

# Evolutionary algorithms for the selection of time lags for time series forecasting by fuzzy inference systems

Kristina Lukoseviciute, Minvydas Ragulskis \*

Research Group for Mathematical and Numerical Analysis of Dynamical Systems, Kaunas University of Technology, Studentu 50-222, Kaunas LT-51638, Lithuania

## ARTICLE INFO

### Article history:

Received 2 March 2009  
Received in revised form  
18 February 2010  
Accepted 23 February 2010  
Communicated by G.P. Zhang  
Available online 25 March 2010

### Keywords:

Time series forecasting  
Fuzzy inference system  
Evolutionary algorithm  
Attractor embedding

## ABSTRACT

Time series forecasting by fuzzy inference systems based on optimal non-uniform attractor embedding in the multidimensional delay phase space is analyzed in this paper. A near-optimal set of time lags is identified by evolutionary algorithms after the optimal dimension of the reconstructed phase space is determined by the FNN (false nearest neighbors) algorithm. The fitness function is constructed in such a way that it represents the spreading of the attractor in the delay coordinate space but does not contain any information on prediction error metrics. The weighted one-point crossover rule enables an effective identification of near-optimal sets of non-uniform time lags which are better than the globally optimal set of uniform time lags. Thus the reconstructed information on the properties of the underlying dynamical system is directly elaborated in the fuzzy prediction system. A number of numerical experiments are used to test the functionality of this method.

© 2010 Elsevier B.V. All rights reserved.

## 1. Introduction

Time series forecasting by fuzzy inference systems is one of many diverse methods used for extrapolating past behavior into future [1–8]. In general, the object of these techniques is to build a model of the process and then use this model on the last values of the time series to extrapolate past behavior into future. Attractor embedding helps to reveal properties of a dynamical system represented by a scalar time series and can be used to increase the accuracy of prediction [9]. In general, attractor embedding is a nonlinear analysis procedure and comprises two basic steps: determination of the minimal dimension of the reconstructed delay phase space and the optimal time lag [10,11].

Non-uniform attractor embedding when time lags between adjacent coordinates are not necessarily equal can yield more accurate representations of properties of the analyzed dynamical system [9]. Such non-uniform embedding techniques have been successfully exploited in time series forecasting [12]. A feedforward neural network with structural learning is exploited for identification of a near-optimal set of time lags in [12]. Alternatively, a deterministic method for the selection of optimal time lags for non-uniform embedding is proposed in [13]. This method emphasizes the multidimensionality of the problem, is able to cope with optimization problems in a multi-parameter

space of arguments and is well applicable in time series forecasting routines based on fuzzy inference systems [13].

But the optimization procedure for the identification of the optimal set of non-uniform time lags can be computationally hard when signals contaminated with noise are analyzed. Optimal embedding dimension of such signals is much higher compared to deterministic signals [10]. Thus, the selection of the optimal set of non-uniform time lags becomes a difficult computational task due to the reason that the number of different combinations of time lags becomes very high. The objective of this paper is to adapt evolutionary algorithms for the selection of a near-optimal set of time lags which could be exploited for time series forecasting by fuzzy inference systems. This is in contrary to structural learning techniques used in [12]. We quickly identify a near-optimal set of time lags but still keep the deterministic approach for the description of the reconstructed dynamical system.

It can be noted that our approach is quite different from direct applications of evolutionary algorithms in predictions based on fuzzy inference systems. For example, the fuzzy model integrates various factors with influential degree as the input variables, and the evolutionary algorithm adjusts the influential degree of each input variable dynamically in [14]. An adaptive network-based fuzzy inference system with the evolutionary learning algorithm is used to predict the workpiece surface roughness for the end milling process in [15] where the hybrid Taguchi-evolutionary learning algorithm is applied to find optimal system parameters by directly minimizing the root-mean-squared-error performance criterion. On the contrary, evolutionary algorithms do not influence the prediction process directly in our approach.

\* Corresponding author. Tel.: +370 69822456.

E-mail address: [minvydas.ragulskis@ktu.lt](mailto:minvydas.ragulskis@ktu.lt) (M. Ragulskis).

We use evolutionary algorithms to find the near-optimal set of time lags, while the criterion determining the quality of a nonlinear embedding is kept deterministic. Such an approach preserves the structure of the concept based on non-uniform embedding of chaotic attractors and is well applicable even for time series with multiple time scales when the state of the system changes in time and variable embedding is necessary to provide optimal system's description in the reconstructed phase space.

In general, the idea to use evolutionary procedures to determine time lag sets is not new. The optimal selection of time lags for temporal decorrelation source separation based on genetic algorithms is proposed in [16]. The time-delay added evolutionary forecasting approach is proposed in [17] where an evolutionary search for the minimum dimension of the delay phase space is performed. The morphological-rank-linear time-lag added evolutionary forecasting method is proposed in [18] where a modified genetic algorithm searches for the particular time lags capable of a fine tuned characterization of the time series. The novelty of our method lies in the fact that our technique first determines the optimal dimension of the reconstructed phase space, and then finds a near-optimal set of time lags using a fitness function based on the attractor's metrics in the multi-dimensional phase space. Thus the reconstructed information on the properties of the underlying dynamical system is directly elaborated in the fuzzy prediction system.

This paper is organized as follows: Section 2 describes a fitness estimate of a set of non-uniform time lags. The structure of evolutionary algorithms used for the identification of a near-optimal set of time lags is discussed in Section 3. The selection of parameters of evolutionary algorithms is discussed in Section 4. Predictions of different time series are presented in Section 5 and concluding remarks are discussed in the final Section.

## 2. A fitness estimate for a set of non-uniform time lags

Suppose that a scalar time series is given as  $x_1, x_2, \dots, x_N$ . Then time-delay vectors produced by a non-uniform embedding can be reconstructed as follows ([10]):

$$y_p^{(d)} = (x_p, x_{p+\tau_1}, \dots, x_{p+\tau_1+\dots+\tau_{d-1}}); \quad p = 1, 2, \dots, N - \sum_{i=1}^{d-1} \tau_i, \quad (1)$$

where  $d$  is the embedding dimension of the delay coordinate space;  $y_p^{(d)}$  is the  $p$ -th reconstructed vector with the embedding dimension  $d$ ;  $\tau_i \in \mathbb{N}$ . If the time span between two adjacent points in the scalar time series is  $\delta$  (the sampling rate is  $1/\delta$ ),  $\delta \sum_{i=1}^{d-1} \tau_i$  is the total embedding window. The function of embedding quality is defined as the average of parameters of embedding quality in all possible different planar projections in [13]:

$$Q(\tau_1, \dots, \tau_{d-1}, \omega) = \frac{2}{d(d-1)} \left( \sum_{i=1}^{d-1} |\sin(\omega \delta \tau_i)| + \sum_{i=1}^{d-2} |\sin(\omega \delta (\tau_i + \tau_{i+1}))| + \dots + \left| \sin \left( \omega \delta \sum_{j=1}^{d-1} \tau_j \right) \right| \right) \quad (2)$$

In case of a uniform embedding the function of embedding quality reads

$$Q(\tau, \dots, \tau, \omega) = \frac{2}{d(d-1)} \sum_{k=1}^{d-1} (d-k) Q_k = \frac{2}{d(d-1)} \sum_{k=1}^{d-1} (d-k) |\sin(k\omega\tau\delta)| \quad (3)$$

It is clear that  $0 \leq Q(\tau, \omega) \leq 1$ . Equality  $Q=0$  means that a scalar time series is compressed to line segments in all possible

projections. Every harmonic component of the original time series will be affected by the function of embedding quality when the original time series will be embedded into a  $d$ -dimensional delay coordinate space. Harmonic components with frequencies where  $Q$  is small will be suppressed (in average) in all plane projections of the delay coordinate space. Similarly, harmonic components with frequencies where  $Q$  is high will experience rich representation in the reconstructed phase space. Keeping those considerations an objective function  $F$  which characterizes the magnitude of the attractor's spreading in the embedded space is defined in [13]

$$F(\tau_1, \dots, \tau_{d-1}) = \frac{\pi}{2} \frac{\int_0^\infty A(\omega) Q(\tau_1, \dots, \tau_{d-1}, \omega) d\omega}{\int_0^\infty A(\omega) d\omega}, \quad (4)$$

where  $A(\omega)$  is Fourier amplitude spectrum of the original time series. We calculate indefinite integrals over the whole frequency range (in practice one should compute definite integrals up to a preset upper frequency bound). The integral of  $A(\omega)$  in the denominator is used in order to normalize the magnitude of the objective function (this integral can be computed once for a concrete time series before the optimization procedure is commenced). We will use the numerical value of the objective function in Eq. (4) as a fitness estimate of a set of non-uniform time lags.

As noted previously, definite integrals in Eq. (4) are calculated up to a preset upper frequency bound. Similarly, we present an upper bound also for possible numerical values of time delays

$$1 \leq \tau_i \leq T; \quad i = 1, 2, \dots, d-1; \quad T \in \mathbb{N} \quad (5)$$

As shown in [13], the number of different combinations of time lags is  $T^{d-1}$  (the number of different combinations without permutations is  $(T+d-2)!/((d-1)!(T-1)!)$ ). Full sorting becomes a difficult computational task when the dimension of the reconstructed phase space becomes high. For example 278 256 different combinations of time lags without permutations must be sorted when  $d=6$  and  $T=30$ ; it is clear that full sorting is really an unfeasible approach for higher dimensions of the phase space. It is necessary to construct near-optimal inexpensive integer programming solution strategies in such cases. We modify and adapt classical evolutionary algorithms for the selection of a near-optimal set of time lags which will be used for time series prediction using an adaptive network-based fuzzy inference system (ANFIS) as illustrated in Fig. 1. We use a five layers feedforward standard ANFIS system in Matlab FuzzyLogic Toolbox. ANFIS inputs are time lagged values of the original scalar time series, the single output is the forecasted and forward-lagged value of the time series; the horizon of

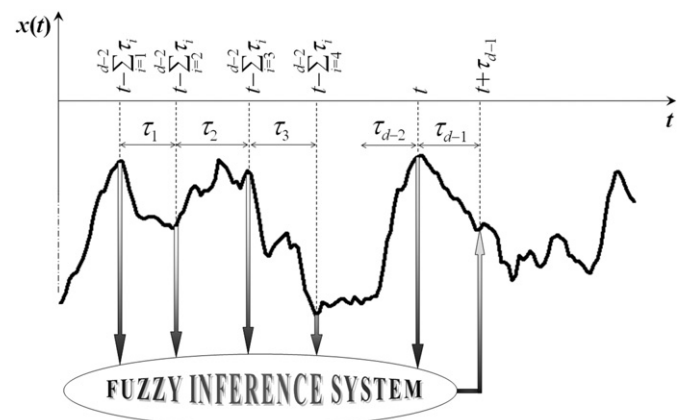


Fig. 1. A schematic diagram illustrating the functionality of the time series prediction algorithm based on a fuzzy inference system and non-uniform embedding.

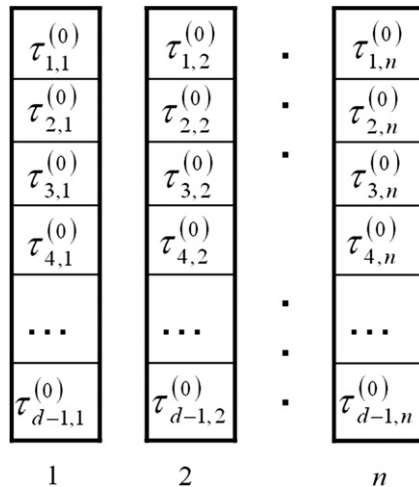
prediction is  $\tau_{d-1}$ . Two membership functions are assigned to each input, the rule number depends from the number of inputs and equals to  $2^{d-1}$ . ANFIS learning rules are described in detail in [19].

It can be noted that evolutionary algorithms, in particular genetic algorithms, are widely used in global optimization tasks. A single front genetic algorithm is used for parallel multi-objective optimization in [20]. A population-based immune-inspired algorithms are used for global numerical optimization in [21]. The optimal selection of time lags for temporal decorrelation source separation based on genetic algorithms is proposed in [16]. We also adapt a classical genetic algorithm for the identification of a near-optimal set of time lags.

### 3. The structure of evolutionary algorithms

We construct an evolutionary algorithm in such a way that every chromosome represents an array of time lags (genes). Every gene is an integer number; the length of each chromosome is  $(d-1)$ . We assume, that  $T=30$ . The parameter  $T$  is in fact not a parameter of the evolutionary algorithm (though it has a direct influence to the functionality of these algorithms).  $T$  defines the maximum value of a time delay used in the search algorithm (Eq. (5)). Since the optimal dimension of the phase space often equals to 6 or even a higher number (see Section 5), the embedding window (the number of points between the first ANFIS input and the output) would be  $(d-1)T=150$  at  $d=6$  and  $T=30$ . A number of data vectors must be used for training the ANFIS system. That sets a certain constrain for the minimum length of the analyzed time series. On the other hand, computational experiments show that the components of the near-optimal set of time lags are rarely higher than 15. That justifies the selection  $T=30$  for all analyzed time series in Section 5.

Initial population comprises  $n$  chromosomes with randomly generated values of genes (Fig. 2). As noted previously, the objective function  $F$  characterizes the magnitude of the attractor's spreading in the embedded space (Eq. (4)) and is used as fitness value (which is a real number) associated with every chromosome. An even number of chromosomes is selected to the mating population from the initial population. We use a random roulette method for the selection of chromosomes. The higher is the fitness value of a chromosome, the higher is a chance that this chromosome will be selected to the mating population. Nevertheless, a probability that a chromosome with a low fitness

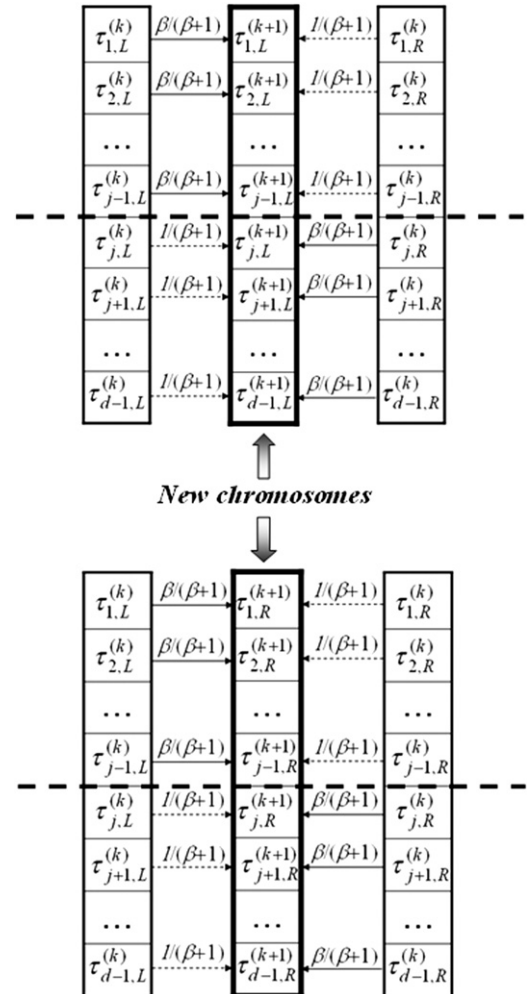


**Fig. 2.** A schematic diagram illustrating the initial population of  $n$  chromosomes;  $\tau_{ij}^{(0)}$  denotes an  $i$ -th element of a  $j$ -th chromosome;  $\tau_{ij}^{(0)} \in \mathbb{N}$ ;  $1 \leq \tau_{ij}^{(0)} \leq T$ ;  $1 \leq i \leq d-1$ ;  $1 \leq j \leq n$ .

value will be selected is not zero. Also, several copies (clones) of the same chromosome can be selected to the mating population. All chromosomes are grouped into pairs after the selection process to the mating population is over.

The crossover between two chromosomes in a pair is executed for all pairs in the mating population. We use a one-point crossover method [22] (the location of the point is random for every pair). Moreover, we modify the classical crossover algorithm by introducing the parameter of similarity  $\beta$  ( $\beta \geq 0$ ). The role of the parameter of similarity in the procedure of crossover is illustrated in Fig. 3. We do not only exchange genes, but also let two new chromosomes to become more or less similar to each other and to their parent chromosomes. Such a crossover algorithm extended by the parameter of similarity converges to the classical crossover algorithm when  $\beta$  tends to zero or to infinity (both offspring chromosomes are equal when  $\beta=1$ ). In general, this algorithm can be described by the following equality:

$$\begin{aligned} \tau_{j,L}^{(k+1)}|_- &= \text{round}\left(\frac{\beta}{\beta+1}\tau_{j,L}^{(k)} + \frac{1}{\beta+1}\tau_{j,R}^{(k)}\right); \\ \tau_{j,R}^{(k+1)}|_- &= \text{round}\left(\frac{1}{\beta+1}\tau_{j,L}^{(k)} + \frac{\beta}{\beta+1}\tau_{j,R}^{(k)}\right); \\ \tau_{j,L}^{(k+1)}|_+ &= \text{round}\left(\frac{1}{\beta+1}\tau_{j,L}^{(k)} + \frac{\beta}{\beta+1}\tau_{j,R}^{(k)}\right); \\ \tau_{j,R}^{(k+1)}|_+ &= \text{round}\left(\frac{\beta}{\beta+1}\tau_{j,L}^{(k)} + \frac{1}{\beta+1}\tau_{j,R}^{(k)}\right), \end{aligned} \quad (6)$$



**Fig. 3.** A schematic diagram illustrating the crossover algorithm;  $\beta$  is the parameter of similarity. The dashed line represents the crossover point.

where  $k$  is the generation number;  $\tau_{j,L}^{(k)}$  is a  $j$ -th gene of the left parent chromosome in the pair;  $\tau_{j,R}^{(k)}$  is a  $j$ -th gene of the right parent chromosome in the pair;  $\tau_{j,L}^{(k+1)}$  is the  $j$ -th gene of the left offspring chromosome;  $\tau_{j,R}^{(k+1)}$  is the  $j$ -th gene of the right offspring chromosome; the minus sign in the subscript denotes that the  $j$ -th gene is above the crossover point; the plus sign in the subscript denotes that the  $j$ -th gene is below the crossover point. In any case,  $j$ -th genes of offspring chromosomes will fit into interval  $[\tau_{j,L}^{(k)}; \tau_{j,R}^{(k)}]$ .

We also introduce a crossover coefficient  $\kappa$  which characterizes a probability that the crossover procedure will be executed for a pair of chromosomes in the mating population. It can be noted that this crossover procedure is one of the factors helping to avoid incest, which is especially dangerous to our computational setup. Genes in the initial population of chromosomes are generated as random integer numbers. A crossover without domination would not produce new genes (new time delays). Even a full sorting procedure in the subspace of generated genes would produce a set of time delays which can be far away from the global optimal set.

The crossover procedure with the parameter of similarity can generate new genes. Anyway, this procedure would not be able to cover the whole space of arguments if the initial set of genes does not span over all possible time delays from 1 to  $T$  (strictly speaking, this is only necessary, but not sufficient condition). We will prove this statement by contradiction. Let's assume that the minimum value of all genes in the initial population is  $a$ ; the maximum value of all genes is  $b$ . Whatever is the value of the parameter  $\beta$ , new genes in any of offspring chromosomes will be located in the interval  $[a; b]$ . Moreover, the crossover procedure with domination will never produce a chromosome with a gene  $a$  (the minimum time delay) in a different location (inside an offspring chromosome) than it was originally located in a chromosome from the initial population. In other words, the maximum possible subspace of time delay vectors covered by the crossover procedure with domination is

$$\tau_{i\min} = \min_{1 \leq j \leq n} (\tau_{ij}^{(0)}); \quad \tau_{i\max} = \max_{1 \leq j \leq n} (\tau_{ij}^{(0)}); \quad i = 1, 2, \dots, d-1. \quad (7)$$

Therefore, we use a mutation procedure which prevents convergence on one local solution and helps to seek the global solution. It is a classical mutation procedure. The mutation parameter  $\mu$  ( $0 \leq \mu < 1$ ) determines the intensity of the mutation process. We run through genes of all chromosomes in the  $k$ -th generation and generate a random number evenly distributed in an interval  $[0; 1]$  for every gene. The current gene is changed by a random integer from an interval  $[1; T]$  if the random number generated for this gene is lower than  $\mu$ . Usually the value of the mutation parameter  $\mu$  is low ( $\mu < 0.01$ ). Higher intensity of mutation may prevent the convergence of evolutionary algorithms.

#### 4. The selection of parameters of evolutionary algorithms

In general, the selection of parameters of evolutionary algorithms is an empirical process, though some common principles are described in [22,23]. In [13] we have shown that the best set of time lags for uniform embedding of Mackey–Glass chaotic time series is  $\{9, 9, 9, 9\}$ ;  $F(9, 9, 9, 9) = 1.0990$  (the embedding dimension  $d=6$ ). Thus, the goal of the evolutionary algorithm is to find such a set of time lags which would produce the value of the fitness function at least greater than 1.0990 (for Mackey–Glass time series). We will use these results as a criterion to identify optimal parameters of the evolutionary algorithm. It can be noted that the Mackey–Glass chaotic time series has

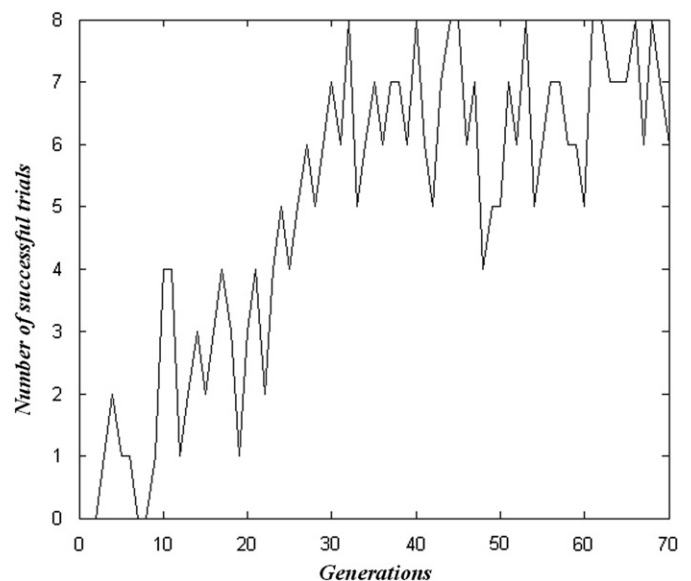
been extensively studied in time series prediction routines [24–26].

The following parameters of the evolutionary algorithm must be pre-selected: the crossover coefficient  $\kappa$ ; the mutation parameter  $\mu$ ; the parameter of similarity  $\beta$ ; the size of population  $n$  and the number of generations. We will use recommendations for a classical model of an evolutionary algorithm [22]. The crossover coefficient  $\kappa$  will be selected from an interval  $[0.6; 0.8]$  and the mutation parameter  $\mu$  from an interval  $[0; 0.1]$ . As mentioned previously, a numerical value of the parameter of domination  $\beta = \infty$  (or  $\beta = 0$ ) corresponds to the classical model of evolutionary algorithms. We will investigate the interval  $1 \leq \beta \leq 10$  instead.

The size of a population used in our experiments is  $n=24$  (both the mating, initial and the current population). As mentioned previously, the objective is to generate a set of non-uniform time lags which is better than the optimal set of uniform time lags. Computational experiments with the Mackay–Glass time series and  $n=16$ ; 20; 24; 28 and 32 have shown that the highest number of successful trials is reached at  $n=24$ .

There are no definitive methods of establishing how many generations an evolutionary algorithm should run for. Simple problems may converge on good solutions after only 20 or 30 generations. More complex problems may need more. It is not unusual to run an evolutionary algorithm for 400 generations for more complex problems such as jobshops [27]. The most reliable method of deciding on this is trial and error, although a number of authors have suggested methods for determining how long a solution should live [22,27]. We will use 40 generations in our model, since further increase of the number of generations does not show improvement in the number of successful trials (Fig. 4).

A single execution of an evolutionary algorithm produces one set of time lags. Clearly, the outcome depends on the initial population of chromosomes (among other random factors). Therefore, we execute the evolutionary algorithm (at fixed values of parameters) for 100 times and calculate how many times the produced set of time lags yields the value of the objective function higher than 1.0990 (this number is denoted as  $k$ ). Simulation results are presented in Table 1. It can be noted that a



**Fig. 4.** The relationship between the number successful trials (from 100 trials) and the number of generations of the evolutionary algorithm. A successful trial is counted when a generated set of non-uniform time lags is better than the optimal set of uniform time lags. Computational experiments performed with the Mackey–Glass time series;  $\mu=0.002$ ;  $\kappa=0.8$ ;  $\beta=2$ ,  $n=24$ .



simple one-point crossover (at  $\beta=0$ ) does not produce satisfactory results.

The evolutionary algorithm has produced a set of time lags better than {9,9,9,9} the highest number of times (five times) from 100 trials at  $\mu=0.001$ ;  $\kappa=0.8$  and  $\beta=3$ . But the sum of successful trials (the sum of  $k$  in a column at different combinations of  $\mu$  and  $\kappa$ ) is highest at  $\beta=2$  (Table 1).

**Table 1**

The relationship between the number of successful trials and the parameter of similarity  $\beta$ .

The parameter of similarity $\beta$	0	1	2	3	5	10	15	20	30
The sum of successful trials	0	3	27	12	6	4	0	0	0

**Table 2**

Simulation results produced by evolutionary algorithms.  $F_{\max}$  denotes the best result (the highest value of the fitness function) produced after 100 trials at fixed values of parameters;  $k$  denotes the number of times when a generated set of time lags was better than the optimal set of uniform time lags (the value of the fitness function for the optimal set of uniform time lags is 1.0990).

$\mu$	$\kappa$	$\beta=0$		$\beta=1$		$\beta=2$		$\beta=3$		$\beta=5$		$\beta=10$	
		$F_{\max}$	$k$	$F_{\max}$	$k$	$F_{\max}$	$k$	$F_{\max}$	$k$	$F_{\max}$	$k$	$F_{\max}$	$k$
0.001	0.6	1.0853	0	1.0959	0	1.0993	2	1.0983	0	1.0963	0	1.0992	1
0.002	0.6	1.0866	0	1.101	1	1.1044	3	1.1014	1	1.0963	0	1.0956	0
0.005	0.6	1.0875	0	1.1007	1	1.1044	2	1.1004	1	1.0997	1	1.0997	1
0.01	0.6	1.0898	0	1.0957	0	1.1023	3	1.0988	0	1.0955	0	1.0914	0
0.001	0.7	1.0941	0	1.0974	0	1.101	2	1.0986	0	1.0988	0	1.0973	0
0.002	0.7	1.0965	0	1.0999	1	1.0969	0	1.0993	2	1.1014	2	1.097	0
0.005	0.7	1.0885	0	1.0981	0	1.1005	1	1.1004	1	1.099	1	1.0936	0
0.01	0.7	1.0963	0	1.0926	0	1.1026	2	1.1018	1	1.101	1	1.097	0
0.001	0.8	1.0936	0	1.0964	0	1.1017	3	1.103	5	1.0968	0	1.1014	2
0.002	0.8	1.0956	0	1.0962	0	1.1049	4	1.097	0	1.1002	1	1.0956	0
0.005	0.8	1.0842	0	1.0988	0	1.1029	3	1.0992	1	1.0963	0	1.0925	0
0.01	0.8	1.0820	0	1.0981	0	1.1012	2	1.0968	0	1.093	0	1.0973	0

Moreover, the best set of time lags {7,4,5,5,6} is generated at  $\mu=0.002$ ;  $\kappa=0.8$  and  $\beta=2$ ; the value of the objective function then becomes  $F(7,4,5,5,6)=1.1049$  (Table 2).

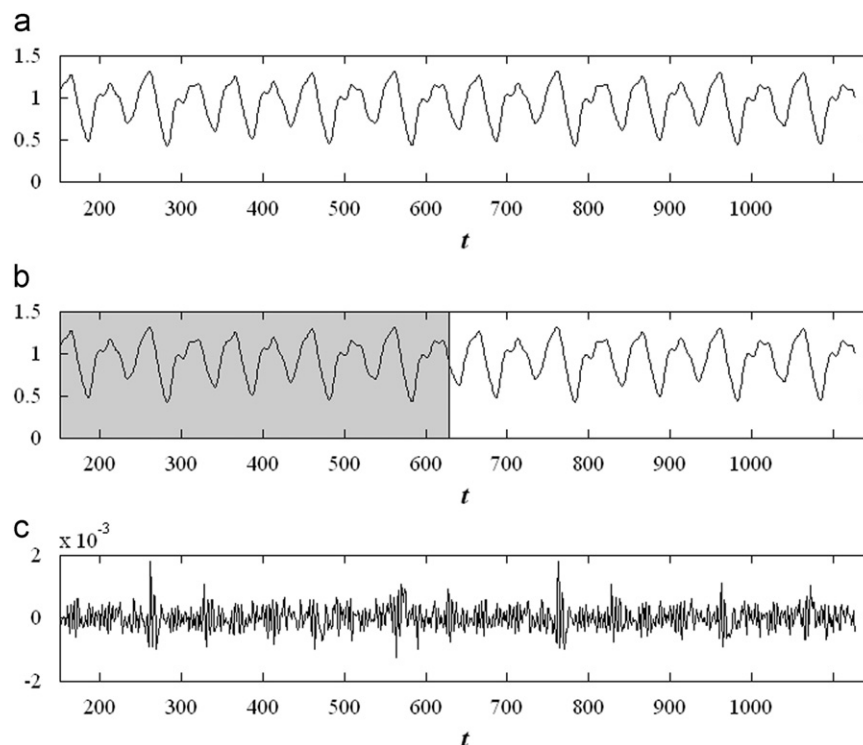
We fix these values of parameters of the evolutionary algorithm ( $\mu=0.002$ ;  $\kappa=0.8$ ;  $\beta=2$ ) for application to real world time series, but first we check prediction errors of the fuzzy inference system when the near optimal set of time lags {7,4,5,5,6} is used for the Mackey–Glass time series. We predict  $x(t+6)$  from past values  $x(t-21)$ ,  $x(t-14)$ ,  $x(t-10)$ ,  $x(t-5)$ , and  $x(t)$ . We construct 1000 data vectors from  $t=122$  to 1121, use the first 500 ones for training while the others – for determining the accuracy of the prediction. Relative prediction errors are presented in Fig. 5; the standard deviation of these errors is 3.4989E-004. It can be noted that the standard deviation of errors in case of uniform embedding was 7.2862E-004 [13].

## 5. Experiments

As mentioned in [13] the proposed method for forecasting a time series based on non-uniform embedding of a nonlinear attractor in a multidimensional phase space works well with a stationary chaotic sequence. We will test this technique for a number of different time series and then make some generalizations.

### 5.1. Mackey–Glass time series

Mackey–Glass time series is exploited to tune the parameters of the evolutionary algorithm. Nonetheless, we compare our results with prediction results reported in [28] where 50%/50% training/test split is used also. We use RMSE (root mean square error) error metrics for comparisons. RMSE errors are listed in Table 3. It can be seen that the best result is achieved by ANFIS with non-uniform embedding; our RMSE is 7 times lower



**Fig. 5.** Prediction of the Mackey–Glass time series: (a) original signal; (b) ANFIS prediction with non-uniform embedding; the gray shaded zone stands for training of the fuzzy inference system; data in the white background denote prediction results; and (c) prediction with non-uniform embedding.

**Table 3**Comparison of prediction errors using *Mackey–Glass time series* for different forecasting methods.

Prediction Method	Autoregressive model	Back-Propagation NN	Classical RBF(with 23 neurons)	Hybrid ANN's and ARMA methodology	Fuzzy inference system with uniform embedding	Fuzzy inference system with non-uniform embedding
RMSE	0.19	0.02	0.0114	0.0025	0.0007284	0.0003497

**Table 4**Comparison of prediction errors using *Daily brightness of a variable star* for different forecasting methods.

Prediction Method	ANN	Hybrid ANN's and ARMA methodology	Fuzzy inference system with uniform embedding	Fuzzy inference system with non-uniform embedding
RMSE	1.438	0.868	0.4259	0.3815

compared to the Hybrid ANN's and ARMA methodology proposed in [28] (and even much better compared to other methods listed in Table 3).

The quality of our prediction method can be explained by the fact that the predicted time series is a stationary chaotic process. By selecting the near-optimal set of time lags we build a model of the process and then use this model to extrapolate the past behavior into the future. Our fitness function does not incorporate any error metrics; we work only with metrics of the reconstructed attractor in the multidimensional phase space. Such a computational setup is deliberately constructed to predict long-term behavior of stationary chaotic systems. Experiments with the Mackey–Glass time series demonstrate the effectiveness of such approach.

We can describe the computational tool used in the experiments: MATLAB using an AMD Sempron™ Processor 3400+, 1.81 GHz, 512 MB RAM. Computational costs of our method are comprised from 4 parts: (i) the determination of an optimal embedding dimension; (ii) the determination of the optimal set of uniform time lags; (iii) the determination of a near-optimal set of time lags using evolutionary algorithms; and (iv) ANFIS time series prediction. Adding all these 4 computational costs together probably would not give a right insight into the computational complexity of the problem. Therefore we enlist evaluations of the computational complexity for all 4 parts separately.

The determination of the optimal embedding dimension  $d$  requires 7 seconds for a scalar time series comprising 1000 points. Computational costs for the determination of the optimal set of uniform time lags depend from  $d$  (7 seconds for  $d=6$ ). Hundred trials of evolutionary algorithms for the Mackey–Glass time series take 97 s. Time series prediction using 50%/50% training/test split takes 8 min.

Although the search for a best time series forecasting method continues, it is agreeable that no single method will outperform all others in all situations. It is clear that our method cannot predict well unexpected impulses, steps, seasonalities—if these were not present in the training set. As mentioned previously, our method works best with ergodic stationary chaotic signals. We compare the functionality of different methods for different time series in the following sections.

### 5.2. The daily brightness of a variable star

We will use the developed technique to predict a real world time series describing the daily brightness of a variable star on 600 successive midnights [29]. We need to determine the embedding dimension of the time series first. The False Nearest Neighbors (FNN) algorithm suggests that  $d=6$ . The optimal set of

time lags for uniform embedding is determined to be {4,4,4,4,4};  $F(4,4,4,4,4)=1.0719$ , RMSE of the prediction is 0.4259 (Table 4). It can be noted that ANFIS prediction with uniform embedding is already more than 2 times lower compared to the Hybrid ANN's and ARMA methodology proposed in [28].

We continue with non-uniform embedding. Initially, a near-optimal set of time lags is determined using the evolutionary algorithm with the same set of parameters used to predict the Mackey–Glass series. The identified set of time lags is {5,4,5,4,6}, RMSE of the prediction is 0.3815 (prediction results and errors are presented in Fig. 6). Computational costs for this time series are the same as for the Mackey–Glass time series except ANFIS prediction which required 1 additional minute (probably due to the more complex structure of the real world time series).

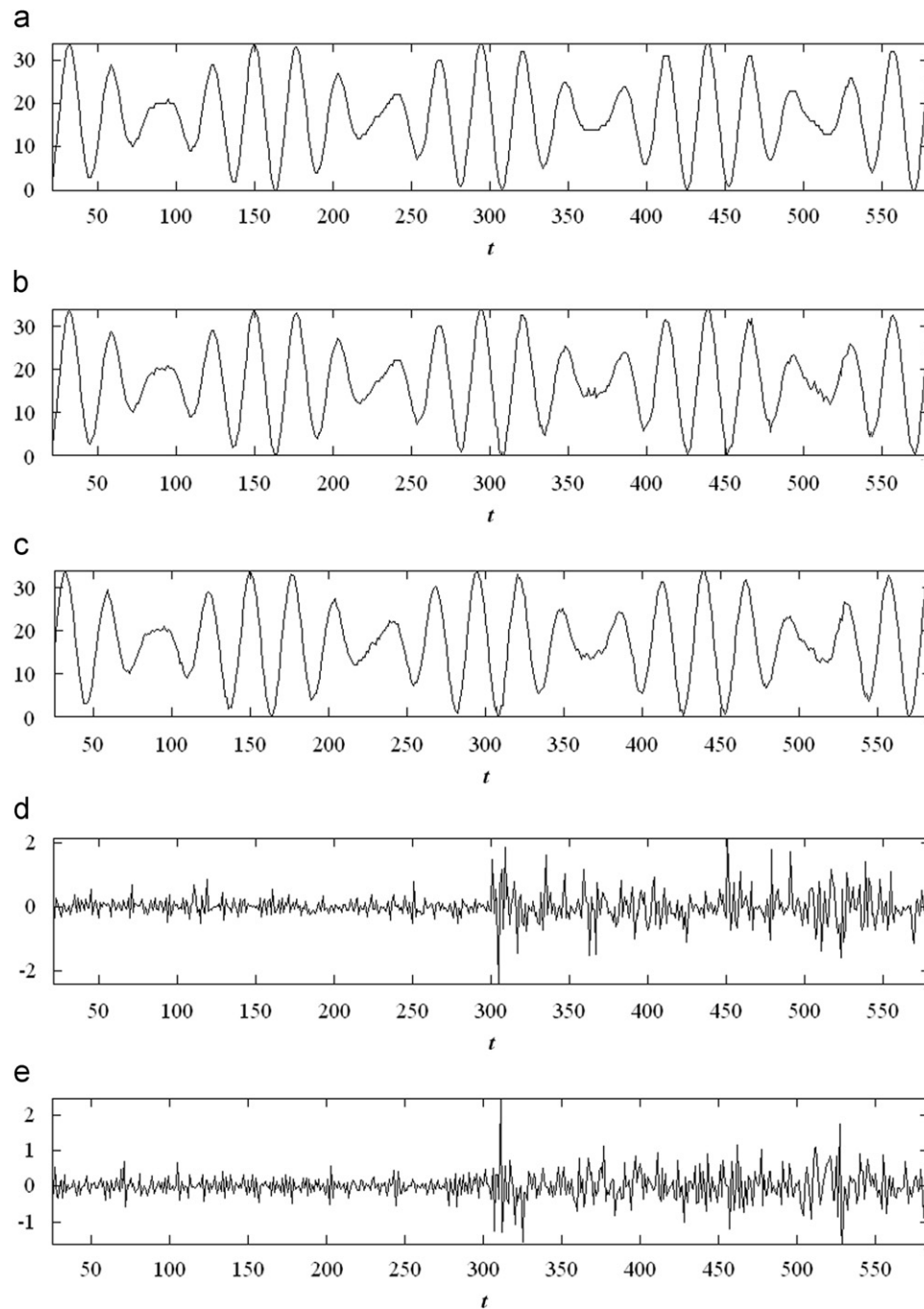
### 5.3. ECG series

We select a highly non-stationary and noisy ECG signal available in [30]. The sampling rate of the registered time series is  $1/\delta=250$  Hz. We need to determine the embedding dimension of the time series first. FNN algorithm [10] suggests that  $d=8$ . We continue with non-uniform embedding. As mentioned previously, full sorting of time lags becomes an unfeasible computational task. We use the evolutionary algorithm for the identification of a near optimal set of time lags instead.

The near-optimal set of time lags determined by the evolutionary algorithm is {14,12,15,13,12,14,13}; the value of the objective functions is  $F(14,12,15,13,12,14,13) = 1.0530$ . We predict  $x(t+13\delta)$  from the past values  $x(t-80\delta)$ ,  $x(t-66\delta)$ ,  $x(t-54\delta)$ ,  $x(t-39\delta)$ ,  $x(t-26\delta)$ ,  $x(t-14\delta)$ , and  $x(t)$ . We construct 1000 data vectors  $y_p^{(d)}$  from  $t=275+180\delta$  to  $275+1179\delta$ ; use the first 500 ones for training while the others – for determining the accuracy of the prediction. The original time series, the predicted time series and relative prediction errors are presented in Fig. 7; RMSE is 0.1131. Though prediction errors are in general quite high, ANFIS technique manages to predict main peaks of the ECG signal (though future predictions are based on already inaccurate previously predicted data). As mentioned previously, our approach is best applicable for stationary chaotic signals.

The determination of the optimal embedding dimension  $d$  requires 7 s. Nine seconds are required for the determination of the optimal set of uniform time lags. Hundred trials of evolutionary algorithms take 177 s. Time series prediction using 50%/50% training/test split takes 14 min.

In order to make comparison with other prediction methods we use the same time series and carry out prediction with Box and Jenkins procedure ARIMA(4,1,3) as the experiments found the 4-1-3 architecture as the best model (the order of the auto-



**Fig. 6.** Prediction of the daily brightness of a variable star time series: (a) original signal; (b) ANFIS prediction with uniform embedding; (c) ANFIS prediction with non-uniform embedding; (d) ANFIS prediction errors with uniform embedding; and (e) prediction with non-uniform embedding.

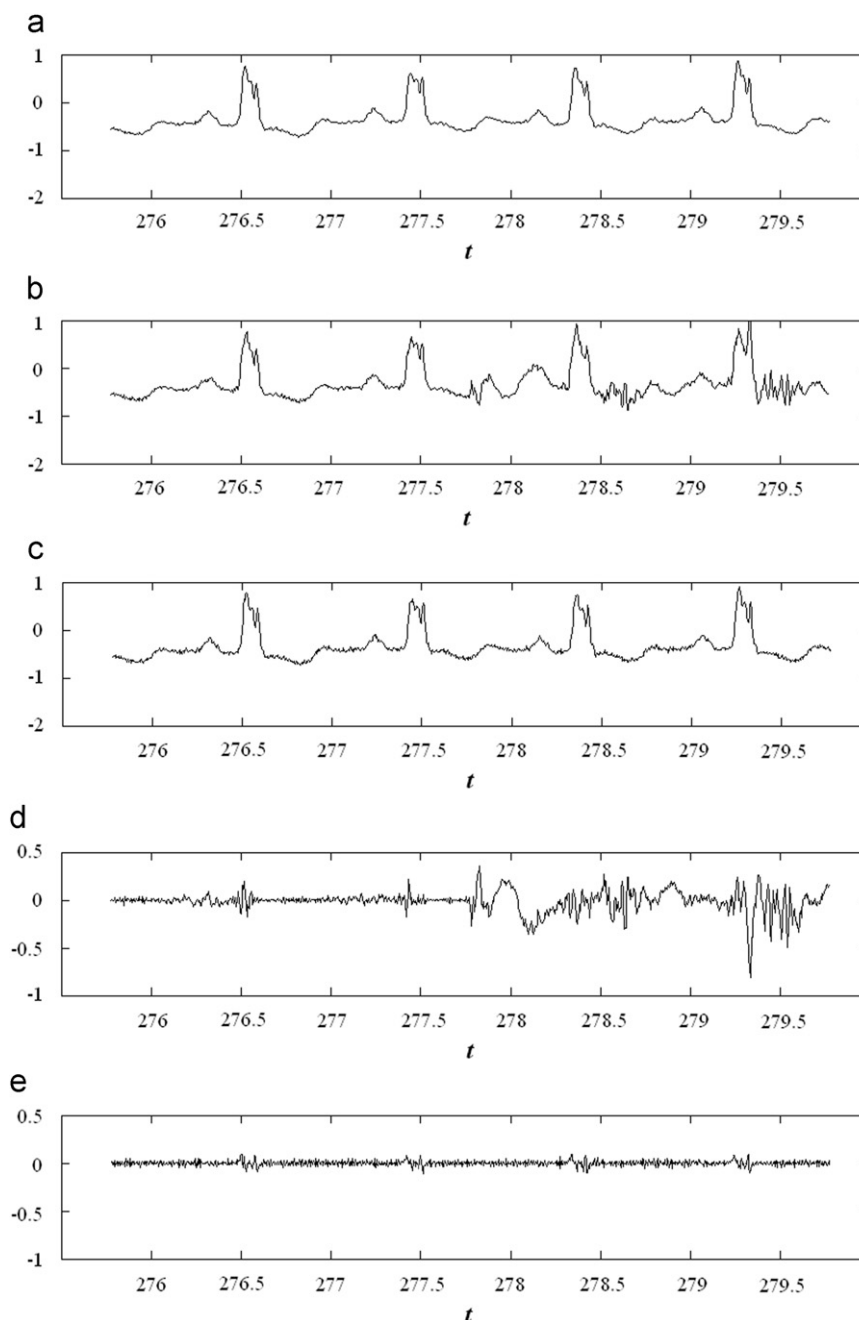
regressive part is 4; the order of the integrated part is 1 and the order of the moving average part is 3). RMSE of the prediction is presented in Table 5 and is about 5 times lower compared to our method; prediction errors are shown in Fig. 7.

Straightforward comparison between computational costs of our method and ARIMA prediction is hardly possible. The computational costs of ARIMA prediction should be split into 2 parts: (i) the identification of the optimal architecture (the autoregressive part, the integrated part and the moving average part); (ii) time series prediction. Again, adding computational costs of these two separate parts would not discriminate the efficiency of ARIMA prediction; actually the effect would be exactly opposite. In general, ARIMA prediction is much faster compared to our method. But the identification of the optimal

ARIMA architecture may take much longer compared to the determination of a near-optimal set of time lags for ANFIS prediction.

#### 5.4. Monthly returns of value-weighted S and P (standard and poor) 500 stock

We select a time series of monthly returns of value-weighted S and P (standard and poor) 500 stock data from 1926 to 1991 [29]. FNN algorithm [10] suggests that the optimal embedding dimension for this time series is  $d=8$ . The optimal set of uniform time lags is {5,5,5,5,5,5}. The near-optimal set of time lags determined by the evolutionary algorithm is {9,16,10,11,9,11,10};



**Fig. 7.** Prediction of the ECG signal: (a) original time series; (b) ANFIS prediction with non-uniform embedding; (c) ARIMA prediction; (d) ANFIS prediction errors with non-uniform embedding; and (e) ARIMA prediction errors.

**Table 5**

Comparison of prediction errors using *Medical ECG time series* for different forecasting methods.

Prediction Method	ARIMA(4,1,3)	Fuzzy inference system with non-uniform embedding
RMSE	0.0257	0.1131

the value of the objective functions is  $F\{9,16,10,11,9,11,10\}=1.0198$ . We predict  $x(t+10\delta)$ , where  $\delta=1$ , from the past values  $x(t-66\delta)$ ,  $x(t-57\delta)$ ,  $x(t-41\delta)$ ,  $x(t-31\delta)$ ,  $x(t-20\delta)$ ,  $x(t-11\delta)$ , and  $x(t)$ . We construct 1000 data vectors  $y_p^{(d)}$  from  $t=267$  to 716; use the first 225 ones for training while the others – for determining the accuracy of the prediction. The original time series, the predicted time series and relative

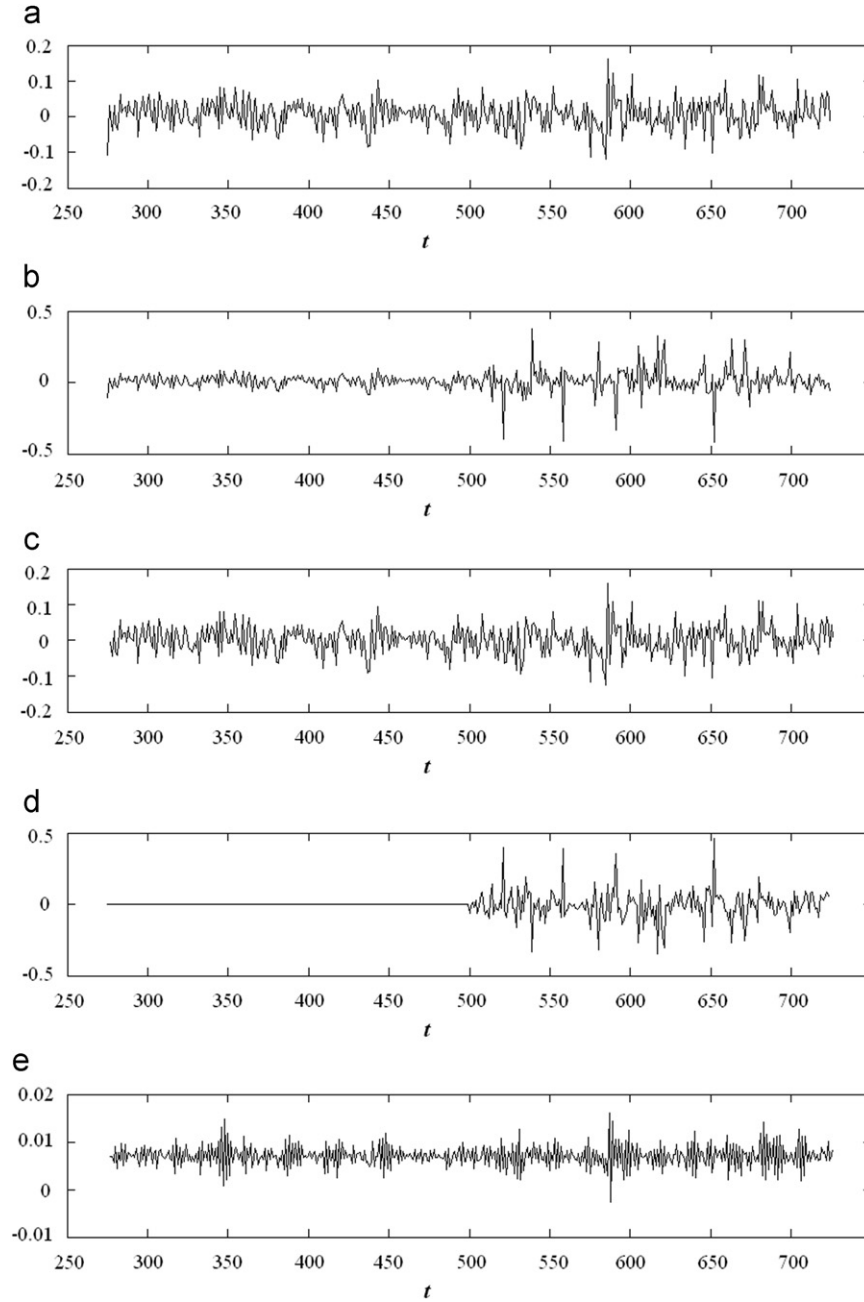
prediction errors are presented in Fig. 8; RMSE is 0.0660. Computational costs for this time series are the same as for ECG time series except ANFIS prediction which required 1 min less.

In order to make comparison with other prediction methods we use the same time series and carry out prediction with Box and Jenkins procedure ARIMA(1,0,1) as the experiments found the 1-0-1 architecture as the best model. RMSE of the prediction is presented in Table 6 and is about 1.5 times lower compared to our method; prediction errors are shown in Fig. 8.

### 5.5. Rossler time series

The Rossler chaotic map is one of the paradigm problems thoroughly explored in nonlinear dynamics [31–33]. The Rossler





**Fig. 8.** Prediction of the S and P 500 signal: (a) original time series; (b) ANFIS prediction with non-uniform embedding; (c) ARIMA prediction; (d) ANFIS prediction errors with non-uniform embedding; and (e) ARIMA prediction errors.

system comprises three ordinary differential equations

$$\begin{cases} \frac{dx}{dt} = -y - z \\ \frac{dy}{dt} = x + ay \\ \frac{dz}{dt} = b + z(x - c) \end{cases} \quad (8)$$

We select a post-transient Rossler attractor at  $a=b=0.1$  and  $c=20$ . We use time marching techniques with constant time step 0.02 to construct a numerical solution of Eq. (8). Next, we select every twelfth data point from the numerical solution and construct a new data series which will be used for forecasting (what results into time step  $\delta=0.24$ ). FNN algorithm [10] suggests that the optimal embedding dimension for this time series is  $d=6$ . The near-optimal set of time lags determined by the evolutionary

algorithm is  $\{4,4,3,4,4\}$ ; the value of the objective functions is  $F\{4,4,3,4,4\}=1.0855$ . We predict  $x(t+\delta)$ , from the past values  $x(t-15\delta)$ ,  $x(t-11\delta)$ ,  $x(t-7\delta)$ ,  $x(t-4\delta)$  and  $x(t)$ . We construct 1000 data vectors  $y_p^{(d)}$  from  $t=252.80$  to  $252.80+1000\delta$ ; use the first 500 ones for training while the others – for determining the accuracy of the prediction. The original time series, the predicted time series and relative prediction errors are presented in Fig. 9; RMSE is 0.4839. Computational costs for this time series are the same as for the Mackey–Glass time series.

In order to make comparison with other prediction methods we use the same time series and carry out prediction with Box and Jenkins procedure ARIMA(2,0,3) as the experiments found the 2-0-3 architecture as the best model. RMSE of the ARIMA prediction is presented in Table 7 and is more than 3 times higher compared to our method; prediction errors are shown in Fig. 9.

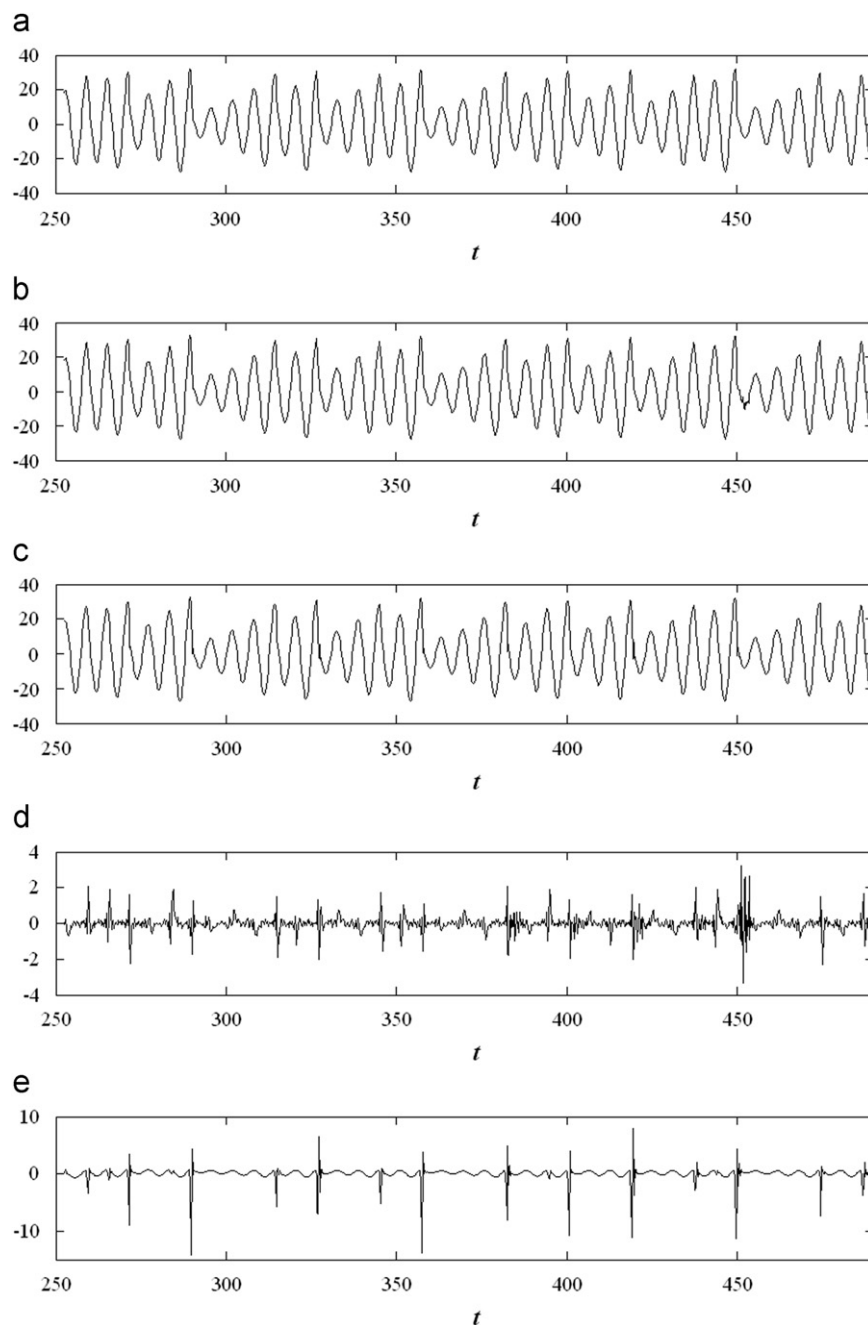
Our prediction method works very well with stationary chaotic time series even strongly contaminated with noise. Our method tries to build the nonlinear model of the system. In general, the dynamics of the time series in our fitness function (and in the procedure of evolutionary optimization) is represented through its Fourier amplitude spectrum (Eq. 4). In fact there is almost no difference (except for the frequency window) if the Fourier amplitude spectrum is calculated using only the training set or

both the training and the test sets, if only the process is stationary. Though ANFIS prediction is a separate procedure executed after the near-optimal set of time lags is identified, excellent results with the Mackey–Glass time series and the Rossler time series can be explained first of all due to the fact that both time series are stationary. In other words, the optimal set of time lags is the same for the training set and the validation set (what is not the case for the ECG series for example).

**Table 6**

Comparison of prediction errors using *S* and *P* 500 stock time series for different forecasting methods.

Prediction Method	ARIMA(1,0,1)	Fuzzy inference system with uniform embedding	Fuzzy inference system with non-uniform embedding
RMSE	0.0387	0.0635	0.066



**Fig. 9.** Prediction of the Rössler time series: (a) original time series; (b) ANFIS prediction with non-uniform embedding; (c) ARIMA prediction; (d) ANFIS prediction errors with non-uniform embedding; and (e) ARIMA prediction errors.

**Table 7**Comparison of prediction errors using *Rossler time series* for different forecasting methods.

Prediction Method	ARIMA(2,0,3)	Fuzzy inference system with uniform embedding	Fuzzy inference system with nonuniform embedding
RMSE	1.2723	0.4926	0.4839

The parameters of evolutionary algorithms for the identification of a near-optimal set of time lags are tuned for the Mackey–Glass time series. But this is because we know the exact optimal solution (full sorting has been performed in advance). But that does not mean that the developed strategy works well only with the Mackey–Glass time series. Evolutionary algorithms do not find the global optimum for the Mackey–Glass series. Similarly, evolutionary algorithms do not find the global maximum for the Rossler time series, though ANFIS prediction is nevertheless better compared to ARIMA. If the computational resources would be unlimited, one could exclude evolutionary algorithms and make full sorting of all sets of time lags. Then ANFIS prediction would be even better.

Naturally, our method is not the best choice if the model of the predicted system changes strongly in time. As mentioned previously, one method cannot outperform all other methods in all situations. Experiments with the sunspot series have shown that our method cannot outperform a novel intelligent system methodology based on artificial neural networks presented in [17]. But coupling of our method with the functionality of other prediction methods (when the fitness function incorporates both error and attractor metrics for example) could be an interesting object of future research.

## 6. Concluding remarks

We have extended time series prediction technique based on non-uniform embedding [13] by adding the functionality of evolutionary algorithms into the procedure of determination of a near-optimal set of time lags. The proposed set-up of evolutionary algorithms copes well with the discrete multi-dimensional optimization problem and allows satisfactory prediction of complex nonlinear signals.

The developed prediction method is especially effective when the predicted time series is a stationary chaotic process. The proposed fitness function does not incorporate any error metrics; metrics of the reconstructed attractor in the multidimensional delay phase space is used instead. Such a computational setup is deliberately constructed to predict long-term behavior of stationary chaotic systems.

The Mackey–Glass, the daily brightness of a variable star and the Rossler time series are used to illustrate the effectiveness of the proposed method. Although the search for a best time series forecasting method continues, it is agreeable that no single method will outperform all others in all situations. It is clear that the proposed method cannot predict well unexpected impulses, steps, seasonalities – if these were not present in the training set. Other different approaches could be advantageous then. Computational experiments with Box and Jenkins procedure ARIMA illustrate that it produces better predictions for such time series as ECG or Standard and Poor series compared to the proposed method. RMSE error metrics are used in this paper to assess predictions of different time series. Since the proposed fitness function does not include error metrics, the developed prediction method may work well if other forecasting metrics are used (for example Mean Absolute Percent Error).

Comparisons between different methods could be enriched by discussions about costs and benefits of these methods. Probably a

much more important question would be the availability of a pre-test which could be used to check if the prediction of a time series by our method would be feasible (or some other methods should be selected instead). As mentioned previously, our method works well with stationary signals, even if they are strongly nonlinear, chaotic or contaminated with noise. Since our fitness function does not include error metrics and uses the metrics of the reconstructed attractor instead, the pre-test could be based on Fourier amplitude spectrum. Several different time windows should be selected in the same time series and Fourier amplitude spectrums should be calculated for each of those time windows. If the differences between these Fourier amplitude spectrums are large, one should not expect very good results with our prediction method. The reason is simple – optimal sets of time lags would be very much different at different parts of the time series. How large differences between Fourier amplitude spectrums can be still tolerated by our method, and when one should be recommended to select for example ARIMA method instead, is a very interesting question and is a definite object of future research.

We have shown that evolutionary algorithms can be effectively used to find a near-optimal set of time lags. In principle, a near-optimal set of time lags could be determined using Simulated Annealing, Tabu Search, Differential Evolution or Particle Swarm Optimization algorithms. However, we have shown that classical evolutionary algorithms (at  $\beta=0$ ) do not produce satisfactory results, and special modifications are necessary in order to achieve a better set of non-uniform time lags than the best set of uniform time lags. It is probable that some modifications would be necessary also in other optimization algorithms. These questions remain out of scope of interests for this paper, but also are definite objectives for future research.

## Acknowledgements

We would like to thank all anonymous reviewers for their comments and suggestions which helped to improve the paper.

## References

- [1] H. Gu, H. Wang, Fuzzy prediction of chaotic time series based on singular value decomposition, *Applied Mathematics and Computation* 185 (2007) 1171–1185.
- [2] T. Hui-Kuang Yu, Kun-Huang Huarng, A bivariate fuzzy time series model to forecast the TAIEX, *Expert Systems with Applications* 34 (2008) 2945–2952.
- [3] M. Khashei, S. Reza, M. Bijari, A new hybrid artificial neural networks and fuzzy regression model for time series forecasting, *Fuzzy Sets and Systems* 159 (2008) 769–786.
- [4] M. Zounemat-Kermani, M. Teshnehlab, Using adaptive neuro-fuzzy inference system for hydrological time series prediction, *Applied Soft Computing* 8 (2008) 928–936.
- [5] D.J. Pedregal, R. Rivas V. Feliua, L. Sánchez, A. Linares, A simple model to predict soil moisture: bridging event and continuous hydrological (BEACH) modeling, *Environmental Modelling & Software* 24 (2009) 502–509.
- [6] M.K. Das, N. Kishor, Adaptive fuzzy model identification to predict the heat transfer coefficient in pool boiling of distilled water, *Expert Systems with Applications* 36 (2) (2009) 1142–1154.
- [7] M. Khashei, M. Bijari, G.A.R. Ardali, Improvement of auto-regressive integrated moving average models using fuzzy logic and artificial neural networks (ANNs), *Neurocomputing* 72 (4–6) (2009) 956–967.
- [8] L.J. Herrera, H. Pomares, I. Rojas, A. Guillén, J. González, M. Awad, A. Herrera, Multigrid-based fuzzy systems for time series prediction: CATS competition, *Neurocomputing* 70 (13–15) (2007) 2410–2425.

- [9] T. Schreiber, Interdisciplinary application of nonlinear time series methods, *Physics Reports—Review Section of Physics Letters* 308 (1999) 2–64.
- [10] T. Sauer, M. Casdagli, J.A. Yorke, Embedology, *Journal Statistical Physics* 65 (1991) 579–616.
- [11] T. Buzug, G. Pfister, Optimal time delay and embedding dimension for delay time coordinates by the analysis of global statical and local dynamical behaviour of strange attractors, *Physical Review A* 45 (1992) 7073–7084.
- [12] Y. Manabe, B. Chakraborty, A novel approach for estimation of optimal embedding parameters of nonlinear time series by structural learning of neural network, *Neurocomputing* 70 (2007) 1360–1371.
- [13] M. Ragulskis, K. Lukoseviciute, Non-uniform attractor embedding for time series forecasting by fuzzy inference systems, *Neurocomputing* 72 (10–12) (2009) 2618–2626.
- [14] D.Y. Chiu, P.J. Chen, Dynamically exploring internal mechanism of stock market by fuzzy-based support vector machines with high dimension input space and genetic algorithm, *Expert Systems with Applications* 36 (2, Part 1) (2009) 1240–1248.
- [15] W.H. Ho, J.T. Tsai, B.T. Lin, J.H. Chou, Adaptive network-based fuzzy inference system for prediction of surface roughness in end milling process using hybrid Taguchi-genetic learning algorithm, *Expert Systems with Applications* 36 (2, Part 2) (2009) 3216–3222.
- [16] Z.L. Sun, D.S. Huang, C.H. Zheng, L. Shang, Optimal selection of time lags for TDSEP based on genetic algorithm, *Neurocomputing* 69 (2006) 884–887.
- [17] T.A.E. Ferreira, G.C. Vasconcelos, P.J.L. Adeodato, A new intelligent system methodology for time series forecasting with artificial neural networks, *Neural Processing Letters* 28 (2008) 113–129.
- [18] R. de A. Araújo, T.A.E. Ferreira, An intelligent hybrid morphological-rank-linear method for financial time series prediction, *Neurocomputing* 72 (10–12) (2009) 2507–2524.
- [19] J.S.R. Jang, Adaptive-network-based fuzzy inference system, *IEEE Transactions on Systems, Man and Cybernetics* 23 (1993) 665–685.
- [20] M. Camara, J. Ortega, F. de Toro, A single from genetic algorithm for parallel multi-objective optimization in dynamic environments, *Neurocomputing* 72 (2009) 3570–3579.
- [21] M. Gong, L. Jiao, X. Zhang, A population-based artificial immune system for numerical optimization, *Neurocomputing* 72 (2008) 149–161.
- [22] J.H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Application to Biology, Control, and Artificial Intelligence*, The MIT Press, 1992.
- [23] D. Whitley, A genetic algorithm tutorial, *Statistics and Computing* 4 (2) (1994) 65–85.
- [24] H. Gu, H. Wang, Fuzzy prediction of chaotic time series based on singular value decomposition, *Applied Mathematics and Computation* 185 (2007) 1171–1185.
- [25] C. Harpharm, C.W. Dawson, The effect of different basis functions on a radial basis function network for time series prediction: a comparative study, *Neurocomputing* 69 (2006) 2161–2170.
- [26] L. Zunxiong, L. Jianhui, Chaotic time series multi-step direct prediction with partial least squares regression, *Journal of Systems Engineering and Electronics* 18 (2007) 611–615.
- [27] <<http://www.edc.ncl.ac.uk/highlight/rhjanuary2007g01.php>>.
- [28] I. Rojas, O. Valenzuela, F. Rojas, A. Guillen, L.J. Herrera, H. Pomares, L. Marquez, M. Pasadas, Soft-computing techniques and ARMA model for time series prediction, *Neurocomputing* 71 (2008) 519–537.
- [29] <<http://robjhyndman.com/TSDL/>>.
- [30] E. Keogh, J. Lin, S.H. Lee, H.V. Herle, Finding the most unusual time series subsequence: algorithms and applications, *Knowledge and Information Systems* 11 (1) (2006) 1–27.
- [31] O.E. Röessler, An equation for continuous chaos, *Physics Letters* 57A (1976) 397–398.
- [32] S.H. Strogatz, *Nonlinear Dynamics and Chaos*, Perseus Publishing, Massachusetts, 1994.
- [33] T. Gedeon, M. Holzer, M. Parnarowski, Attractor reconstruction from interspike intervals is incomplete, *Physica D: nonlinear phenomena* 178 (2003) 149–172.



**Lukoseviciute Kristina** received the M.Sc. degree in mathematics in 2006 from the University of Technology, Lithuania. She is currently an associate lecturer and a Ph.D. degree student within the Department of Mathematical Research in Systems, Kaunas University of Technology. Her current areas of research interest are neuro-fuzzy systems and time series forecasting.



**Ragulskis Minvydas** received the Ph.D. in 1992 from Kaunas University of Technology, Lithuania. Since 2002 he is a professor at the Department of Mathematical Research in Systems and the head of Research Group for Mathematical and Numerical Analysis of Dynamical Systems, Kaunas University of Technology ([www.personalas.ktu.lt/~mragul](http://www.personalas.ktu.lt/~mragul)). His research interests include nonlinear dynamical systems and numerical analysis.