Population-Based Algorithm with Selectable Evolutionary Operators for Nonlinear Modeling

Krystian Łapa(✉)

Institute of Computational Intelligence,
Częstochowa University of Technology, Częstochowa, Poland
krystian.lapa@iisi.pcz.pl

Abstract. In this paper a new population-based algorithm for nonlinear modeling is proposed. Its advantage is the automatic selection of evolutionary operators and their parameters for individuals in population. In this approach evolutionary operators are selected from a large set of operators, however only the solutions that use low number of operators are promoted in population. Moreover, assigned operators can be changed during evolution of population. Such approach: (a) eliminates the need for determining detailed mechanism of the population-based algorithm, and (b) reduces the complexity of the algorithm. For the simulations typical nonlinear modeling benchmarks are used.

Keywords: Population-based algorithms · Fuzzy systems · Nonlinear modeling · Selection of evolutionary operators

1 Introduction

Nonlinear modeling issues can be solved using, among the others, population-based algorithms which are part of computational intelligence methods (see e.g. [4]). The idea of population-based algorithms relies on iterative processing of individuals. In each iteration a new generation of individuals is generated (characterized, by definition, by improved values of the objective function). Population-based algorithms differ from traditional optimization methods: (a) they do not process parameters directly, but their encoded form, (b) they search solutions based not on a single point, but on a population of the points (the population contains individuals, each individual encodes single solution), (c) they use objective function directly, not its derivatives (this function determines the quality of the solutions in the population), (d) they use probabilistic mechanisms, not deterministic. As a result, they have advantage over other optimization techniques such as: analytical methods, random methods, etc. [6].

Population-based algorithms can be also combined with other computational intelligence methods, such as: artificial neural networks, fuzzy systems, decision trees, etc. Then, they can be used for optimization of parameters and structures of mentioned methods. It is worth to mention that, in the optimization process an extensive objective function (called also fitness function or evaluation function) can be used. Thanks to that, the quality criteria (e.g. related with accuracy and interpretability) as well as

L. Borzemski et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017*,
Advances in Intelligent Systems and Computing 655, DOI 10.1007/978-3-319-67220-5_2

quantitative criteria (e.g. related with complexity) can be included in evaluation of the individuals. Such an approach was contemplated in previous research [17].

Currently, many varieties of population-based algorithms exist and they can be divided to: (a) single-population algorithms (see e.g. [22]) and multi-population algorithms (see e.g. [17]), (b) algorithms generating single solutions (see e.g. [1]) and fronts of solutions (see e.g. [8]), (c) algorithms using single-objective function (see e.g. [24]), and multi-objective function (see e.g. [5, 11]). The efficiency of those algorithms strongly depends on evolutionary operators used for exploration and exploitation of the search space (search space determines acceptable boundaries for parameters of solutions). Evolutionary operators usually have a set of parameters that must be selected before starting the evolution process (this is done usually by trial and error method) or modified in this process (usually by specified).

In this paper we have attempted to construct an algorithm that automatically selects evolutionary operators (for exploration and exploitation of search space) and their parameters. The idea behind this solution is that each individual in the population has (beside typical list of parameters) a list of available (typical) evolutionary operators and list of their parameters. At the beginning of the algorithm's operation, this list is initialized randomly, however while evolution of population it is updated simultaneously with typical parameters. In such approach individuals are modified basing on the values of list of available evolutionary operators and their parameters. In our previous work [16] this solution was partially considered. The problem was the large number of evolutionary operators used by the population. Due to that, the following improvements were made: (a) the list of available operators was improved, and (b) objective function was changed to reduce the number of used evolutionary operators. As a result, the individuals of the population that use fewer operators are better rated. Proposed approach has been used for nonlinear modeling using fuzzy systems.

The structure of this paper is as follows: in Sect. 2 a description of proposed method is placed, Sect. 3 contains simulation results and in Sect. 4 the conclusions are drawn.

2 Description of Proposed Method

Characteristics of the proposed method can be summarized as follows:

- The method is characterized by the ability of an automatic creation of model during the process of evolution. The most important feature of the method is the fact that during the process of evolution the selection and configuration of the evolution operators used for exploration and exploitation takes place simultaneously. This eliminates the need for selection of the type of operators and their parameters by trial and error method. This approach also allows to achieve an appropriate balance between exploration and exploitation of the search space.
- The method is based on the capabilities of fuzzy systems, which are a convenient tool for nonlinear modeling. Modeling can be done in different ways: (a) fuzzy system can model input-output dependences (see e.g. [9]), (b) fuzzy system can model elements of state variables matrix (see e.g. [3]), (c) fuzzy system can provide

appropriate cooperation of partial models representing various operating states of the modeled object (see e.g. [10]). In this paper the first type of modeling will be considered, and the description of system used in the modeling is given in Sect. 2.1.

– The method uses hybrid approach introduced by us earlier, enabling the simultaneous selection of real parameters and binary parameters (see e.g. [17]). Due to this, the structure and parameters used for modeling of fuzzy system and used in the process of evolution operators can be automatically selected.
– The method using fitness function component that promotes individuals that use lower number of evolutionary operators. Such solution not only affect the computational time but also causes the algorithm to work consistently.

The proposed method is based on the approach analogical to particle swarm optimization (PSO, [13]). PSO algorithm uses population of individuals, in which each individual encodes parameters of potential solution to the problem under consideration (marked as $\mathbf{X}_{ch}^{\text{par}} = \{X_{ch,1}^{\text{par}}, \ldots, X_{ch,L^{\text{par}}}^{\text{par}}\}$), velocity vector, used for modification of potential solution parameters ($\mathbf{X}_{ch}^{\text{vel}} = \{X_{ch,1}^{\text{vel}}, \ldots, X_{ch,L^{\text{par}}}^{\text{vel}}\}$), and best found so far parameters of potential solution $\mathbf{p}$ (marked as $\mathbf{X}_{ch}^{\text{bst}} = \{X_{ch,1}^{\text{bst}}, \ldots, X_{ch,L^{\text{par}}}^{\text{bst}}\}$). In the algorithm the best solution found in population is additionally remembered (marked as $\mathbf{X}^{\text{glb}} = \{X_1^{\text{glb}}, \ldots, X_{L^{\text{par}}}^{\text{glb}}\}$). In the evolution process, the parameters $\mathbf{X}_{ch}^{\text{vel}}$ and $\mathbf{X}_{ch}^{\text{par}}$ are subject to modification according to the following equation:

$$\begin{cases} X_{ch,g}^{\text{vel}} := w \cdot X_{ch,g}^{\text{vel}} + c_1 \cdot \mathrm{U}(0,1) \cdot \left(X_{ch,g}^{\text{bst}} - X_{ch,g}^{\text{par}}\right) + c_2 \cdot \mathrm{U}(0,1) \cdot \left(X_g^{\text{glb}} - X_{ch,g}^{\text{par}}\right) \\ X_{ch,g}^{\text{par}} := X_{ch,g}^{\text{par}} + X_{ch,g}^{\text{vel}}, \end{cases} \tag{1}$$

where $ch = 1, \ldots, Npop$ is index of individual in population, $Npop$ stands for number of individuals in population, $g = 1, \ldots, L^{\text{par}}$ is index of real parameter (gene), L^{par} stands for number of real parameters (actual number of parameters is different, which is explained in Sect. 2.2), w is inertia weight (usually $w \in [0.8, 1.0]$), c_1 and c_2 are cognitive and social parameters (see [13]), and $\mathrm{U}(a, b)$ is function returning random number from range $[a, b]$.

On the basis of PSO algorithm a generalization of (1) was proposed. The generalization was designed in a way that any evolutionary operator of exploration and exploitation can be used. Moreover, operators and their parameters can be selected dynamically in evolution process. This is consistent with two facts on the population algorithms: (a) in some algorithms operators of exploration and exploitation are intermingled, (b) in modification of parameters multiple operators can be used simultaneously. The generalized form of Eq. (1) takes the following form:

$$\begin{cases} X_{ch,g}^{\text{vel}} := w \cdot X_{ch,g}^{\text{vel}} + \sum_{o=1}^{L^{\text{op}}} X_{ch,o}^{\text{op}} \cdot \mathrm{op}_o\left(X_{ch,g}^{\text{par}}, X_g^{\text{glb}}, X_{ch,g}^{\text{bst}}, X_{ch1,g}^{\text{par}}, X_{ch2,g}^{\text{par}}, X_{ch,g}^{\text{imp}}\right) \\ X_{ch,g}^{\text{par}} := X_{ch,g}^{\text{par}} + X_{ch,g}^{\text{vel}}, \end{cases}, \tag{2}$$

where vector $\mathbf{X}_{ch}^{\text{op}} = \{X_{ch,1}^{\text{op}}, \ldots, X_{ch,L^{\text{op}}}^{\text{op}}\}$ contains information on binary keys that stand for activation state of operators connected with them (an assumption was taken that 0 value stands for excluded operator and vice versa), L^{op} stands for number of considered operators (see Table 1), $\mathrm{op}_{\mathrm{o}}(\cdot)$ stand for functions representing operator with index o, $\mathbf{X}_{ch1}^{\mathrm{par}}$ and $\mathbf{X}_{ch2}^{\mathrm{par}}$ stand for parent individuals chosen by selection method (for example roulette wheel method, [20]), $\mathbf{X}_{ch}^{\mathrm{imp}}$ means additional individual used in assimilation operator (Table 1, $o = 10$, [2]). The individual $\mathbf{X}_{ch}^{\mathrm{imp}}$ stands for imperialist of current solution selected as follows:

$$X_{ch}^{\mathrm{imp}} = \begin{cases} X_{ch,g}^{\mathrm{par}} & \text{for } ch \leq Nimp \\ X_{Nimp\cdot((ch-Nimp)/(Npop-Nimp)),g}^{\mathrm{par}} & \text{for } ch > Nimp \end{cases}, \tag{3}$$

where *Nimp* stands for number of empires [2].

The set of considered evolutionary operators is presented in Table 1. In this table the following marks are additionally used: $\alpha = \mathrm{U}(0,1)$ stands for a random number generated individually for each parameter under modification, $\beta = \mathrm{U}(0,1)$ stands for a random number generated individually for each individual under modification, *RInd* is randomly chosen index of parameter, *RSet* contains a set of randomly chosen indexes of parameters, ff(.) stand for objective function of individual, ff_{min} is a smallest value of objective function for current population, $\mathrm{U}_G(1,1)$ is random number according to Gaussian distribution with mean 1 and variance 1, A^t is parameter additionally multiplied by coefficient α^t each time when individual modification improves solution (see e.g. [24]), $dist(\mathbf{X}_{ch1}, \mathbf{X}_{ch})$ stands for Manhattan distance between genes of two individuals.

2.1 Fuzzy System Used for Nonlinear Modeling

As previously mentioned, a multi-input, multi-output fuzzy system of the Mamdani-type that maps $\mathbf{X} \to \mathbf{Y}$ is used, where $\mathbf{X} \subset \mathbf{R}^n$ and $\mathbf{Y} \subset \mathbf{R}^m$ is used for nonlinear modeling. The fuzzy rule base of the system consists of a collection of N fuzzy if-then rules that takes the following form:

$$R^k : \left[\mathrm{IF}\left(x_1 \,\mathrm{is}\, A_1^k\right)\mathrm{AND}\ldots\mathrm{AND}\left(x_n \,\mathrm{is}\, A_n^k\right)\mathrm{THEN}\left(y_1 \,\mathrm{is}\, B_1^k\right), \ldots, \left(y_m \,\mathrm{is}\, B_m^k\right)\right], \tag{4}$$

where $\mathbf{x} = [x_1, \ldots, x_n] \in \mathbf{X}$, $\mathbf{y} = [y_1, \ldots, y_m] \in \mathbf{Y}$, $A_1^k, \ldots, A_n^k$ are fuzzy sets characterized by membership functions $\mu_{A_i^k}(x_i)$, $i = 1, \ldots, n$, $k = 1, \ldots, N$, n stands for number of inputs, $B_1^k, \ldots, B_m^k$ are fuzzy sets characterized by membership functions $\mu_{B_j^k}\left(y_j\right)$, $j = 1, \ldots, m$, $k = 1, \ldots, N$, and m stands for number of outputs. In this paper a Gaussian membership type functions are considered [20].

In the Mamdani approach (see e.g. [20]) output signal $\bar{y}_j$, $j = 1, \ldots, m$, of the fuzzy system is described by the following formula (for more details see [9]):

$$\bar{y}_j = \frac{\sum_{r=1}^{R} \bar{y}_{j,r}^{\mathrm{def}} \cdot \overset{N}{\underset{k=1}{S}} \left\{ T\left\{ \overset{n}{\underset{i=1}{T}} \left\{ \mu_{A_i^k}(\bar{x}_i) \right\}, \mu_{B_j^k}\left(\bar{y}_{j,r}^{\mathrm{def}}\right) \right\} \right\}}{\sum_{r=1}^{R} \overset{N}{\underset{k=1}{S}} \left\{ T\left\{ \overset{n}{\underset{i=1}{T}} \left\{ \mu_{A_i^k}(\bar{x}_i) \right\}, \mu_{B_j^k}\left(\bar{y}_{j,r}^{\mathrm{def}}\right) \right\} \right\}}, \tag{5}$$

where $\bar{y}^{\text{def}}_{j,r}$, $j = 1,\ldots,m$, $r = 1,\ldots,R$, are discretization points, R is a number of discretization points, $T\{\cdot\}$ is a t-norm, and $S\{\cdot\}$ is a t-conorm (see e.g. [15]).

2.2 Encoding of the Individuals

The selected encoding refers to Pittsburgh approach [12]. Thus, the single individual $\mathbf{X}_{ch}$ encodes the information on:

- Keys of operators (binary parameters). They are encoded as $\mathbf{X}^{\text{op}}_{ch}$ and they are used in Eq. (2) (see Table 1):

$$\mathbf{X}^{\text{op}}_{ch} = \left\{op_{ch,1},\ldots,op_{ch,L^{op}}\right\} = \left\{X^{\text{op}}_{ch,1},\ldots,X^{\text{op}}_{ch,L^{op}}\right\}, \tag{6}$$

  where L^{op} stands for number of evolutionary operators that can be used by individual encoded as $\mathbf{X}_{ch}$.
- Parameters of fuzzy system (5) and parameters of operators (for the list of operators parameters see Table 1):

$$\begin{aligned}\mathbf{X}^{\text{par}}_{ch} &= \begin{cases} \bar{x}^{A}_{ch,1,1}, \sigma^{A}_{ch,1,1},\ldots,\bar{x}^{A}_{ch,n,1}, \sigma^{A}_{ch,n,1},\ldots\bar{x}^{A}_{ch,1,N}, \sigma^{A}_{ch,1,N},\ldots,\bar{x}^{A}_{ch,n,N}, \sigma^{A}_{ch,n,N}, \\ \bar{y}^{B}_{ch,1,1}, \sigma^{B}_{ch,1,1},\ldots,\bar{y}^{B}_{ch,m,1}, \sigma^{B}_{ch,m,1},\ldots\bar{y}^{B}_{ch,1,N}, \sigma^{B}_{ch,1,N},\ldots,\bar{y}^{B}_{ch,m,N}, \sigma^{B}_{ch,m,N}, \\ \bar{y}^{\text{def}}_{ch,1,1},\ldots,\bar{y}^{\text{def}}_{ch,1,R},\ldots,\bar{y}^{\text{def}}_{ch,m,1},\ldots,\bar{y}^{\text{def}}_{ch,m,R}, \\ w, c_1, c_2, p_c, m_r, p_m, F^{\text{DE}}, d, \alpha^{\text{FA}}, \beta^{\text{FA}}, \gamma, CR, f_{\min}, f_{\max}, A^t, \alpha^t, \hat{A}, F^{\text{BTO}} \end{cases} \\ &= \left\{X^{\text{par}}_{ch,1},\ldots,X^{\text{par}}_{ch,L^{\text{par}}}\right\},\end{aligned} \tag{7}$$

  where $\{\bar{x}^{A}_{i,k}, \sigma^{A}_{i,k}\}$ stand for parameters of membership functions of input Gaussian fuzzy sets $A^k_1,\ldots,A^k_n$, $\{\bar{y}^{B}_{j,k}, \sigma^{B}_{j,k}\}$ stand for parameters of membership functions of output Gaussian fuzzy sets $B^k_1,\ldots,B^k_m$, and L^{par} is the number of genes of part $\mathbf{X}^{\text{par}}_{ch}$.
- Velocity vector of parameters $\mathbf{X}^{\text{par}}_{ch}$ (see (7)). They are encoded as set $\mathbf{X}^{\text{vel}}_{ch}$.
- Best location of individual $\mathbf{X}^{\text{bst}}_{ch}$ (they are copy of best parameters of $\mathbf{X}^{\text{par}}_{ch}$).

A individual is thus a collection of components: $\mathbf{X}_{ch} = \{\mathbf{X}^{\text{op}}_{ch}, \mathbf{X}^{\text{par}}_{ch}, \mathbf{X}^{\text{vel}}_{ch}, \mathbf{X}^{\text{bst}}_{ch}\} = \{X_{ch,1},\ldots,X_{ch,L}\}$ with a total length of genes equal to $L = 3 \cdot L^{\text{par}} + L^{\text{op}}$.

2.3 Evaluation of the Individuals

The aim of the objective function (fitness function) is to minimize following error:

$$\text{ff}(\mathbf{X}_{ch}) = T\{\epsilon(\mathbf{X}_{ch}), \delta(\mathbf{X}_{ch})\}. \tag{8}$$

Table 1. Chosen functions that represent evolutionary operators of exploration and exploitation considered in this paper.

o	Base method	$\text{op}_o\left(X^{\text{par}}_{ch,g}, X^{\text{glb}}_{g}, X^{\text{bst}}_{ch,g}, X^{\text{par}}_{ch1,g}, X^{\text{par}}_{ch2,g}, X^{\text{imp}}_{ch,g}\right)$	Parameters
1	PSO-best [13]	$c_1 \cdot \text{U}(0,1) \cdot \left(X^{\text{bst}}_{ch,g} - X^{\text{par}}_{ch,g}\right)$	$c_1 \in [1.5, 2.5]$
2	PSO-global [13]	$c_2 \cdot \text{U}(0,1) \cdot \left(X^{\text{glb}}_{g} - X^{\text{par}}_{ch,g}\right)$	$c_2 \in [1.5, 2.5]$
3	GA-crossover [20]	$\begin{cases} \text{U}(0,1) \cdot \left(X^{\text{par}}_{ch1,g} - X^{\text{par}}_{ch,g}\right) & \text{for} \quad \beta < p_c \\ 0 & \text{for} \quad \text{otherwise} \end{cases}$	$p_c \in [0.7, 1.0]$
4	GA-mutation [20]	$\begin{cases} \text{U}(-1,1) \cdot m_r & \text{for} \quad \alpha < p_m \\ 0 & \text{for} \quad \text{otherwise} \end{cases}$	$m_r \in [0.01, 0.20]$, $p_m \in [0.05, 0.50]$
5	DE-crossover [1]	$\begin{cases} F^{\text{DE}} \cdot \left(X^{\text{par}}_{ch1,g} - X^{\text{par}}_{ch2,g}\right) \\ \text{for} \quad (\alpha < CR) \text{ or } (ch = RInd) \\ 0 \quad \text{for} \quad \text{otherwise} \end{cases}$	$F^{\text{DE}} \in [0, 2]$, $CR \in [0, 1]$
6	BAT-movement [24]	$\text{U}(f_{\min}, f_{\min} + f_{\max}) \cdot \left(X^{\text{glb}}_{g} - X^{\text{par}}_{ch,g}\right)$	$f_{min} \in [0.0, 0.5]$, $f_{max} \in [0.0, 1.0]$
7	BAT-walk [24]	$\text{U}(-1,1) \cdot \sum_{ch3=1}^{Npop} \frac{A^t_{ch3}}{Npop}$	$A^t \in [0.0, 0.5]$, $\alpha^t \in [0.9, 1.0]$
8	FWA-explosion [22]	$\begin{cases} \frac{\text{U}(-1,1) \cdot \hat{A} \cdot (\text{ff}(\mathbf{x}_{ch}) - \text{ff}_{\min})}{\sum_{ch3=1}^{Npop} (\text{ff}(\mathbf{x}_{ch3}) - \text{ff}_{\min})} & \text{for} \quad ch \in RSet \\ 0 & \text{for} \quad \text{otherwise} \end{cases}$	$\hat{A} \in [0.1, 2.0]$
9	FWA-mutation [22]	$\begin{cases} X^{\text{par}}_{ch,g} \cdot \text{U}_G(1,1) - X^{\text{par}}_{ch,g} & \text{for} \quad ch \in RSet \\ 0 & \text{for} \quad \text{otherwise} \end{cases}$	
10	ICA-assimilation [2]	$d \cdot U(0,1) \cdot \left(X^{\text{imp}}_{ch,g} - X^{\text{par}}_{ch,g}\right)$	$d \in [0.5, 2]$
11	FA-movement [23]	$\begin{cases} \beta^{\text{FA}} \cdot \left(X^{\text{par}}_{ch1,g} - X^{\text{par}}_{ch,g}\right) \cdot \\ \cdot e^{-\gamma \cdot dist(\mathbf{X}_{ch1}, \mathbf{X}_{ch})^2} & \text{for} \quad \left(\begin{matrix} \text{ff}(\mathbf{X}_{ch1}) < \\ \text{ff}(\mathbf{X}_{ch}) \end{matrix}\right) \\ 0 & \text{for} \quad \text{otherwise} \end{cases}$	$\beta^{\text{FA}} \in [0.9, 1.1]$, $\gamma \in [0.01, 100]$
12	FA-walk [23]	$\alpha^{\text{FA}} \cdot U\left(-\frac{1}{2}, \frac{1}{2}\right)$	$\alpha^{\text{FA}} \in [0.01, 0.2]$

The component $\epsilon(\mathbf{X}_{ch})$ stands for standardized error of nonlinear modeling calculated as follows:

$$\epsilon(\mathbf{X}_{ch}) = \frac{1}{m} \cdot \sum_{j=1}^{m} \left(\frac{1}{Z} \cdot \sum_{z=1}^{Z} \left| d_{z,j} - \bar{y}_{z,j} \right| \right) \Big/ \left(\max_{z=1,\ldots,Z} \{d_{z,j}\} - \min_{z=1,\ldots,Z} \{d_{z,j}\} \right), \quad (9)$$

where $d_{z,j}$ is expected value of j-th output for z-th data sample ($z = 1, \ldots, Z$), Z stands for number of data samples, $\bar{y}_j$ is fuzzy system (5) output value calculated for data sample $\bar{\mathbf{x}}_z$. The purpose of normalization is to eliminate the differences between the errors of different outputs of the system (5) in case where $m > 1$.

The component $\delta(\mathbf{X}_{ch})$ stands for penalty of using too few or too many operators:

$$\delta(\mathbf{X}_{ch}) = \frac{1}{L^{op} - Nop} \left| \sum_{o=1}^{L^{op}} op_{ch,o} - Nop \right|, \tag{10}$$

where $Nop \geq 1$ stands for desired number of active operators.

2.4 Processing of the Individuals

The proposed method for processing the population taking into account the Eq. (2) allows use of any combination of evolutionary operators. This process is executed according to the following steps:

- Step 1. Initialization of population. In this step all genes of all *Npop* individuals are set randomly. It takes into account the specification of the problem under consideration and operators parameters ranges stated in Table 1. In case of genes $\mathbf{X}_{ch}^{\text{vel}}$ the assumptions that the genes do not exceed 20% ranges of the corresponding genes from $\mathbf{X}_{ch}^{\text{par}}$ are additionally taken.
- Step 2. Evaluation of population. This step aims to evaluate each individual by using fitness function (9).
- Step 3. Reproduction. It is based on mechanisms analogical to algorithm PSO (with improvements that makes generalization possible):
  - For each individual $\mathbf{X}_{ch}$ a copy is created (which allows to use of any selection method-see Step 4).
  - Binary genes of the copy $\mathbf{X}_{ch}^{\text{op}}$ are modified using crossover and mutation operators from genetic algorithm [20]. Parameters of these operators (p_m and p_c) are encoded in set $\mathbf{X}_{ch}^{\text{par}}$.
  - Real genes of the copy $\mathbf{X}_{ch}^{\text{vel}}$ and $\mathbf{X}_{ch}^{\text{par}}$ are updated according to (2). Genes $\mathbf{X}_{ch}^{\text{par}}$ are also repaired (narrowed down to specified boundaries).
  - The copies of $\mathbf{X}_{ch}$ are evaluated by fitness function (9). If the fitness function value of the copy is better than the fitness function value of the individual, the genes $\mathbf{X}_{ch}^{\text{bst}}$ of the copy are set as genes $\mathbf{X}_{ch}^{\text{par}}$ of the copy.
- Step 4. New generation selection. Proposed approach allows to use different strategy for selection new population individuals:
  - Strategy S1. In this strategy, the copy of an individual replaces the individual (as in the algorithm PSO).
  - Strategy S2. In this strategy, the copy of an individual replaces the individual only if the fitness function value of the copy is better than the fitness function value of the individual (similar to BAT algorithm [24]).
  - Strategy S3. In this strategy individuals and their modified copies are placed in temporary population, and then the best (according to fitness function) *Npop* individuals are selected for next generation.
- Step 5. Stop condition. In this step a stop condition is met (for example if the specified number of algorithm iterations is achieved). If this condition is not met, the algorithm goes back to Step 3, otherwise the best solution is presented and algorithm stops.

3 Simulations

The goals of the simulations were to test:

– Classic algorithm PSO ($X^{op}_{ch,g} = 1$, $g = 0, 1$, and $X^{op}_{ch,g} = 0$, $g = 2, 3, \ldots 11$), algorithm PSO enriched by using other evolutionary operators (PSO-OP) ($X^{op}_{ch,g} = 1$, $g = 0, 1$, and $X^{op}_{ch,g} \in \{0, 1\}$, $g = 2, 3, \ldots 11$) and proposed algorithm with flexible selectable evolutionary operators (OP) ($X^{op}_{ch,g} \in \{0, 1\}$, $g = 0, 1, \ldots 11$).
– Different strategies S1-S2 for selecting next generation of individuals, described in Sect. 2.4.

Table 2. Nonlinear benchmarks used in simulations.

Item no.	Problem name and reference	Short name	Number of inputs (n)	Number of outputs (m)	Number of data rows (Z)
1.	Auto MPG [19]	ampg	7	1	398
2.	Chemical Plant [21]	cplant	3	1	70
3.	Computer Hardware [14]	cpu	9	1	209
4.	Concrete Slump Test [7]	slump	7	3	103
5.	Yacht Hydrodynamics [18]	yacht	6	1	308

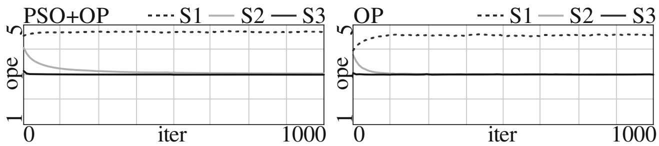

Fig. 1. Average number of used operators by single individuals (ope) averaged for all considered benchmarks. Dotted line stands for strategy S1, light line stands for strategy S2, and dark line stands for strategy S3.

The simulations were performed for typical benchmarks from nonlinear modeling field (see Table 2). The parameters of simulations were set as follows: number of fuzzy rules $N = 3$, number of discretization points $R = 3$, triangular norms: algebraic, number of individuals $Npop = 100$, number of empires $Nimp = 10$, number of expected operators $Nop = 3$, number of iterations: 1000, number of repeats of simulations for each problem and case: 20, selection method for choosing parents of the individual: roulette wheel.

The average simulation results are presented in Tables 3 and 4. The average number of used operators is shown in Figs. 2 and 3. The average number of used operators per individual is shown in Fig. 1.

The simulations conclusions are as follows:

– The best results in terms of the fitness function was obtained for OP method, regardless of the used strategy (see Table 3). This has a bearing on the results in terms of RMSE (see Table 4).

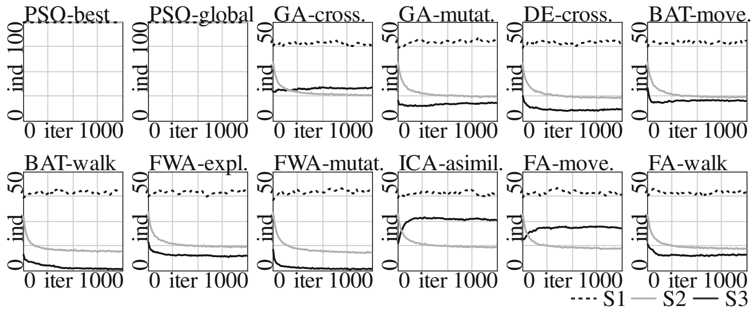

Fig. 2. Average number of individuals (ind) that use specified operators for algorithm PSO+OP averaged for all considered benchmarks. Dotted line stands for strategy S1, light line stands for strategy S2, and dark line stands for strategy S3.

Table 3. Averaged values of ff(.) for considered benchmarks. Best results for each strategy are shown in bold.

Strategy	Method	ampg	cplant	cpu	slump	yacht	Avg.
S1	PSO	0.0747	0.0891	0.0397	0.0788	0.0876	0.0740
	PSO+OP	0.0751	0.0798	0.0295	0.0782	0.0765	0.0678
	OP	**0.0546**	**0.0483**	**0.0190**	**0.0672**	**0.0531**	**0.0484**
S2	PSO	0.0822	0.0792	0.0458	0.0780	0.0900	0.0750
	PSO+OP	0.0497	0.0350	0.0174	0.0586	0.0398	0.0401
	OP	**0.0336**	**0.0081**	**0.0090**	**0.0435**	**0.0247**	**0.0238**
S3	PSO	0.0747	0.0679	0.0307	0.0762	0.0827	0.0664
	PSO+OP	0.0468	0.0289	0.0172	0.0551	0.0424	0.0381
	OP	**0.0330**	**0.0052**	**0.0066**	**0.0312**	**0.0255**	**0.0203**

Table 4. Averaged values of $\epsilon(\mathbf{X}_{ch})$ for considered benchmarks. Best results for each strategy are shown in bold.

Strategy	Method	ampg	cplant	cpu	slump	yacht	Avg.
S1	PSO	0.1383	0.1671	0.0515	0.1465	0.1641	0.1335
	PSO+OP	0.1224	0.1410	0.0313	0.1378	0.1437	0.1152
	OP	**0.0813**	**0.0689**	**0.0194**	**0.1066**	**0.1062**	**0.0765**
S2	PSO	0.1643	0.0843	0.0730	0.1561	0.1523	0.1260
	PSO+OP	0.0993	0.0701	0.0348	0.1173	0.0795	0.0802
	OP	**0.0671**	**0.0161**	**0.0179**	**0.0871**	**0.0494**	**0.0475**
S3	PSO	0.1494	0.0895	0.0613	0.1525	0.1470	0.1199
	PSO+OP	0.0935	0.0578	0.0344	0.1101	0.0848	0.0761
	OP	**0.0661**	**0.0104**	**0.0132**	**0.0624**	**0.0509**	**0.0406**

- The best results in terms of used fitness function were obtained for strategy S3, regardless of the used method. Strategy S1 used the largest number of operators, strategy S2 gradually reduced the number of used operators, while strategy S3 smoothly adjusted the number of used operators during the evolution of population (see Figs. 2 and 3).
- The best results was obtained for OP method and strategy S3 (see Tables 3 and 4). Such an approach allowed a dynamic matching of the number and type of used operators during the evolution of a population (see Fig. 3).
- The least used operators for OP method and strategy S3 were: PSO-best, BAT-walk, and FWA-mutation (after 50 iterations of algorithm, the usage of these operators has dropped to almost zero - see Fig. 3).
- The most used operators for OP method and strategy S3 were: DE-crossover, BAT-move, and FA-move (number of used operators increased during the evolution of population - see Fig. 3).
- The use of S1 strategy led to use over 4 operators by each individual in the population (see Fig. 1). For other strategies, the number of used operators sought to the *Nop* value. For the S2 strategy it was achieved after about 150 iterations (see Fig. 1), and for strategy S3 after about 30 iterations.

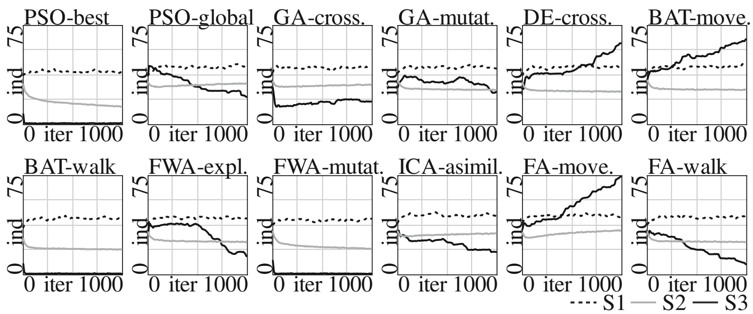

Fig. 3. Average number of individuals (ind) that use specified operators for algorithm OP averaged for all considered benchmarks. Dotted line stands for strategy S1, light line stands for strategy S2, and dark line stands for strategy S3.

4 Conclusions

The proposed in this paper population-based algorithm for nonlinear modeling with the selection of evolutionary operators allows, among others, for: (a) a precise control of the mechanisms of exploration and exploitation, (b) elimination of the need for selection of evolutionary operators by trial and error method, (c) reduction of the number of evolutionary operators, and (d) speeding up the optimization. This allows the algorithm to work well with other methods, especially fuzzy systems, for solving nonlinear modeling problems. Simulation studies have confirmed the effectiveness of presented approach.

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

The authors would like to thank the reviewers for very helpful suggestions and comments in the revision process.

References

1. Ali, M., Pant, M., Abraham, A.: Unconventional initialization methods for differential evolution. Appl. Math. Comput. **219**(9), 4474–4494 (2013)
2. Atashpaz-Gargari, E., Lucas, C.: Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. IEEE Congr. Evol. Comput. **7**, 4661–4666 (2007)
3. Bartczuk, Ł., Przybył, A., Koprinkova-Hristovak P.: New method for nonlinear fuzzy correction modelling of dynamic objects. Lecture Notes in Computer Science, vol. 8467, pp. 169–180 (2014)
4. Borzemski, L., Kliber, M., Nowak, Z.: Using data mining algorithms in web performance prediction. Cybern. Syst. Int. J. **40**(2), 176–187 (2009)
5. Brester, C., Semenkin, E., Sidorov, M.: Multi-objective heuristic feature selection for speech-based multilingual emotion recognition. J. Artif. Intell. Soft Comput. Res. **6**(4), 243–253 (2016)
6. Che, J., Wang, J., Li, K.: A Monte Carlo based robustness optimization method in new product design process: a case study. Am. J. Ind. Bus. Manag. **4**(7), 360–369 (2014)
7. Cheng, Y.I.: Modeling slump flow of concrete using second-order regressions and artificial neural networks. Cem. Concr. Compos. **29**(6), 474–480 (2007)
8. Corne, D.W., Knowles, J.D., Oates, M.J.: The Pareto envelope-based selection algorithm for multiobjective optimisation. In: Schoenauer, S., et al. (eds.) Parallel Problem Solving from Nature – PPSN VI, pp. 839–848. Springer, Berlin (2000)
9. Cpałka, K., Łapa, K., Przybył, A., Zalasiński, M.: A new method for designing neuro-fuzzy systems for nonlinear modelling with interpretability aspects. Neurocomputing **135**, 203–217 (2013)
10. Dziwiński, P., Bartczuk, Ł., Przybył, A., Avedyan, E.D.: A new algorithm for identification of significant operating points using swarm intelligence. Artificial Intelligence and Soft Computing, vol. 8468, pp. 349–362 (2014)
11. Ehrgott, M.: Multicriteria Optimization. Springer Verlag, Heidelberg (2000)
12. Ishibuchi, H., Nakashima, T., Murata, T.: Comparison of the Michigan and Pittsburgh approaches to the design of fuzzy classification systems. Electron. Commun. Jpn. **80**, 10–19 (1997)
13. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
14. Kibler, D., Aha, D.: Instance-based prediction of real-valued attributes. In: Proceedings of the CSCSI (Canadian AI) Conference, vol. 5, pp. 51–57 (1988)
15. Klement, E.P., Mesiar, R., Pap, E.: Triangular Norms. Springer, Heidelberg (2000)
16. Łapa, K., Cpałka, K., Wang, L.: A method for nonlinear fuzzy modelling using population based algorithm with flexibly selectable operators. Artificial Intelligence and Soft Computing, Lecture Notes in Computer Science. Springer (2017) (in print)
17. Łapa, K., Szczypta, J., Venkatesan, R.: Aspects of structure and parameters selection of control systems using selected multi-population algorithms. Artificial Intelligence and Soft Computing, vol. 9120, pp. 247–260 (2015)

18. Ortigosa, I., Lopez, R., Garcia, J.: A neural networks approach to residuary resistance of sailing yachts prediction. In: Proceedings of the International Conference on Marine Engineering (2007)
19. Quinlan, R.: Combining instance-based and model-based learning. In: Proceedings of the Tenth International Conference of Machine Learning, pp. 236–243 (1993)
20. Rutkowski, L.: Computational Intelligence. Methods and Techniques. Springer-Verlag, Heidelberg (2008)
21. Sugeno, M., Yasukawa, T.: A fuzzy-logic-based approach to qualitative modeling. IEEE Trans. Fuzzy Syst. **1**(1), 7–31 (1993)
22. Tan, Y., Zhu, Y.: Fireworks algorithm for optimization. In: Advances in Swarm Intelligence, pp. 355–364 (2010
23. Yang, X.S.: Nature-Inspired Metaheuristic Algorithms. Luniver Press, Bristol (2008)
24. Yang, X.S.: A new metaheuristic bat-inspired algorithm. In: Nature Inspired Cooperative Strategies for Optimization, NICSO 2010. SCI, vol. 284, pp. 65–74. Springer (2010)

Springer

http://www.springer.com/978-3-319-67219-9

Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017
Part I
Borzemski, L.; Świątek, J.; Wilimowska, Z. (Eds.)
2018, XVIII, 358 p. 157 illus., Softcover
ISBN: 978-3-319-67219-9