed obtain high readability of fuzzy sets (see Fig. 1), high readability of weights of fuzzy sets and rules (see Fig. 2) and good accuracy of the system (see Fig. 3). **(3)** Example (c) (as predicted) allowed obtain better accuracy of the system than example b (b) with acceptable compromise between semantic interpretability and complex-based interpretability (see Table 1).

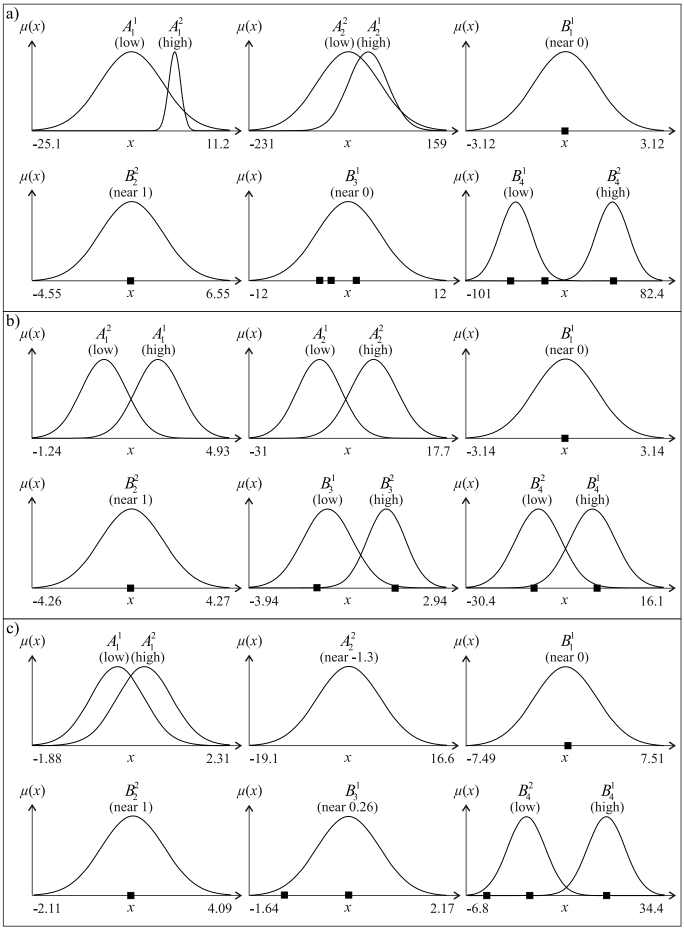

Fig. 1. Inputs and outputs fuzzy sets of the neuro-fuzzy system (2) for the van der Pol oscillator problem for three various settings of the function (7) (for the best chromosomes): a) high accuracy case, b) high interpretability case, c) high accuracy and interpretability case

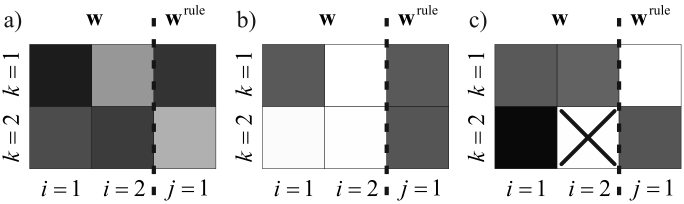

Fig. 2. Weights representation in the neuro-fuzzy system (2) (dark areas correspond to low values of weights and vice versa) for the van der Pol oscillator problem for different weights configuration of the function (7) (for the best chromosomes): a) high accuracy case, b) high interpretability case, c) high accuracy and interpretability case

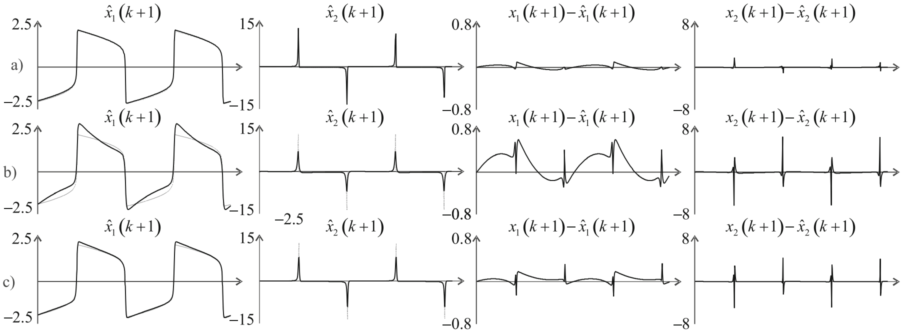

Fig. 3. Trajectories obtained for the van der Pol oscillator for three different weights configuration of the function (7) (for the best chromosomes): a) high accuracy case, b) high interpretability case, c) high accuracy and interpretability case

5 Conclusions

In this article we proposed a new method for designing fuzzy systems for nonlinear modelling using different criteria of interpretability. Proposed interpretability criteria consider both complexity of the system and semantic interpretability of knowledge accumulated within it. Those criteria were used in the fitness function of presented new hybrid population-based method which uses possibilities of genetic algorithm, evolutionary strategy (μ, λ) and bees algorithm. This method works in two phases, in the first the structure and parameters of the system are chosen, in the second phase the system components are tuned. Our simulation results affirmed effectiveness of the proposed approaches.

Acknowledgment. The project was financed by the National Science Centre (Poland) on the basis of the decision number DEC-2012/05/B/ST7/02138.

References

1. Bartczuk, Ł., Dziwiński, P., Starczewski, J.T.: A new method for dealing with unbalanced linguistic term set. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part I. LNCS (LNAI), vol. 7267, pp. 207–212. Springer, Heidelberg (2012)
2. Bartczuk, Ł., Dziwiński, P., Starczewski, J.T.: New Method for Generation Type-2 Fuzzy Partition for FDT. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010, Part I. LNCS, vol. 6113, pp. 275–280. Springer, Heidelberg (2010)
3. Bartczuk, Ł., Przybył, A., Dziwiński, P.: Hybrid state variables - fuzzy logic modelling of nonlinear objects. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2013, Part I. LNCS, vol. 7894, pp. 227–234. Springer, Heidelberg (2013)

4. Bartczuk, Ł., Rutkowska, D.: A New Version of the Fuzzy-ID3 Algorithm. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 1060–1070. Springer, Heidelberg (2006)
5. Bartczuk, Ł., Rutkowska, D.: Medical Diagnosis with Type-2 Fuzzy Decision Trees. In: Kącki, E., Rudnicki, M., Stempczyńska, J. (eds.) Computers in Medical Activity. AISC, vol. 65, pp. 11–21. Springer, Heidelberg (2009)
6. Bilski, J., Rutkowski, L.: Numerically Robust Learning Algorithms for Feed Forward Neural Networks. In: Advances in Soft Computing - Neural Networks and Soft Computing, pp. 149–154. Physica-Verlag, A Springer-Verlag Company (2003)
7. Cpalka, K.: A Method for Designing Flexible Neuro-fuzzy Systems. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 212–219. Springer, Heidelberg (2006)
8. Cpałka, K., Rutkowski, L.: Flexible Takagi Sugeno Neuro-fuzzy Structures for Nonlinear Approximation. WSEAS Transactions on Systems 4(9), 1450–1458 (2005)
9. Dekker, M.: Advanced Process Identification and Control, Incorporated, ch. 1 (2002)
10. Dziwiński, P., Bartczuk, Ł., Starczewski, J.T.: Fully controllable ant colony system for text data clustering. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) SIDE 2012 and EC 2012. LNCS, vol. 7269, pp. 199–205. Springer, Heidelberg (2012)
11. Dziwiński, P., Rutkowska, D.: Algorithm for generating fuzzy rules for WWW document classification. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 1111–1119. Springer, Heidelberg (2006)
12. Dziwiński, P., Rutkowska, D.: Ant focused crawling algorithm. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 1018–1028. Springer, Heidelberg (2008)
13. Dziwiński, P., Starczewski, J.T., Bartczuk, Ł.: New linguistic hedges in construction of interval type-2 FLS. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010, Part II. LNCS, vol. 6114, pp. 445–450. Springer, Heidelberg (2010)
14. El-Abd, M.: On the hybridization on the artificial bee colony and particle swarm optimization algorithms. Journal of Artificial Intelligence and Soft Computing Research 2(2), 147–155 (2012)
15. Gabryel, M., Cpałka, K., Rutkowski, L.: Evolutionary strategies for learning of neuro-fuzzy systems. In: Proceedings of the I Workshop on Genetic Fuzzy Systems, Granada, pp. 119–123 (2005)
16. Gacto, M.J., Alcala, R., Herrera, F.: Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures. Information Sciences 181, 4340–4360 (2011)
17. Ghandar, A., Michalewicz, Z.: An experimental study of Multi-Objective Evolutionary Algorithms for balancing interpretability and accuracy in fuzzy rule base classifiers for financial prediction. In: 2011 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics, pp. 1–6 (2011)
18. Greblicki, W., Rutkowska, D., Rutkowski, L.: An orthogonal series estimate of time-varying regression. Annals of the Institute of Statistical Mathematics 35(2), 215–228 (1983)
19. Horzyk, A., Tadeusiewicz, R.: Self-Optimizing Neural Networks. In: Yin, F.-L., Wang, J., Guo, C. (eds.) ISNN 2004. LNCS, vol. 3173, pp. 150–155. Springer, Heidelberg (2004)

20. Jelonkiewicz, J., Przybył, A.: Accuracy improvement of neural network state variable estimator in induction motor drive. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 71–77. Springer, Heidelberg (2008)
21. Kamyar, M.: Takagi-Sugeno Fuzzy Modeling for Process Control. In: Industrial Automation Robotics and Artificial Intelligence (EEE8005), School of Electrical, Electronic and Computer Engineering (2008)
22. Korytkowski, M., Nowicki, R., Rutkowski, L., Scherer, R.: AdaBoost Ensemble of DCOG Rough–Neuro–Fuzzy Systems. In: Jędrzejowicz, P., Nguyen, N.T., Hoang, K. (eds.) ICCCI 2011, Part I. LNCS, vol. 6922, pp. 62–71. Springer, Heidelberg (2011)
23. Korytkowski, M., Rutkowski, L., Scherer, R.: On combining backpropagation with boosting. In: Proceedings of the IEEE International Joint Conference on Neural Network (IJCNN), vols. 1-10, pp. 1274–1277 (2006)
24. Korytkowski, M., Rutkowski, L., Scherer, R.: From Ensemble of Fuzzy Classifiers to Single Fuzzy Rule Base Classifier. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 265–272. Springer, Heidelberg (2008)
25. Laskowski, Ł.: A novel hybrid-maximum neural network in stereo-matching process. Neural Comput. & Applic. 23, 2435–2450 (2013)
26. Laskowski, Ł.: Objects auto-selection from stereo-images realised by self-correcting neural network. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part I. LNCS, vol. 7267, pp. 119–125. Springer, Heidelberg (2012)
27. Li, X., Er, M.J., Lim, B.S., Zhou, J.H., Gan, O.P., Rutkowski, L.: Fuzzy Regression Modeling for Tool Performance Prediction and Degradation Detection. International Journal of Neural Systems 20(5), 405–419 (2010)
28. Lobato, F.S., Steffen Jr., V.: A new multi-objective optimization algorithm based on differential evolution and neighborhood exploring evolution strategy. Journal of Artificial Intelligence and Soft Computing Research 1(4), 259–267 (2011)
29. Lobato, F.S., Steffen Jr., V., Silva Neto, A.J.: Solution of singular optimal control problems using the improved differential evolution algorithm. Journal of Artificial Intelligence and Soft Computing Research 1(3), 195–206 (2011)
30. Łapa, K., Zalasiński, M., Cpałka, K.: A new method for designing and complexity reduction of neuro-fuzzy systems for nonlinear modelling. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2013, Part I. LNCS, vol. 7894, pp. 329–344. Springer, Heidelberg (2013)
31. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer (1999)
32. Nowicki, R.: On classification with missing data using rough-neuro-fuzzy systems. International Journal of Applied Mathematics and Computer Science 20(1), 55–67 (2010)
33. Nowicki, R., Rutkowski, R.: Soft Techniques for Bayesian Classification. In: Rutkowski, L., Kacprzyk, J. (eds.) Neural Networks and Soft Computing. Advances in Soft Computing, pp. 537–544. Springer Physica-Verlag (2003)
34. Nowicki, R., Scherer, R., Rutkowski, L.: A method for learning of hierarchical fuzzy systems. In: Sincak, P., Vascak, J., Kvasnicka, V., Pospichal, J. (eds.) Intelligent Technologies - Theory and Applications, pp. 124–129. IOS Press (2002)
35. Patan, K., Korbicz, J.: Nonlinear model predictive control of a boiler unit: A fault tolerant control study. Applied Mathematics and Computer Science 22(1), 225–237 (2012)

36. Pławiak, P., Tadeusiewicz, R.: Approximation of phenol concentration using novel hybrid computational intelligence methods. Applied Mathematics and Computer Science 24(1) (in print, 2014)
37. Pham, D.T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M.: The Bees Algorithm, A Novel Tool for Complex Optimisation Problems. In: Proceedings of the 2nd International Virtual Conference on Intelligent Production Machines and Systems, pp. 454–459 (2006)
38. Prampero, P.S., Attux, R.: Magnetic particle swarm optimization. Journal of Artificial Intelligence and Soft Computing Research 2(1), 59–72 (2012)
39. Przybył, A., Cpałka, K.: A new method to construct of interpretable models of dynamic systems. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part II. LNCS, vol. 7268, pp. 697–705. Springer, Heidelberg (2012)
40. Przybył, A., Jelonkiewicz, J.: Genetic algorithm for observer parameters tuning in sensorless induction motor drive. In: Rutkowski, L., Kacprzyk, J. (eds.) Networks and Soft Computing (6th International Conference on Neural Networks and Soft Computing 2002), Zakopane, Poland, pp. 376–381 (2003)
41. Przybył, A., Smoląg, J., Kimla, P.: Distributed Control System Based on Real Time Ethernet for Computer Numerical Controlled Machine Tool (in Polish). Przeglad Elektrotechniczny 86(2), 342–346 (2010)
42. Rutkowski, L.: Computational Intelligence. Springer (2008)
43. Rutkowski, L.: An application of multiple Fourier series to identification of multivariable nonstationary systems. International Journal of Systems Science 20(10), 1993–2002 (1989)
44. Rutkowski, L.: The real-time identification of time-varying systems by nonparametric algorithms based on the Parzen kernels. International Journal of Systems Science 16, 1123–1130 (1985)
45. Rutkowski, L.: Flexible structures of neuro-fuzzy systems. In: Sincak, P., Vascak, J. (eds.) Quo Vadis Computational Intelligence. STUDFUZZ, vol. 54, pp. 479–484. Springer, Heidelberg (2000)
46. Rutkowski, L., Cpałka, K.: Flexible weighted neuro-fuzzy systems. In: Proceedings of the 9th International Conference on Neural Information Processing (ICONIP 2002), Orchid Country Club, Singapore, CD, November 18-22 (2002)
47. Rutkowski, L., Przybył, A., Cpałka, K., Er, M.J.: Online Speed Profile Generation for Industrial Machine Tool Based on Neuro-fuzzy Approach. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010, Part II. LNCS (LNAI), vol. 6114, pp. 645–650. Springer, Heidelberg (2010)
48. Rutkowski, L., Przybył, A., Cpałka, K.: Novel on-line speed profile generation for industrial machine tool based on flexible neuro-fuzzy approximation. IEEE Transactions on Industrial Electronics 59, 1238–1247 (2012)
49. Scherer, R.: Neuro-fuzzy relational systems for nonlinear approximation and prediction. Nonlinear Analysis Series A: Theory, Methods and Applications 71(12), e1420–e1425 (2009)
50. Scherer, R., Rutkowski, L.: Connectionist fuzzy relational systems. In: Halgamuge, S.K., Wang, L. (eds.) Computational Intelligence for Modelling and Prediction. SCI, vol. 2, pp. 35–47. Springer, Heidelberg (2005)
51. Siwek, K., Osowski, S., Szupiluk, R.: Ensemble neural network approach for accurate load forecasting in a power system. Applied Mathematics and Computer Science 19(2), 303–315 (2009)

52. Starczewski, J.T.: A Type-1 Approximation of Interval Type-2 FLS. In: Di Gesù, V., Pal, S.K., Petrosino, A. (eds.) WILF 2009. LNCS, vol. 5571, pp. 287–294. Springer, Heidelberg (2009)
53. Starczewski, J.T., Rutkowski, L.: Connectionist Structures of Type 2 Fuzzy Inference Systems. In: Wyrzykowski, R., Dongarra, J., Paprzycki, M., Waśniewski, J. (eds.) PPAM 2001. LNCS, vol. 2328, pp. 634–642. Springer, Heidelberg (2002)
54. Starczewski, J.T., Rutkowski, L.: Interval type 2 neuro-fuzzy systems based on interval consequents. In: Rutkowski, L., Kacprzyk, J. (eds.) Neural Networks and Soft Computing. Advances in Soft Computing, pp. 570–577. Springer, Heidelberg (2003)
55. Starczewski, J.T., Scherer, R., Korytkowski, M., Nowicki, R.: Modular type-2 neuro-fuzzy systems. In: Wyrzykowski, R., Dongarra, J., Karczewski, K., Wasniewski, J. (eds.) PPAM 2007. LNCS, vol. 4967, pp. 570–578. Springer, Heidelberg (2008)
56. Szaleniec, M., Goclon, J., Witko, M., Tadeusiewicz, R.: Application of artificial neural networks and DFT-based parameters for prediction of reaction kinetics of ethylbenzene dehydrogenase. Journal of Computer-Aided Molecular Design 20(3), 145–157 (2006)
57. Zhou, S.M., Gan, J.Q.: Low-level interpretability and high-level interpretability: a unified view of data-driven interpretable fuzzy system modelling. Fuzzy Sets and Systems 159, 3091–3131 (2008)
58. Zalasiński, M., Łapa, K., Cpałka, K.: New Algorithm for Evolutionary Selection of the Dynamic Signature Global Features. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2013, Part II. LNCS (LNAI), vol. 7895, pp. 113–121. Springer, Heidelberg (2013)
59. Zalasiński, M., Cpałka, K.: A new method of on-line signature verification using a flexible fuzzy one-class classifier, pp. 38–53. Academic Publishing House EXIT (2011)
60. Zalasiński, M., Cpałka, K.: New Approach for the On-Line Signature Verification Based on Method of Horizontal Partitioning. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2013, Part II. LNCS (LNAI), vol. 7895, pp. 342–350. Springer, Heidelberg (2013)
61. Zalasiński, M., Cpałka, K.: Novel algorithm for the on-line signature verification. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part II. LNCS (LNAI), vol. 7268, pp. 362–367. Springer, Heidelberg (2012)
62. Zalasiński, M., Cpałka, K.: Novel Algorithm for the On-Line Signature Verification Using Selected Discretization Points Groups. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2013, Part I. LNCS (LNAI), vol. 7894, pp. 493–502. Springer, Heidelberg (2013)
63. Żebrowski, J., Grudziński, K.: Observations and modelling of unusual patterns in human heart rate variability. Acta Physica Polonica B 36, 1881–1894 (2005)