EVOLUTION–FUZZY RULE BASED SYSTEM WITH PARAMETERIZED CONSEQUENCES

PIOTR CZEKALSKI

Institute of Computer Sciences, Silesian University of Technology
ul. Akademicka 16, 44–100 Gliwice, Poland
e-mail: pczekalski@zmitac.iinf.polsl.gliwice.pl

While using automated learning methods, the lack of accuracy and poor knowledge generalization are both typical problems for a rule-based system obtained on a given data set. This paper introduces a new method capable of generating an accurate rule-based fuzzy inference system with parameterized consequences using an automated, off-line learning process based on multi-phase evolutionary computing and a training data covering algorithm. The presented method consists of the following steps: obtaining an initial set of rules with parameterized consequences using the Michigan approach combined with an evolutionary strategy and a covering algorithm for the training data set; reducing the obtained rule base using a simple genetic algorithm; multi-phase tuning of the fuzzy inference system with parameterized consequences using the Pittsburgh approach and an evolutionary strategy. The paper presents experimental results using popular benchmark data sets regarding system identification and time series prediction, providing a reliable comparison to other learning methods, particularly those based on neuro-fuzzy, clustering and ε-insensitive methods. An examplary fuzzy inference system with parameterized consequences using the Reichenbach implication and the minimum t-norm was implemented to obtain numerical results.

Keywords: evolutionary strategy, fuzzy inference system, off-line learning, hybrid system

1. Introduction

The complexity of real-world problems, the absence of explaining theories and the lack of knowledge on analyzed phenomena as well as high expenses of human experts require sophisticated methods of preparing accurate intelligent systems capable of reflecting reality and able to generalize their knowledge in unforeseen cases. A rule-based Fuzzy Inference System (FIS) introduced by Zadeh (1971) became fundamental for further work and lead to the Mamdani-Assilian (Mamdani and Assilian, 1975) linguistic FIS, which is perhaps the most popular one. The disadvantages of Mamdani's FIS (particularly, high complexity due to the aggregation operation) lead to simplifications, i.e., Larsen's and Sugeno-Yakusawa's FISs. Takagi, Sugeno and Kang (TSK) presented a low-complexity, computationally effective model without fuzzy sets within rule consequences (Sugeno and Kang, 1988; Takagi and Sugeno, 1985). The synthesis of Mamdani and TSK models was done by Czogała and Łęski, cf. a fuzzy inference system with parameterized conclusion (Czogała and Łęski, 1996; 1999).

Even a most advanced rule-based FIS is unable to provide correct results without a proper Rule Base (RB). Obtaining an optimal set of rules is a state-of-the art piece of work. Whenever there is a lack of knowledge on the problem considered or it is impossible to achieve a model RB from a human expert, automated learning methods are set to work. Various on-line and off-line learning methods have been developed since the beginnings of soft computing. Some pertaining methods are neural networks (Tadeusiewicz, 1998), neuro-fuzzy (Czogała and Łęski, 1999; Fuller, 1999) and evolving-fuzzy systems (Cordón *et al.*, 2004). Pure neural network based methods and hybrid neural methods are considered to be imperfect as there are known problems related to global optimization results. Genetic algorithms and evolution strategies are considered as powerful global optimization tools, and are thus interesting components for the construction of hybrid systems (Angelov, 2002; Cordón *et al.*, 2001; Herrera and Verdegay, 1996; Pedrycz, 1997). The next section presents an overview of multi-phase off-line learning methods and a proposal of a rival solution. The following sections describe the components and processes of the presented method. The subsequent one discusses practical experiments and results on system identification, training data approximation and chaotic time series prediction, as well as further research directions.

2. Fuzzy Inference System with Parameterized Consequences

A linguistic fuzzy inference system with fuzzy sets in consequences, whose locations are described as linear combinations of input singletons, was introduced and presented in depth by Czogała and Łęski (1996; 1999). Compared with the Mamdani FIS and the TSK FIS, a parameterized consequent provides an extra degree of freedom for every single rule, thus making such a system potentially more accurate than the above-mentioned ones, but still able to represent its knowledge base in a form acceptable by the human. We assume the following: a multiple input single output (MISO) model with n inputs where every i-th fuzzy if-then rule $R^{(i)}$ is specified by a set of fuzzy input sets $A_n^{(i)}$, an output fuzzy set $B^{(i)}$ and a set of parameters $\theta^{(i)}$ that can be represented using the following notation (Czogała and Łęski, 1999):

$$R^{(i)}\colon \ \textbf{IF} \ \underset{n=1}{\overset{N}{\textbf{and}}} X_n \ \textbf{is} \ A_n^{(i)} \ \textbf{THEN} \ Y \ \textbf{is} \ B^{(i)}(\underline{\theta}, \underline{x}_0), \quad (1)$$

where X_n and Y denotes linguistic variables (inputs and the output, respectively) and $B^{(i)}(\underline{\theta}, \underline{x}_0)$ represents a parameterized linguistic value of the output linguistic variable Y. Assuming the logical interpretation of if-then rules, the output fuzzy value can be written as (Czogała and Łęski, 1999):

$$\begin{aligned}\mu_{B'}(y, \underline{x}_0) &\bigoplus_{i=1}^{I} \mu_{B^{(i)}}(y) \\ &= \bigoplus_{i=1}^{I} \Psi\big(\mu_{\underline{A}^{(i)}}(\underline{x}_0), \mu_{B^{(i)}}(y, \underline{x}_0)\big), \quad (2)\end{aligned}$$

where Ψ denotes fuzzy implication, $\bigoplus$ stands for the aggregation operation, $\mu_{B'}$ and $\mu_{\underline{A}}$ represent fuzzy membership functions (the output and inputs, respectively). Assuming isosceles triangular membership functions for the rule consequent part, the output crisp value y_0 of the FIS, excited by the vector of singletons $\underline{x}_0$ representing input data, can be obtained using the modified indexed center-of-gravity (MICOG) defuzzifier (Czogała and Łęski, 1999):

$$y_0 = \frac{\sum_{i=1}^{I} g\big(F^{(i)}(\underline{x}_0), w^{(i)}\big) y^{(i)}(\underline{x}_0)}{\sum_{i=1}^{I} g\big(F^{(i)}(\underline{x}_0), w^{(i)}\big)}, \quad (3)$$

where $y^{(i)}(\underline{x}_0)$ denotes the location of the center of gravity on the X axis of the consequent fuzzy set for the i-th rule, $F^{(i)}(\underline{x}_0)$ denotes the firing strength (the activation level) of the i-th rule (Czogała and Łęski, 1999), $w^{(i)}$ denotes the support length of the triangle consequent fuzzy set for the i-th rule. The form of the function g depends on the chosen fuzzy implication. In this article, the Reichenbach fuzzy implication was selected and thus g is calculated as (Czogała and Łęski, 1999):

$$g = \frac{w^{(i)}}{2} F^{(i)}(\underline{x}_0). \quad (4)$$

The Gaussian membership function for the input linguistic variables was selected as

$$\mu_A(x) = e^{-\frac{(x-c)^2}{2\sigma^2}}, \quad (5)$$

where c and σ denote the core location and the Gaussian bell deviation, respectively, and the firing strength $F^{(i)}(\underline{x}_0)$ i-th rule is calculated using a selected t-norm (Czogała and Łęski, 1999):

$$F^{(i)}(\underline{x}_0) = \mu_{A_1^{(i)}}(x_{0,1}) \star_T \cdots \star_T \mu_{A_N^{(i)}}(x_{0,N}), \quad (6)$$

where $\star_T$ stands for the t-norm.

3. Multi-Phase FIS Development Method Using a GA and an ES

Various hybrid fuzzy systems have been developed in the last decade (Cordón *et al.*, 2004). Literature analysis shows that the most prominent systems are those using a knowledge base (KB) of fuzzy if-then rules, where genetic algorithms (GAs) and evolution strategies (ESs) constitute a toolbox for processing on different levels of complexity, starting from FIS parameter optimization, but even learning a total FIS KB. The following sections present a multiphase learning method, capable of obtaining an FIS KB: its rule-base set (RB) and the fuzzy membership functions, the so-called fuzzy database (DB), from a training data set. The first phase contains an iterative algorithm that extracts fuzzy if-then rules and a fuzzy database simultaneously, with the use of an iterative rule learning approach (Cordón and Herrera, 1997b; 2001; Cordón *et al.*, 1999; González and Pérez, 1999) and an evolutionary strategy (ES) using the Michigan approach (Bonarini, 1996; Holland and Reitman, 1978; Ishibuchi *et al.*, 1999; Parodi and Bonelli, 1993; Valenzuela-Rendón, 1991; Velasco, 1998). The following phase performs rule-base reduction, selecting the best fuzzy rules (in terms of selected criteria) and removing redundant ones using GAs. The final phases perform fuzzy DB tuning with the use of ESs, applying the Pittsburgh approach (Baron *et al.*, 2001; Carse *et al.*, 1996; Hoffmann and Pfister, 1997; Holland and Reitman, 1978; Lee and Takagi, 1993; Magdalena and Monasterio, 1997; Park *et al.*, 1994; Pham and Karaboga, 1991; Thrift, 1991). The whole process is presented in Fig. 1.

3.1. Fuzzy Knowledge Base Extraction from Examples. The iterative rule learning processs consists in using two elements: a covering algorithm and an ES, responsible for a fuzzy rule discovery, based on the training data set E_{N_T}. The primary training data set is considered to be noiseless and composed of N_T numerical input-output pairs $e_{N_T}[j] = (\underline{x}_0[j], t_0[j]), \underline{x}_0[j] = x_{0,0}, \ldots, x_{0,n}$ for $j = 1, \ldots, N_T$. Each iteration generates a single fuzzy

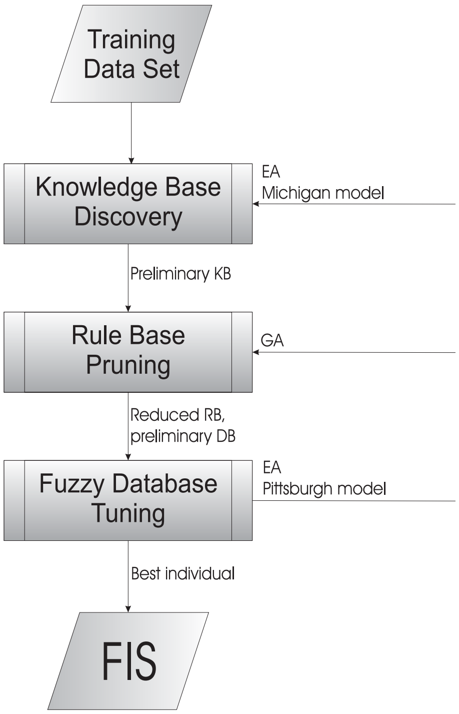

Fig. 1. General scheme of the multi phase process.

rule $R^{(i)}$ constituting an element of the FIS KB referenced as $\mathcal{R}$. The i-th rule description contains membership function definitions of the rule premises $\mu_{A^{(i)}}(\cdot)$ and the consequences $\mu_{B^{(i)}}(\cdot)$, but also linear parameters—a matrix $\underline{\Theta}$, describing the locations of fuzzy membership functions for the rule consequences. Additionally, a training data subset $\mathcal{E_C}$ is introduced, containing currently processed training data pairs. Assuming that every iteration of the covering algorithm adds a newly discovered rule to $\mathcal{R}$, the training data set $\mathcal{E_C}$ is modified to make it contain only those elements from E_{N_T} that are left uncovered. During the first iteration, we have $\mathcal{E_C} = E_{N_T}$.

3.1.1. Covering algorithm.

The covering value of the pair $e_{N_T}[j]$ is defined as (Cordón and Herrera, 1997a):

$$CV_{\mathcal{R}}\big(e_{N_T}[j]\big) = \sum_{i=1}^{I} R_c^{(i)}\big(e_{N_T}[j]\big). \tag{7}$$

The learning process ends when for all N_T tuples within the training data set E_{N_T} we have (Cordón and Herrera, 1997a):

$$CV_{\mathcal{R}}\big(e_{N_T}[j]\big) \geq \epsilon, \tag{8}$$

where $j = 1, \ldots, N_T$. Here $R_c^{(i)}(e_{N_T}[j])$ denotes the compatibility degree between the i-th rule and j-th training data pair, computed as

$$R_c^{(i)}\big(e_{N_T}[j]\big) = \mu_{\underline{A}^{(i)}}\big(\underline{x}_0[j]\big) \star_{T_c} \mu_{B^{(i)}}\big(t_0[j]\big), \tag{9}$$

where $\star_{T_c}$ stands for a t-norm. In this article, the *minimum* t-norm was chosen. To obtain the value of $R_c^{(i)}(e_{N_T}[j])$, it is necessary to estimate the linear consequence parameter vector $\underline{\theta}^{(i)}$ in order to locate the output triangle fuzzy set, and hence the weighted least squares estimator (Łęski, 2006) on the training data set E_{N_T} was used:

$$\underline{\theta}^{(i)} = \big(\underline{X}^{\mathsf{T}}\underline{G}^{(i)}\underline{X}\big)^{-1}\underline{X}^{\mathsf{T}}\underline{G}^{(i)}\underline{T}, \tag{10}$$

where $\underline{X}$ denotes the modified input data matrix

$$\underline{X} = \begin{bmatrix} \underline{x}_0^{'}[1]^{\mathsf{T}} \\ \vdots \\ \underline{x}_0^{'}[N_T]^{\mathsf{T}} \end{bmatrix}, \tag{11}$$

$$\underline{x}_0^{'}[j] = \begin{bmatrix} 1 \\ \underline{x}_0[j] \end{bmatrix}. \tag{12}$$

The matrix $\underline{G}$ represents the weights of the training data pairs for the i-th rule, defined as

$$\underline{G}^{(i)} = \begin{bmatrix} g\big(F^{(i)}\big(x_0[1]\big)\big) & 0 & \cdots & 0 \\ 0 & & & \vdots \\ & & \ddots & \\ \vdots & & & 0 \\ 0 & \cdots & 0 & g\big(F^{(i)}\big(x_0[N_T]\big)\big) \end{bmatrix}, \tag{13}$$

where the function g depends on the chosen fuzzy implication, cf. (4), and $\underline{T}$ denotes the output values vector of the training data set

$$\underline{T} = \big[t_0(1), \ldots, t_0(N_T)\big]^{\mathsf{T}}. \tag{14}$$

The global rule learning process estimates the consequent parameter matrix $\underline{\Theta}$ composed of the vectors $\underline{\theta}^{(i)}$ (one per rule) whenever it is necessary to simultaneously obtain these parameters for all the rules constituting the FIS,

$$\underline{\Theta} = \Big[\underline{\theta}^{(1)\mathsf{T}}, \ldots, \underline{\theta}^{(I)\mathsf{T}}\Big]^{\mathsf{T}}. \tag{15}$$

The process utilizes the least-squares estimator defined on the training data set E_{N_T} (Łęski, 2006):

$$\underline{\Theta} = \big(\underline{D}^{\mathsf{T}}\underline{D}\big)^{-1}\underline{D}^{\mathsf{T}}\underline{T}, \tag{16}$$

where $\underline{D}$ denotes the rule activation matrix defined as

$$\underline{D} = \begin{bmatrix} \overline{g}^{(1)}(x_0[1])\underline{x}_0^{'}[1]^{\mathsf{T}} & \cdots & \overline{g}^{(I)}(x_0[1])\underline{x}_0^{'}[1]^{\mathsf{T}} \\ \vdots & \ddots & \vdots \\ \overline{g}^{(1)}(x_0[N_T])\underline{x}_0^{'}[N_T]^{\mathsf{T}} & \cdots & \overline{g}^{(I)}(x_0[N_T])\underline{x}_0^{'}[N_T]^{\mathsf{T}} \end{bmatrix}. \tag{17}$$

Here $\overline{g}^{(i)}$ represents the normalized output value for the i-th rule,

$$\overline{g}^{(i)}\big(\underline{x}_0(j)\big) = \frac{g\left(F^{(i)}\left(\underline{x}_0\left(j\right)\right), w^{(i)}\right)}{\sum\limits_{k=1}^{I} g\left(F^{(k)}\left(\underline{x}_0\left(j\right)\right), w^{(k)}\right)}. \tag{18}$$

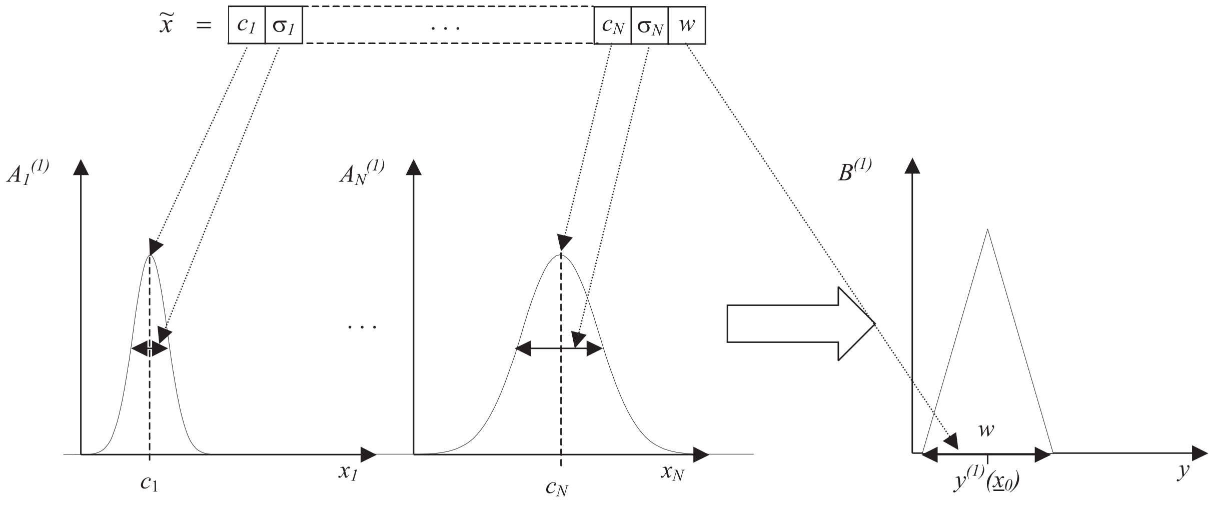

Fig. 2. Michigan rule coding.

3.1.2. Rule discovery. The rule discovery algorithm utilizes an $(\mu + \lambda)$ ES. The training data subset $\mathcal{E}_{\mathcal{C}}$, used during the rule discovery process, consists of the uncovered elements from the training data set, directing the method towards rules that potentially differ from those already constituting the KB. The ES utilizes a real coding (Arabas, 2001), an elitist strategy and stochastic sampling (Holland, 1975), max-min-arithmetic crossover (Herrera *et al.*, 1995) and periodic-nonuniform mutation operators. In accordance with the Michigan model, each chromosome represents a single rule. The unconstrained free semantics (Cordón and Herrera, 1997a) was selected, and thus a chromosome is a vector of $2N + 1$ real numbers representing the parameters of membership functions, two values for each input Gaussian fuzzy set and one value representing the support of the output isoscele triangle fuzzy set (see Fig. 2). Whenever it is necessary to estimate the output value for a rule, the local rule learning process is executed, cf. (10).

The starting pool consists of $\mu \quad (= M_1 + M_{1,2} + M_2 + M_3)$ chromosomes generated heterogeneously:

- M_1 chromosomes are generated using a fuzzy c-means (FCM) algorithm (Bezdek, 1981) performed on the data set $\mathcal{E}_{\mathcal{C}}$, (the total number of chromosomes obtained in this way is denoted by n_{FCM});
- $M_{1,2}$ chromosomes are also generated with the use of FCM, including only two cluster seeds (in fact, no more than two rules if the FCM clustering succeeds at all);
- M_2 chromosomes are generated within the training data input and output intervals of performance, obtained from the training data set E_{N_T};
- M_3 chromosomes are generated randomly without any reference to the training data set.

The t estimator sets a balance between the clustered and random pools and is defined as

$$t = \min\left(\text{round}\,(0.75\mu), \text{card}(\mathcal{E}_{\mathcal{C}})\right). \tag{19}$$

Thus, it is required that

$$M_1 + 2 \leq \text{round}(t/2), \tag{20}$$

and

$$\begin{aligned} M_1 &= \min\left(n_{FCM}, \text{card}\left(\mathcal{E}_{\mathcal{C}} - (t/2) + 1\right)\right), \\ M_{1,2} &\in \{0, 2\}, \\ M_2 &= t - (M_1 + M_{1,2}), \\ M_3 &= \mu - t. \end{aligned} \tag{21}$$

M_2 chromosomes are generated heuristically using a training data subset where $\min\left(M_2, \text{card}\left(\mathcal{E}_{\mathcal{C}}\right)\right)$ pairs are selected from the subset $\mathcal{E}_{\mathcal{C}}$ with uniform distribution and the DB parameters of the chromosome are determined as

$$\begin{aligned} c_{j,n} &= \mathcal{E}_{\mathcal{C}}\left(x_{0,n}[j]\right), \\ \Delta x_n &= \max\left(\mathcal{E}_{\mathcal{C}}\left(x_{0,n}[j]\right) - \overleftarrow{x_{0,n}}, \overrightarrow{x_{0,n}} - \mathcal{E}_{\mathcal{C}}\left(x_{0,n}[j]\right)\right), \\ \sigma_{j,n} &= \sqrt{\frac{\left(\delta[0.0001, 1]\Delta x_n\right)^2}{-2\ln\left(G_L\right)}}, \\ w_j &= 2\delta[0.0001, 1] \\ &\quad \times \left(\min\left(\mathcal{E}_{\mathcal{C}}(t_0[j]) - \overleftarrow{t_0}, \overrightarrow{t_0} - \mathcal{E}_{\mathcal{C}}(t_0[j])\right)\right), \end{aligned} \tag{22}$$

where $n = 1, \ldots, N$, $j = 1, \ldots, M_2$, $\delta[\cdot, \cdot]$ stands for a uniformly distributed random variable and G_L is a parameter.

M_3 chromosomes are generated randomly within the corresponding intervals of performance, defined as

$$\begin{aligned}
x_{0,n}^{-} &= \min_{1\leq n_t\leq \mathrm{card}(\mathcal{E}_{\mathcal{C}})} \big(x_{0,n}(n_t)\big),\\
x_{0,n}^{+} &= \max_{1\leq n_t\leq \mathrm{card}(\mathcal{E}_{\mathcal{C}})} \big(x_{0,n}(n_t)\big),\\
\underline{x_0}^{-} &= \left[x_{0,1}^{+},\dots,x_{0,N}^{+}\right]^{\mathsf{T}},\\
\underline{x_0}^{+} &= \left[x_{0,1}^{-},\dots,x_{0,N}^{-}\right]^{\mathsf{T}},\\
[\overleftarrow{\underline{x_0}},\overrightarrow{\underline{x_0}}] &= \left[\underline{x_0}^{-}-0.1(\underline{x_0}^{+}-\underline{x_0}^{-}),\underline{x_0}^{+}+0.1(\underline{x_0}^{+}-\underline{x_0}^{-})\right],\\
\left[\overleftarrow{t_0},\overrightarrow{t_0}\right] &= \left[t_0^{-}-0.1(t_0^{+}-t_0^{-}),t_0^{+}+0.1(t_0^{+}-t_0^{-})\right].
\end{aligned} \tag{23}$$

All the DB parameters within the M_3 scope are generated randomly with uniform distribution. It is required that randomized DB parameters $c_{j,n}$, $\sigma_{j,n}$, w_j (Fig. 2) belong to the intervals of performance

$$\begin{aligned}
c_{j,n} &\in [\overleftarrow{x_{0,n}},\overrightarrow{x_{0,n}}],\\
\sigma_{j,n} &\in \left[0.0001\,(\overrightarrow{x_{0,n}}-\overleftarrow{x_{0,n}}),\sqrt{\frac{(\overrightarrow{x_{0,n}}-\overleftarrow{x_{0,n}})^2}{-2\ln(G_L)}}\right],\\
w_j &\in \left[0.0001\left(\overrightarrow{t_0}-\overleftarrow{t_0}\right),\left(\overrightarrow{t_0}-\overleftarrow{t_0}\right)\right],
\end{aligned} \tag{24}$$

where $n = 1,2,\dots,N$ and $j = 1,2,\dots,M_3$. The presented starting pool selection algorithm bases partially on Cordón and Herrera's studies (1997b), but it also equips the staring pool with chromosomes supposed to be close to the desired ones, generated by means of the FCM. On the other hand, a part of the pool is generated randomly and heuristically to spread chromosomes through the intervals of performance and ensure pool diversity.

Two genetic operators were used during the generation of a new chromosome pool in each iteration of the ES: a max-min-arithmetic operator, described in-depth in (Herrera *et al.*, 1995), and periodic nonuniform mutation, based on Michalewicz' nonuniform mutation (Michalewicz, 2003), as a remedy for premature ES convergence. The mutation operator changes a single value in the chromosome with respect to the generation number and the corresponding DB parameter interval of performance, increasing or decreasing its value by

$$\Delta(l,d) = d\Big(1-\delta[0,1]^{\left(1-\frac{\mathrm{mod}\,(l,10)}{\mathcal{L}}\right)^{b}}\Big), \tag{25}$$

where l and $\mathcal{L}$ denote the current pool generation and maximum pool generation numbers, respectively, d stands for the interval of performance and $b > 0$ represents a mutation parameter. The selection of the operators was determined by both experiments by Cordón, Herrera, Lozano, Verdegay and the observed results confirming a proper choice. Also, attention was paid to the crossover operator as a tool to obtain offsprings located between parent individuals within the search domain.

A new chromosome pool consists of μ best chromosomes, selected from among the current pool and the mutated/crossed pool, according to the assessment made by a fitness function.

The fitness function f consists of five partial criteria,

$$f\,(R,\mathcal{R},\mathcal{E}_{\mathcal{C}}) = \overline{\Psi}_{N_T}\big(R^{(i)},\mathcal{E}_{\mathcal{C}}\big)\mathcal{C}_p\big(R^{(i)},\mathcal{E}_{\mathcal{C}}\big)\mathcal{C}_n\big(R^{(i)},\mathcal{E}_{\mathcal{C}}\big) \times \vartheta\big(R^{(i)},\mathcal{E}_{\mathcal{C}}\big)\Xi\,(R,\mathcal{R}). \tag{26}$$

Each criterion performs an assessment of a different context of the chromosome:

- $\overline{\Psi}_{N_T}\left(R^{(i)},\mathcal{E}_{\mathcal{C}}\right)$—the high rule compatibility degree, assessing the compatibility between the chromosome and the training data pairs $\mathcal{E}_{\mathcal{C}}$,

  $$\overline{\Psi}_{N_T}\left(R^{(i)},\mathcal{E}_{\mathcal{C}}\right) = \frac{1}{e}\sum_{j=1}^{e} R_c^{(i)}\big(\mathcal{E}_{\mathcal{C}}(j)\big), \tag{27}$$

  where $e = \mathrm{card}(\mathcal{E}_{\mathcal{C}})$;

- $\mathcal{C}_p\left(R^{(i)},\mathcal{E}_{\mathcal{C}}\right)$—the average covering degree of the subset $\mathcal{E}_{\mathcal{C}}$,

  $$\mathcal{C}_p\left(R^{(i)},\mathcal{E}_{\mathcal{C}}\right) = \sum_{\left(j\,|\,\mathcal{E}_{\mathcal{C}}(j)\in E_{\omega}^{+}(R^{(i)},\mathcal{E}_{\mathcal{C}})\right)} \frac{R_c^{(i)}(\mathcal{E}_{\mathcal{C}}(j))}{n_{\omega}^{+}}, \tag{28}$$

  where n_{ω}^{+} denotes the cardinality of the set E_{ω}^{+}, and E_{ω}^{+} represents the set of positive examples for the rule R_i, defined as

  $$E_{\omega}^{+}\left(R^{(i)},\mathcal{E}_{\mathcal{C}}\right) = \left\{\mathcal{E}_{\mathcal{C}}(j)\in\mathcal{E}_{\mathcal{C}}\mid R_c^{(i)}\big(R^{(i)},\mathcal{E}_{\mathcal{C}}(j)\big)\geq\omega\right\}, \tag{29}$$

  with ω standing for the minimal rule compatibility degree;

- $\mathcal{C}_n\left(R^{(i)},\mathcal{E}_{\mathcal{C}}\right)$—the rule consequence compatibility degree,

  $$\mathcal{C}_n\left(R^{(i)},\mathcal{E}_{\mathcal{C}}\right) = \begin{cases} 1 & \text{if}\quad n^{-}\leq\kappa n_{\omega}^{+},\\ \dfrac{1}{n^{-}-\kappa n_{\omega}^{+}+\exp(1)} & \text{if}\quad n^{-}>\kappa n_{\omega}^{+}, \end{cases} \tag{30}$$

  where $\kappa\in[0,1]$, n^{-} denotes the cardinality of the set

  $$E^{-}(R^{(i)},\mathcal{E}_{\mathcal{C}}) = \Big\{\mathcal{E}_{\mathcal{C}}(j)\in\mathcal{E}_{\mathcal{C}}\mid R_c^{(i)}\left(\mathcal{E}_{\mathcal{C}}(j)\right)=0\wedge\mu_{\underline{A}^{(i)}}\left(\underline{x_0}(j)\right)>0\Big\}; \tag{31}$$

- $\vartheta\left(R^{(i)},\mathcal{E}_{\mathcal{C}}\right)$—the width of membership functions vs. the intervals of performance,

$$\vartheta\left(R^{(i)},\mathcal{E}_{\mathcal{C}}\right) = \frac{\exp\left(1-\left(\frac{w^{(i)}}{t_0^+-t_0^-}+\sum_{n=1}^{N}\frac{2\sqrt{-0.5(\sigma_n^{(i)})^2\ln\left(G_L\right)}}{x_{0,n}^+-x_{0,n}^-}\right)\right)-1}{\exp(1)-1}, \tag{32}$$

where t_0^-, t_0^+, $x_{0,n}^-$ and $x_{0,n}^+$ can be obtained as

$$\begin{aligned} t_0^- &= \min_{j|\{\mathcal{E}_{\mathcal{C}}(j)\in\mathcal{E}_{\mathcal{C}}\}}\left(t_0(j)\right),\\ t_0^+ &= \max_{j|\{\mathcal{E}_{\mathcal{C}}(j)\in\mathcal{E}_{\mathcal{C}}\}}\left(t_0(j)\right),\\ x_{0,n}^- &= \min_{j|\{\mathcal{E}_{\mathcal{C}}(j)\in\mathcal{E}_{\mathcal{C}}\}}\left(x_{0,n}(j)\right),\\ t_{0,n}^+ &= \max_{j|\{\mathcal{E}_{\mathcal{C}}(j)\in\mathcal{E}_{\mathcal{C}}\}}\left(x_{0,n}(j)\right). \end{aligned} \tag{33}$$

Here $\sigma_n^{(i)}$ and $w^{(i)}$ denote the parameters of the premise and consequent membership functions, respectively. This criterion prevents the discovering of rules that may cover most of the training data set, thus dominating the whole KB;

- $\Xi\left(R,\mathcal{R}\right)$ assesses the similarity between the i-th rule $R^{(i)}$ and the RB $\mathcal{R}$, ensuring rule diversity in the KB,

$$\Xi\left(R^{(i)},\mathcal{R}\right) = 1-\max_{\left\{k|R^{(k)}\in\mathcal{R}\right\}}\left(\mu_{\underline{A}^{(i)}}(\underline{C_x}^{(k)})\star_{T_c}\mu_{B^{(i)}}(C_y^{(k)})\right). \tag{34}$$

Here $\underline{C_x}^{(k)}$ denotes a vector of DB parameters, representing Gaussian membership function centers for the premises of the k-th rule, $C_y^{(k)}$ represents the center of the consequence membership function for the k-th rule. To obtain the location of a moved consequent, each rule participating in the assessment was learned locally using the training data set E_{N_T} and excited using the vector of ones as input data to estimate the location of the consequent membership function and, in this way, to make the comparison feasible.

3.2. Rule Base Reduction. The KB obtained during the preceding step may contain useless or undesirable rules, and thus it must be reduced. It is necessary to ensure that the reduced KB contains a minimal set of rules that still cover the training data set E_{N_T}. During this process, only the RB is optimized and the DB remains unchanged. This step utilizes a simple genetic algorithm (Holland, 1975) using binary coding, proportional selection, two-point cross-over and a classic mutation. The GA was chosen to ensure an acceptable time of rule reduction in the case of an increasing complexity (huge training data sets, a high number of rules discovered in the first stage). Each chromosome represents an RB of the FIS. The chromosome fitness function performs an assessment of FIS quality and ensures a minimum covering level for all pairs in the training set. A chromosome consists of I fields, representing an ordered set of rules obtained during the rule discovery process (see Fig. 3). The binary value 1 denotes the rule (identified by a locus position) participating in an RB of the FIS. The DB remains unchanged during the entire reduction process, but to obtain an output FIS value, it is necessary to know the consequence parameters $\underline{\Theta}$. Thus, for each chromosome, a global consequence parameter learning method was used on the training data set E_{N_T}, see (16). The best chromosome (in terms of the fitness function) represents an FIS with a reduced RB, and thus a reduced KB that contains I_{II} rules.

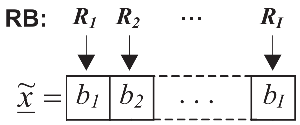

Fig. 3. Rule base coding.

The starting pool consists of $p\ \ (=M+1)$ chromosomes, where M chromosomes are generated randomly with uniform distribution on the RB. The other chromosome is generated as a vector of ones, representing the FIS formed of all rules obtained during the preceding rule discovery phase.

The fitness function f utilized during RB reduction is defined in the following way:

$$f\left(\tilde{x},(E_{N_T})\right) = \frac{1}{\sqrt{f_J\left(\tilde{x},(E_{N_T})\right)}},$$

$$f_J\left(\tilde{x},(E_{N_T})\right) = \begin{cases} J_{S_{\mathrm{RMSE}}}\left(\tilde{x},(E_{N_T})\right) & \text{if } J_{R_c}\left(\tilde{x},(E_{N_T})\right)\geq\tau,\\ \sum\limits_{j=1}^{N_T}\left(t_0(j)\right)^2 & \text{if } J_{R_c}\left(\tilde{x},(E_{N_T})\right)<\tau, \end{cases} \tag{35}$$

where J_{R_c} denotes the covering rate, defined as

$$J_{R_c}\left(\tilde{x},(\mathcal{E}_{\mathcal{C}})\right) = \frac{\sum\limits_{j=1}^{N_T}\sum\limits_{i=1}^{I_{\tilde{x}}}R_c^{(i)}\left(E_{N_T}(j)\right)}{N_T I_{\tilde{x}}}, \tag{36}$$

with $I_{\tilde{x}}$ as the chromosome $\tilde{x}$ rule cardinality, i.e., the number of rules represented by the chromosome. The

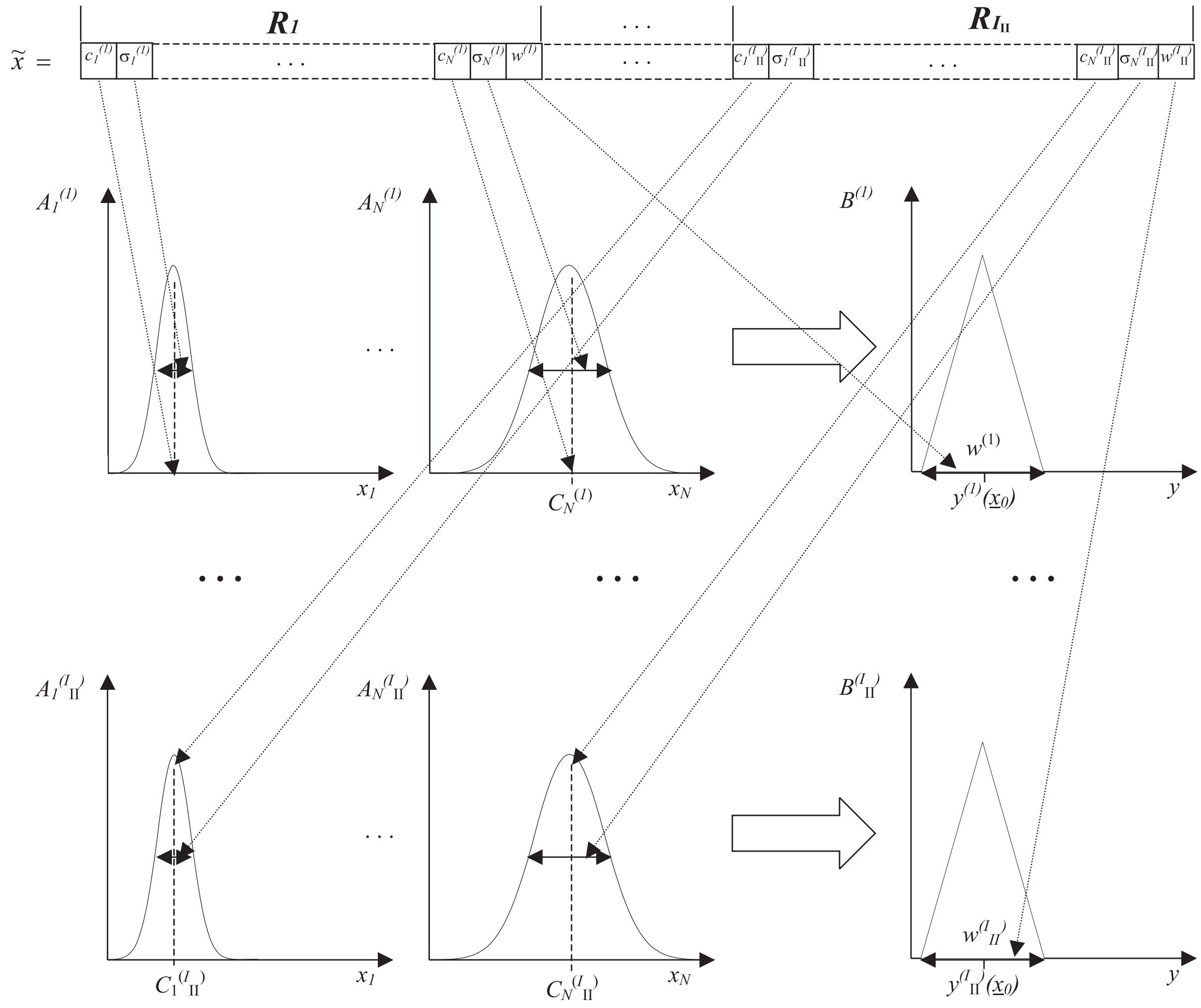

Fig. 4. Pittsburgh rule coding.

quality factor $J_{S_{\text{RMSE}}}$ utilizes the root mean square error, estimated as

$$J_{S_{\text{RMSE}}}(\tilde{x}, (E_{N_T})) = \frac{1}{2N_T} \sum_{j=1}^{N_T} \big(t_0(j) - y_0(x_0(j))\big)^2, \tag{37}$$

where $y_0(x_0(j))$ denotes the crisp output value generated by the FIS with the RB that suits the chromosome $\tilde{x}$ (see Fig. 3), obtained for the training data input vector $x_0(j)$.

It is also necessary to observe that the first stage of the presented method (rule discovery) also contains some elements of rule reducing, hidden within the employed fitness function: protection against discovering low firing strength rules (27), training data incompatible rules (28, 30) and redundant rules (34).

3.3. Fuzzy DB Tuning. The reduced RB obtained in the previous step and the corresponding DB both constitute a new KB $\mathcal{R}_{II}$. This KB underlies the following tuning process, described in this section. The tuning process is executed twice. For every but the first execution of the tuning process, the underlying KB is the best FIS obtained during the former run. The methodology utilizes a $(\mu + \lambda)$ elitist ES model, real coding using the Pittsburgh approach, max-min-arithmetic crossover, periodical nonuniform mutation and stochastic selection. The chromosome length is constant and thus the RB line-up is frozen during the tuning process. Only the DB parameters corresponding to the rules constituting the FIS are tuned. Those parameters represent only the premise and consequent fuzzy membership function sets (Gaussian and isoscele triangle ones, respectively). Assuming unconstrained free semantics, the chromosome vector consists of the $(2N+1)I_{II}$ real numbers that represent all but $\underline{\Theta}$ DB parameters describing the FIS (see Fig. 4). The matrix $\underline{\Theta}$ is estimated with the use of global learning, see (16), whenever it is necessary to obtain an FIS crisp output value. The estimation of the fitness function and the $\underline{\Theta}$ parameters is performed using the full training data set E_{N_T}.

The starting pool consists of μ chromosomes, and each of them is composed of an $\mathcal{R}_{II}$ RB with the respective DB obtained during the rule discovery process (for the first execution of the tuning process) or the best chromosome obtained during the former execution of the tuning phase. The other chromosomes are generated randomly with uniform distribution on intervals of performance,

$$\begin{aligned} \overleftarrow{c}_{0,i} &= c_{0,i} - \sqrt{\frac{(\sigma_{0,i})^2}{-2\ln(G_L)}}, \\ \overrightarrow{c}_{0,i} &= c_{0,i} - \sqrt{\frac{(\sigma_{0,i})^2}{-2\ln(G_L)}}, \\ \underline{\overleftarrow{c}}_0 &= \left[\overleftarrow{c}_{0,1}, \dots, \overleftarrow{c}_{0,N_{I_{II}}}\right]^{\mathsf{T}}, \\ \underline{\overrightarrow{c}}_0 &= \left[\overrightarrow{c}_{0,1}, \dots, \overrightarrow{c}_{0,N_{I_{II}}}\right]^{\mathsf{T}}, \end{aligned} \tag{38}$$

$$\begin{aligned} \overleftarrow{\sigma}_{0,i} &= 0.5\sigma_{0,i}, \\ \overrightarrow{\sigma}_{0,i} &= 1.5\sigma_{0,i}, \\ \underline{\overleftarrow{\sigma}}_0 &= \left[\overleftarrow{\sigma}_{0,1}, \dots, \overleftarrow{\sigma}_{0,N_{I_{II}}}\right]^{\mathsf{T}}, \\ \underline{\overrightarrow{\sigma}}_0 &= \left[\overrightarrow{\sigma}_{0,1}, \dots, \overrightarrow{\sigma}_{0,N_{I_{II}}}\right]^{\mathsf{T}}, \end{aligned} \tag{39}$$

$$\begin{aligned} \overleftarrow{w}_{0,i} &= 0.25w_{0,i}, \\ \overrightarrow{w}_{0,i} &= 1.25w_{0,i}, \\ \underline{\overleftarrow{w}}_0 &= \left[\overleftarrow{w}_{0,1}, \dots, \overleftarrow{w}_{0,N_{I_{II}}}\right]^{\mathsf{T}}, \\ \underline{\overrightarrow{w}}_0 &= \left[\overrightarrow{w}_{0,1}, \dots, \overrightarrow{w}_{0,N_{I_{II}}}\right]^{\mathsf{T}}. \end{aligned} \tag{40}$$

The fitness function defined below estimates the FIS quality factor by means of the root mean square error,

$$\begin{aligned} f\big(\tilde{x}, (E_{N_T})\big) &= \frac{1}{\sqrt[4]{f_J\left(\tilde{x}, (E_{N_T})\right)}}, \\ f_J\big(\tilde{x}, (E_{N_T})\big) &= \frac{1}{2N_T}\sum_{j=1}^{N_T}\big(t_0(j) - y_0\big(x_0(j)\big)\big)^2, \end{aligned} \tag{41}$$

where $\tilde{x}$ denotes the chromosome and $y_0\left(x_0(j)\right)$ is the FIS crisp output value.

The final output is the best chromosome (in terms of the fitness function) obtained during the last run of the ES.

4. Experimental Results

A series of experiments were performed using the presented method. The relevant software was written using the MATLAB 6.5 environment, running on a two-processor PC (P4 XEON@2.4 GHz, Hyper-Threading core) equipped with 4 GB RAM. As the MATLAB environment does not utilize a multi-threading technology, four parallel experiments were executed simultaneously (with different parameters, see Table 4).

For the experiments, the Box-Jenkins (Box and Jenkins, 1976) gas furnace benchmark database and the Sunspots (Weigend *et al.*, 1990) database were chosen. The system quality was measured using the root mean square error factor calculated between the training data output and the investigated FIS output,

$$J_{\mathrm{RMSE}} = \sqrt{\frac{1}{N_T}\sum_{j=1}^{N_T}\big(t_0(j) - y_0\big(x_0(j)\big)\big)^2}. \tag{42}$$

4.1. Data Approximation. The first experiment presents the ability to obtain an accurate FIS, performing an accurate data approximation on the Box-Jenkins benchmark database (using all pairs). The input data consists of 290 vectors organized in the following form:

$$\begin{aligned} &[\underline{x}_n, y_n] \\ &\quad = \big[y(n-1), \dots, y(n-4), x(n-1), \dots, x(n-6), y(n)\big]. \end{aligned} \tag{43}$$

The average $J_{\mathrm{RMSE}} = 0.1280$ (10 runs) for the Reichenbach fuzzy implication and the minimum t-norm was demonstrated. The FIS output compared with the training data set output is presented in Fig. 5. The comparison of J_{RMSE} obtained using different methods is displayed in Table 1—the other results were taken from (Box and Jenkins, 1976; Chen *et al.*, 1998; Czogała and Łęski, 1999; Kim *et al.*, 1997; Lin and Cunningham, 1995; Łęski, 2006; Pedrycz, 1984; Sugeno and Yasukawa, 1993; Tong, 1980; Wang and Langari, 1995; Xu and Lu, 1987; Zikidis and Vasilakos, 1996).

4.2. Knowledge Generalization. The related experiment presents the ability to generate an FIS capable of generalizing its knowledge. The Box-Jenkins benchmark data set, organized as presented in the former experiment, (see Eqn. (43)) was divided into two subsets: the training subset, composed of the first 100 pairs, and the testing data set, composed of the following 190 pairs. The presented division permits a reliable comparison of the presented method and others, i.e., the ones presented in (Łęski, 2006; Łęski and Czogała, 1999). For the Reichenbach fuzzy implication and the minimum t-norm, the average values $J_{\mathrm{RMSE}} = 0.0935$ and $J_{\mathrm{RMSE}} = 0.4423$ were demonstrated (10 runs, for the training data set and the testing data set, respectively). The FIS outputs is presented in Fig. 6 versus the training and testing data set outputs. The comparison of J_{RMSE} obtained using different methods is presented in Table 2—the other results are

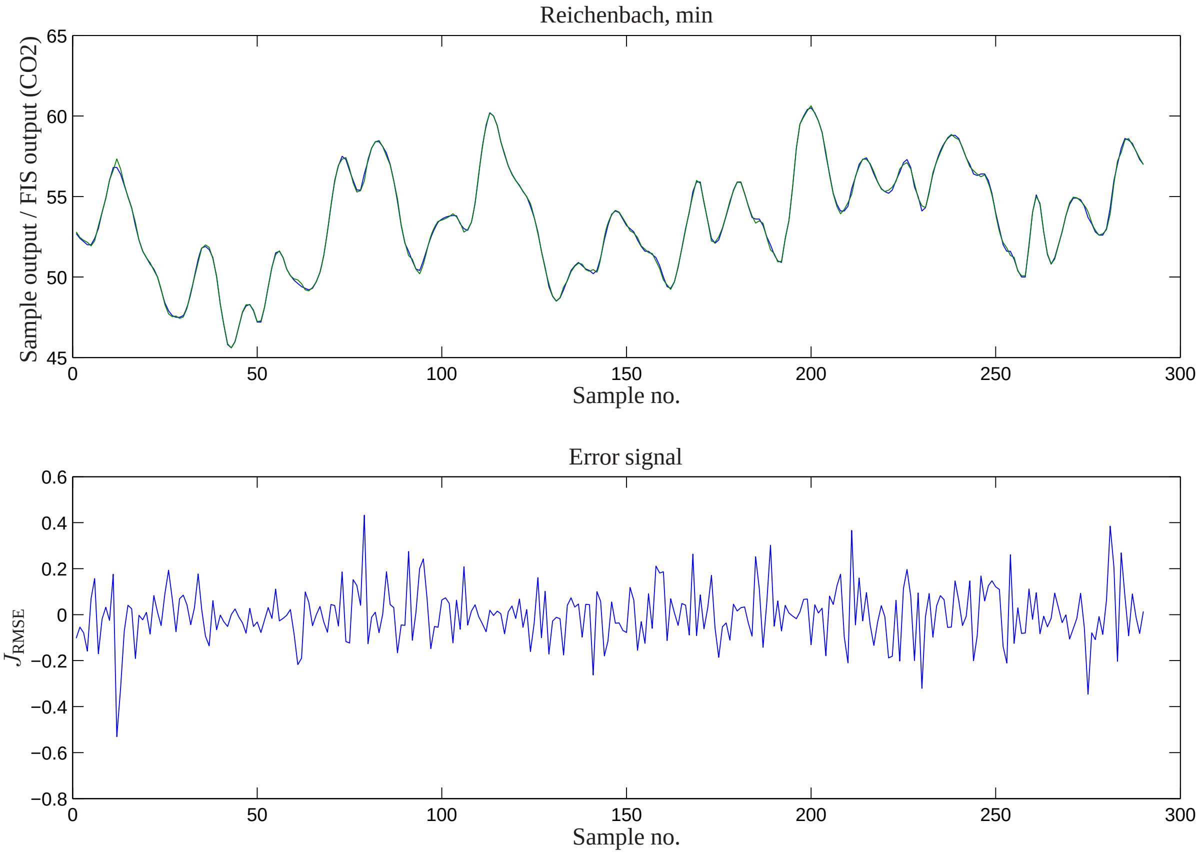

Fig. 5. Box-Jenkins data approximation.

Table 1. RMSE comparison obtained during the Box-Jenkins training data approximation test.

Author	Rules	J_{RMSE}
Tong	19	0.6848
Xu-Lu	25	0.5727
Pedrycz	81	0.5656
Box-Jenkins	—	0.4494
Sugeno-Yasukawa	6	0.4348
Chen *et al.*	3	0.2678
Lin-Cunningham	4	0.2664
Sugeno-Tanaka	2	0.2607
Wang-Langari	2	0.2569
Zikidis-Vasilakos	6	0.2530
Kim-Park-Ji	2	0.2345
Kim-Park	2	0.2190
ANBLIR(fuzzy imp. Gödel)	2	0.1892
ANNBFIS	3	0.1791
Czogała-Łęski	6	0.1445
ANBLIR(fuzzy imp. Fodor)	6	0.1353
The presented method	8	0.1280

Table 2. RMSE comparison obtained during the Box-Jenkins knowledge generalization test.

Author	Rules	J_{RMSE}
Least squares	6	0.5579
Iterative quadratic programming	6	0.5537
ϵ-Insensitive learning	6	0.4956
The presented method	7	0.4423
ANBLIR	8	0.4020
ANBLIR	7	0.3870
Iterative quadratic programming —local rule learning	2	0.3442

taken from (Łęski, 2005; Łęski, 2006). The results show that decreasing the number of rules usually increases the ability to generalize knowledge (thus, according to statistical learning theory, simultaneously decreasing the quality of data approximation).

The other experiment on chaotic time series prediction was performed using the Sunspots database (Weigend *et al.*, 1990). The data set contains time series tuples created as the numbers of sunspots observed from the year

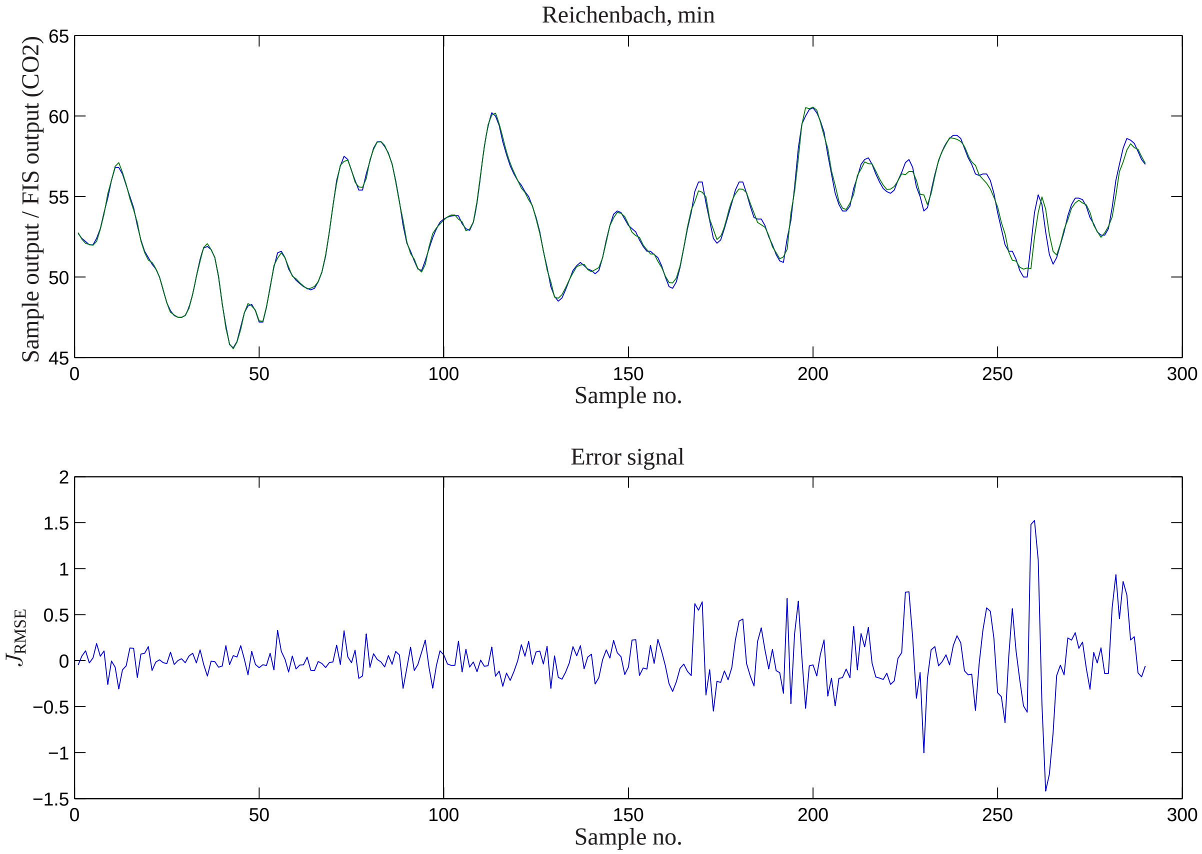

Fig. 6. Box-Jenkins knowledge generalization.

Table 3. RMSE comparison obtained during the Sunspots knowledge generalization test.

Author	Rules	J_{RMSE}
Least squares	3	0.08416
Iterative quadratic programming	3	0.08032
ϵ-Insensitive learning	4	0.07945
Iterative quadratic programming —local rule learning	5	0.07803
The presented method	8	0.07780

1700 to 1979 in the following form:

$$[\underline{x}_n, y_n] = [y(n–1), \dots, y(n–2), x(n–3), \dots, x(n–12), x(n)]. \tag{44}$$

The first 100 tuples constitute the training data set while the following 168 tuples form the testing data set. The FIS outputs are presented in Fig. 7 versus the training and testing data set outputs. The values of $J_{\mathrm{RMSE}} = 0.0608$ and $J_{\mathrm{RMSE}} = 0.0778$ were observed for the training and testing data sets, respectively. This ranks the presented method among the best ones developed. A comparison with other methods is shown in Table 3. All the experiments were performed using the parameters presented in Table 4.

Table 4. Method execution parameters.

Parameter	Phase 1	Phase 2	Phase 3
μ	60	—	100
λ	20	—	20
p	—	50	—
$\mathcal{L}$	50	150	150
Crossover prob.	1	0.75	0.6
Mutation prob.	0.8	0.005	0.6
G_L	0.25	—	0.9
τ	—	0.25	—
ω	0.1	—	—
κ	0.05	—	—
b	5	—	5
ϵ	0.9–1.8	—	—

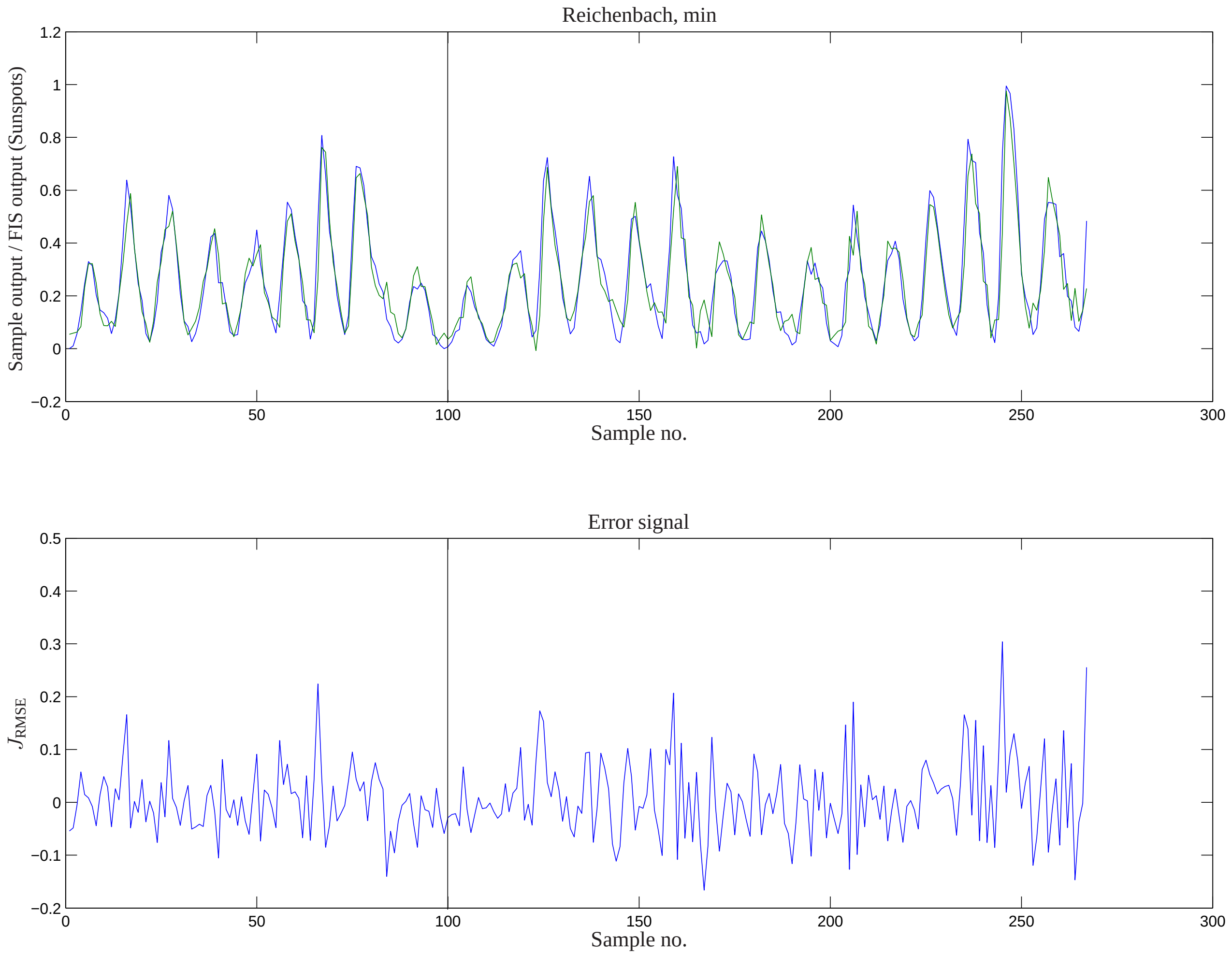

Fig. 7. Sunspots chaotic time-series prediction.

5. Summary and Further Work

The presented method still needs some refinement. During the calculation of the $\underline{\Theta}$ parameters, it is common that nearly singular matrices have to be inverted, thus leading to a rapid growth of the calculation error. It is considered to reorganize the matrix inversion method to a recursive implementation, rather than the iterative one. Other possible t-norms (e.g., the algebraic product) and other fuzzy implications will be the subjects of further research. Other benchmark tests, e.g., a Mackey-Glass chaotic time series prediction, Boston Housing system identification, or other image and pattern recognition data are also considered. Future work is aimed at preparing a system with extra degrees of freedom, like a variable number of rules, exploring different genetic operators, tuning algorithm parameters and working with FISs that utilize membership functions different than Gaussian and isoscele triangles.

Acknowledgments

The author is grateful to the two anonymous referees for their productive comments that helped him to improve the readability and value of the paper.

References

Angelov P. (2002): *Evolving Rule-Based Models. A Tool for Design of Flexible Adaptive Systems.* — Wurzburg: Physica-Verlag.

Arabas J. (2001): *Lectures on Evolutionary Algorithms.* — Warsaw: Wydawnictwa Naukowo-Techniczne, (in Polish).

Baron L., Achiche S. and Balazinski M. (2001): *Fuzzy decision support system knowledge base generation using a genetic algorithm.* — Int. J. Approx. Reason., Vol. 1, No. 28, pp. 125–148.

Bezdek J. (1981): *Pattern Recognition with Fuzzy Objective Function Algorithms.* — New York: Plenum Press.

Bonarini A. (1996): *Evolutionary learning of fuzzy rules: Competition and cooperation*, In: Fuzzy Modelling: Paradigms and Practice (W. Pedrycz, Ed.). — Norwell: Kluwer.

Box G. and Jenkins G. (1976): *Time Series Analysis. Forecasting and Control.* — San Francisco: Holden-Day.

Carse B., Fogarty T.C. and Munro A. (1996): *Evolving fuzzy rule based controllers using genetic algorithms.* — Fuzzy Sets Syst., Vol. 80, No. 3, pp. 273–294.

Chen J.Q., Xi Y.G. and Zhang Z.J. (1998): *A clustering algorithm for fuzzy model identification.* — Fuzzy Sets Syst., Vol. 98, No. 3, pp. 319–329.

Cordón O. and Herrera F. (1997a): *Identification of linguistic fuzzy models by means of genetic algorithms*, In: Fuzzy Model Identification. Selected Approaches (D. Driankow and H. Hellendoorn, Eds.). — Berlin: Springer.

Cordón O. and Herrera F. (1997b): *A three-stage evolutionary process for learning descriptive and approximative fuzzy logic controller knowledge bases from examples.* — Int. J. Approx. Reason., Vol. 17, No. 4, pp. 369–407.

Cordón O. and Herrera F. (2001): *Hybridizing genetic algorithms with sharing scheme and evolution strategies for designing approximate fuzzy rule-based systems.* — Fuzzy Sets Syst., Vol. 118, No. 2, pp. 235–255.

Cordón O., Del Jesús M., Herrera F. and Lozano M. (1999): *MOGUL: A methodology to obtain genetic fuzzy rule-based systems under the iterative rule learning approach.* — Int. J. Intell. Syst., Vol. 14, No. 11, pp. 1123–1153.

Cordón O., Herrera F., Hoffmann F. and Magdalena L. (2001): *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases.* — Singapore: World Scientific.

Cordón O., Gomide F., Herrera F., Hoffmann F. and Magdalena L. (2004): *Ten years of genetic fuzzy systems: Current framework and new trends.* — Fuzzy Sets Syst., Vol. 141, No. 1, pp. 5–31.

Czogała E. and Łęski J. (1996): *A new fuzzy inference system with moving consequent in if-then rules. Application to pattern recognition.* — J. Bull. Polish Acad. Sci., Vol. 45, No. 4, pp. 643–655.

Czogała E. and Łęski J. (1999): *Fuzzy and Neuro-Fuzzy Intelligent Systems.* — Heidelberg: Physica-Verlag.

Fuller R. (1999): *Introduction to Neuro-Fuzzy Systems.* — Würzburg: Physica-Verlag.

González A. and Pérez R. (1999): *SLAVE: A genetic learning system based on an iterative approach.* — IEEE Trans. Fuzzy Syst., Vol. 7, No. 2, pp. 176–191.

Herrera F. and Verdegay J. (1996): *Genetic Algorithms and Soft Computing.* — Wurzburg: Physica-Verlag.

Herrera F., Lozano M. and Verdegay J.L. (1995): *Tuning fuzzy logic controllers by genetic algorithms.* — Int. J. Approx. Reason., Vol. 12, No. 3, pp. 299–315.

Hoffmann F. and Pfister G. (1997): *Evolutionary design of a fuzzy knowledge base for a mobile robot.* — Int. J. Approx. Reason., Vol. 17, No. 4, pp. 447–469.

Holland J. (1975): *Adaptation in Natural and Artificial Systems.* — University of Michigan Press, Ann Arbor.

Holland J. and Reitman J. (1978): *Cognitive systems based on adaptive algorithms*, In: Pattern-Directed Inference Systems (D.A. Waterman and F. Hayes-Roth, Eds.). — New York: Academic Press.

Ishibuchi H., Nakashima T. and Murata T. (1999): *Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems.* — IEEE Trans. Syst. Man Cybern., Vol. 29, No. 5, pp. 601–618.

Kim E., Park M. and Ji S. (1997): *A new approach to fuzzy modeling.* — IEEE Trans. Fuzzy Syst., Vol. 5, No. 3, pp. 328–337.

Lee M.A. and Takagi H. (1993): *Integrating design stages of fuzzy systems using genetic algorithms.* — Proc. 2nd IEEE Int. Conf. *Fuzzy Systems*, San Francisco, CA, pp. 613–617.

Łęski J. (2005): *TSK-fuzzy modeling based on ε-insensitive learning.* — IEEE Trans. Fuzzy Syst. Vol. 13, No. 2, pp. 181–193.

Łęski J. (2006): *Neuro-Fuzzy Systems.* — Warsaw: Wydawnictwa Naukowo-Techniczne, (in Polish).

Łęski J. and Czogała E. (1999): *A new artifficial neural network based fuzzy inference system with moving consequents in if-then rules and selected applications.* — Fuzzy Sets Syst., Vol. 108, No. 3, pp. 289–297.

Lin Y. and Cunningham H. (1995): *A new approach to fuzzy-neural modeling.* — IEEE Trans. Fuzzy Syst., Vol. 3, No. 2, pp. 190–197.

Magdalena L. and Monasterio F. (1997): *A Fuzzy logic controller with learning through the evolution of its knowledge base.* — Int. J. Approx. Reason., Vol. 16, Nos. 3–4, pp. 335–358.

Mamdani E. and Assilian S. (1975): *An experiment in linguistic synthesis with a fuzzy logic controller.* — Int. J. Man-Mach. Stud., Vol. 7, No. 1, pp. 1–13.

Michalewicz Z. (2003): *Genetic Algorithms + Data Structures = Evolution Programs.* — Warsaw: Wydawnictwa Naukowo-Techniczne, (in Polish).

Park D., Kandel A. and Langholz G. (1994): *Genetic-based new fuzzy reasoning models with application to fuzzy control.* — IEEE Trans. Syst. Man Cybern., Vol. 24, No. 1, pp. 39–47.

Parodi A. and Bonelli P. (1993): *A new approach to fuzzy classifier systems.* — Proc. 5-th Int. Conf. *Genetic Algorithms*, Los Altos, pp. 223–230.

Pedrycz W. (1984): *An identification algorithm in fuzzy relational systems.* — Fuzzy Sets Syst., Vol. 13, No. 2, pp. 153–167.

Pedrycz W. (1997): *Fuzzy Evolutionary Computation.* — Dordrecht: Kluwer.

Pham D. and Karaboga D. (1991): *Optimum design of fuzzy logic controllers using genetic algorithms.* — J. Syst. Eng., Vol. 1, No. 2, pp. 114–118.

Sugeno M. and Kang G. (1988): *Structure identification of fuzzy model.* — Fuzzy Sets Syst., Vol. 28, No. 1, pp. 15–33.

Sugeno M. and Yasukawa T. (1993): *A fuzzy-logic based approach to qualitative modeling.* — IEEE Trans. Fuzzy Syst., Vol. 1, No. 1, pp. 7–31.

Tadeusiewicz R. (1998): *Fundamental introduction to neural neworks techniques with sample implementations.* — Warsaw: Akademicka Oficyna Wydawnicza PLJ, (in Polish).

Takagi T. and Sugeno M. (1985): *Fuzzy identification of systems and its application to modelling and control.* — IEEE Trans. Syst. Man Cybern., Vol. 15, No. 1, pp. 116–132.

Thrift P. (1991): *Fuzzy logic synthesis with genetic algorithms.* — Proc. 4-th Int. Conf. *Genetic Algorithms*, Los Altos, pp. 509–513.

Tong R. (1980): *The evaluation of fuzzy models derived from experimental data.* — Fuzzy Sets Syst., Vol. 4, pp. 1–12.

Valenzuela-Rendón M. (1991): *The fuzzy classifier system: Motivations and first results.* — Proc. 1-st Int. Conf. *Parallel Problem Solving from Nature*, Berlin, pp. 330–334.

Velasco J. (1998): *Genetic-based on-line learning for fuzzy process control.* — Int. J. Intell. Syst., Vol. 13, Nos. 10-11, pp. 891–903.

Wang L. and Langari R. (1995): *Building Sugeno-type models using fuzzy discretization and orthogonal parameter estimation techniques.* — IEEE Trans. Fuzzy Syst., Vol. 3, No. 4, pp. 454–458.

Weigend A., Huberman B. and Rumelhart D. (1990): *Predicting the future: A connectionist approach.* — Int. J. Neural Syst., Vol. 1, No. 3, pp. 193–209.

Xu C. and Lu Y. (1987): *Fuzzy modeling identification and selflearining for dynamic systems.* — IEEE Trans. Syst. Man Cybern., Vol. 17, No. 4, pp. 683–689.

Zadeh L. (1971): *Towards a theory of fuzzy systems*, In: Aspects of Network and System Theory (R.E. Kalman and N. De Claris, Eds.). — New York: Holt, Rinehart and Winston.

Zikidis K. and Vasilakos A. (1996): *ASAFES2: A novel, neurofuzzy architecture for fuzzy computing, based on functional reasoning.* — Fuzzy Sets Syst., Vol. 83, No. 1, pp. 63–68.

Received: 23 November 2005
Revised: 2 July 2006