IMPROVED ANTLION OPTIMIZER ALGORITHM AND ITS PERFORMANCE ON NEURO FUZZY INFERENCE SYSTEM

H. Kilic, U. Yuzgec*, C. Karakuzu**

Abstract: Antlion optimizer algorithm (ALO) is inspired by hunting strategy of antlions. In this study, an improved antlion optimization algorithm is proposed for training parameters of adaptive neuro fuzzy inference system (ANFIS). In the standard ALO algorithm, the greatest deficiency is its long running time during optimization process. The random walking model of ants, the selection procedure and boundary checking mechanism have been developed to speed up standard ALO algorithm. To evaluate the performance of the improved antlion optimization algorithm (IALO), it has been tested on dynamic system modelling problems. ANFIS's parameters has been optimized by IALO algorithm to model five dynamic systems. ANFIS training procedure has been performed with 30 independent runs. Each training has been started with the random initial parameters of ANFIS and performance metrics have been obtained at the end of training. The results show that the IALO algorithm is able to provide competitive results in terms of mean, best, worst, standard deviation, training time metrics. According to the training time result, the proposed IALO algorithm has better performance than standard ALO algorithm and the average training time has been reduced to approximately 80 %.

Key words: *heuristic optimization, antlion optimizer, dynamic system modelling, neuro-fuzzy, ANFIS*

Received: July 4, 2018 **DOI:** 10.14311/NNW.2019.29.016
Revised and accepted: August 28, 2019

1. Introduction

Several heuristic algorithms have been developed over the past decades and they are becoming an important role on solving to the optimization problems in many engineering fields. Heuristic algorithms basically are based on some mechanisms from nature such as animal feeding habits, mating motivation or hunting techniques, etc. Heuristic algorithms are investigated into the four parts: physical-based algorithms, bio-inspired algorithms, evolutionary-based algorithms, swarm intelligence algorithms and other nature-inspired algorithms. The best known physical-based

*Haydar Kilic; Ugur Yuzgec – Corresponding author; Cihan Karakuzu; Department of Computer Engineering, Engineering Faculty, Bilecik Seyh Edebali University, 11210, Bilecik, Turkey, E-mail: haydar.kilic@bilecik.edu.tr, ugur.yuzgec@bilecik.edu.tr, cihan.karakuzu@bilecik.edu.tr

heuristic algorithms are Tabu Search and Simulated Annealing algorithms [30]. Artificial Immune algorithm [2] can be given as an example of the bio-inspired algorithm. Genetic Algorithm [29] and Differential Evolution algorithm [34] are the first examples that come to mind for evolutionary-based heuristic algorithms. In swarm intelligence algorithms, the search agents imitate the collective intelligence, such as flocks of birds, ant colonies, fish swarm. Ant Colony algorithm [6], Artificial Bee Colony algorithm [12], Particle Swarm Optimization algorithm [16] are the most popular of swarm intelligence algorithms.

The Antlion Optimizer (ALO) algorithm is a newly introduced heuristic algorithm inspired by antlion's hunting mechanism in nature. In this algorithm, it is recognized that the antlion larvae has a unique hunting mechanism, and this hunting mechanism is imitated. The antlion larvae pass larval periods by setting a hunting trap in the areas where the ant colonies are located. They bury themselves under a cone-shaped pit by spiral-shaped movements, and wait for the ants to come in. When the ants try to get out of the trap, the antlion throws sand and slides them into the bottom of trap. This interesting behaviour inspired Mirjalili's work [22]. After the announcement of the ALO, the scientists have begun working on its applications, on its imrovements and applied it to some optimization problems. Some of the optimization problems that work with ALO are Proportional – Integral (PI), Proportional – Integral – Derivative (PID), and Proportional – Integral – Derivative Plus Second Order Derivative (PID + DD) optimal controller design [28], optimal solution of non-convex and dynamic economic load dispatch problem of electric power system [10], optimal process plan according to all alternative manufacturing resources [27], optimal load dispatch problem [25], optimal route planning of unmanned aerial vehicle [36], governing loop of thermal generators [4], load frequency control of power systems [32], Optimal Reactive Power Dispatch (ORPD) problem [21], nonlinear electric economic power dispatch problem [35], adaptive identification of infinite impulse response (IIR) filters [23], parallel machine scheduling [17] and quadratic assignment problem [18].

The Adaptive Neuro Fuzzy Inference System (ANFIS) comprises combination of the neural network adaptive capabilities and the fuzzy logic qualitative approach. ANFIS is based on that a fuzzy system is trained by a learning algorithm derived from the neural network. These systems are capable of modeling the nonlinear relation between input and output of a system [19]. In the literature, there are different ANFIS applications in areas such as environmental engineering [37], health informatics [7], earth sciences [20], agricultural & biosystems engineering [33], synthesis of production processes [15]. The most important criterion regarding ANFIS structure is to be tuning its antecedent and consequent parameters, this procedure is known as the training phase. Some studies on training ANFIS model using different algorithms are as follows. Parameter tuning of fuzzy sliding mode controller using particle swarm optimization proposed by Karakuzu in 2010 [13]. A hybrid ANFIS training model introduced by Zangeneh et al. in 2011. The consequent parameters are trained by gradient descent and the antecedent parameters are trained by DE [38]. Jiang and his colleagues put forward a study aimed at producing satisfied products by optimizing the parameters suitable for customer satisfaction in 2012 by the PSO-based ANFIS structure. They tested the result of this work by applying the computer production [9]. Karaboga and his colleagues

compared learning performance of ANFIS using ABC algorithm, GA, backpropagation (BP) and hybrid learning (HL) in 2013 [11]. Kilic and his friends proposed the improved ALO algorithm via a tournament selection method for optimizing the parameters used in ANFIS [19].

In this study, some improvements are proposed to eliminate the shortcomings of the original ALO algorithm, such as long run time, local optima stagnation and premature convergence for some problems. In ALO algorithm, we have made the improvements on the hunting capture procedure and reconstruction of the antlion pit, random walking size, and the boundary control mechanism. To evaluate Improved ALO (IALO), the optimization problem of ANFIS parameters for dynamic system modelling, known as one of the difficult optimization problems, has been handled. The parameter training of neuro-fuzzy inference system has been realized with the proposed IALO algorithm, and its performance has been compared with other well-known algorithms known as Particle Swarm Optimizer (PSO), Differential Evolution (DE) algorithm, Simulating Annealing (SA) algorithm, Touring Ant Colony Optimizer (TACO), and Artificial Bee Colony (ABC) algorithm. The most important contribution of this study is to remove the disadvantages of ALO algorithm in solving optimization problems.

There are four primary objectives of this study:

1. to reveal the shortcomings of the original ALO algorithm;
2. to describe the proposed improvements on the ALO algorithm;
3. to evaluate the performance of the proposed IALO-ANFIS model; and
4. to analyse the statistical results between the proposed IALO algorithm and well-known heuristic algorithms for ANFIS training and testing.

2. Antlion Optimizer (ALO)

ALO algorithm is one of the heuristic algorithms that is inspired by antlions larvae's own hunting technique. Antlions use this interesting hunting technique as larvae, usually to hunt by setting traps on a field where ants are found. The skill of building their traps has led to the development of this algorithm by Seyedali Mirjalili [22]. These great antlions dig their traps in spiral paths and turn them into a cone pit, then bury themselves at the bottom of trap. They wait for their prey to come, when the ants enter the trap they try to escape, but the antlions throw sand from under the pit and slide them down, finally grabbing them with their strong jaws. The illustration of this hunting mechanism is given in Fig. 1.

The mathematical description of antlion optimizer is given briefly as follows. After initializing the antlion positions in the population, random walks becomes by the following equation for the ant behavior:

$$\mathbf{X}(t) = [0, \text{cumsum}(2r(t_1) - 1), \text{cumsum}(2r(t_2) - 1), \ldots, \text{cumsum}(2r(t_n) - 1)], \quad (1)$$

where $\mathbf{X}(t)$ denotes the random walk at t step, cumsum refers cumulative sum, n is the maximum number of iteration, t is the step of the random walk, and $r(t)$ is

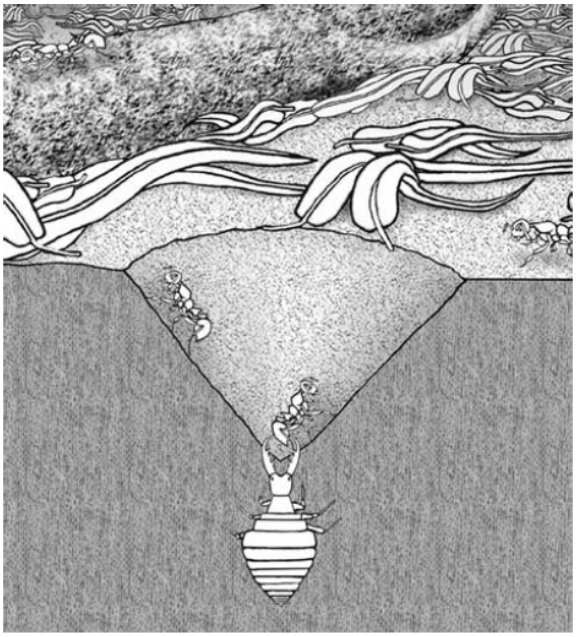


Fig. 1 *Antlion's trap [39].*

the stochastic function as defined:

$$r(t) = \begin{cases} 1 & \text{if rand} > 0.5 \\ 0 & \text{if rand} \leq 0.5 \end{cases}, \tag{2}$$

where rand is the number in interval [0,1]. According to the random walk mechanism, ants update their positions at each step. Fig. 2 shows that three different random walks over 1000 iterations.

In order to keep random walks of ant in the search space, it has to be normalized by the following equation:

$$\mathbf{X}_i^t = \frac{(\mathbf{X}_i^t - a_i)(d_i^t - c_i^t)}{b_i - a_i} + c_i^t, \tag{3}$$

where t denotes the iteration number, i represents the variable index, a stands for the minimum random walk, b represents the maximum random walk value, c denotes minimum variable value, and d is the maximum variable value of antlion's position updated at each iteration as given below.

Ants' walk affected by the antlions, when the ants enter the trap, antlion becomes shoot sands over the ants and slide them down the pit, the following mathematical model explains that:

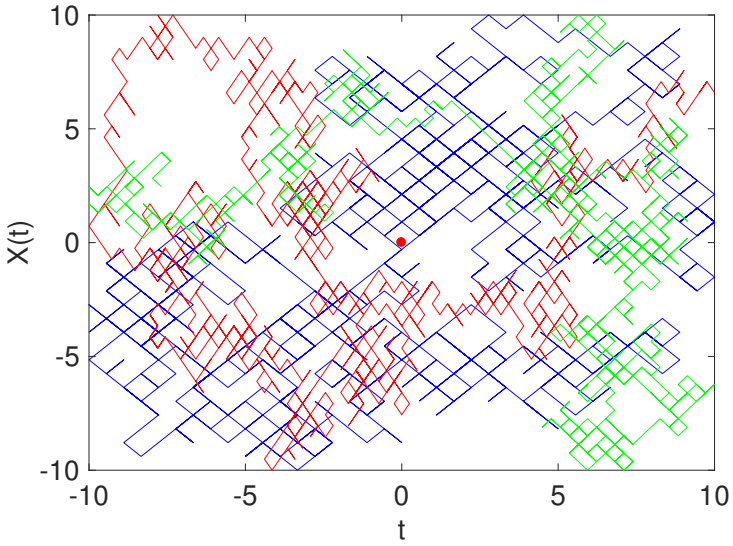


Fig. 2 *Three different random walks for ant behavior, each color shows the different random walk model of the ant and red dot denotes the ant's starting position.*

$$c_i^t = \text{Antlion}^t + c^t, \tag{4}$$

$$d_i^t = \text{Antlion}^t + d^t, \tag{5}$$

$$c^t = \frac{c^t}{I}, \tag{6}$$

$$d^t = \frac{d^t}{I}, \tag{7}$$

where Antlion^t is the position of the selected antlion at t-th iteration, c^t stands for the minimum of all variables at t-th iteration, d^t is the maximum of all variables at t-th iteration, c_i^t denotes the minimum of all variables for i-th ant, d_i^t denotes the maximum of all variables for i-th ant, and I is the sliding ratio that can be changed in different scenarios as follows:

$$I = \begin{cases} 1 + 10^2 \frac{t}{T} & \text{if } 0.1T < t < 0.5T \\ 1 + 10^3 \frac{t}{T} & \text{if } 0.5T < t < 0.75T \\ 1 + 10^4 \frac{t}{T} & \text{if } 0.75T < t < 0.9T \\ 1 + 10^5 \frac{t}{T} & \text{if } 0.9T < t < 0.95T \\ 1 + 10^6 \frac{t}{T} & \text{if } 0.95T < t < T \\ 1 & \text{otherwise,} \end{cases} \tag{8}$$

where t denotes the current iteration, and T represents the maximum iteration. After finding X_i^t from the Eq. (3), ants move around to roulette wheel selected antlion and elite antlion in the current population. Ants' new positions are determined by the following equation:

$$\mathrm{Ant}_i^t = \frac{R_A^t + R_E^t}{2}, \tag{9}$$

where R_A^t is ants' random walks around the antlion chosen by roulette wheel, and R_E^t is ants' random walks around the elite antlion. $\mathrm{Antlion}_j^t$ is the position of the selected j-th antlion at t-th iteration. After consuming prey, antlion is required to update its position according to i-th ant at t-th iteration Ant_i^t by the following equation:

$$\mathrm{Antlion}_j^t = \mathrm{Ant}_i^t \quad \text{if} \quad f(\mathrm{Ant}_i^t) < f(\mathrm{Antlion}_j^t), \tag{10}$$

where f represents the fitness function. ALO algorithm starts with the same number of ant and antlion during the optimization process. The positions of ants are kept in the $\mathbf{M}_{\mathrm{ant}}$ matrix and utilized to solve the optimization problem.

$$\mathbf{M}_{\mathrm{ant}} = \begin{bmatrix} \mathrm{Ant}_{1,1} & \mathrm{Ant}_{1,2} & \dots & \mathrm{Ant}_{1,d} \\ \mathrm{Ant}_{2,1} & \mathrm{Ant}_{2,2} & \dots & \mathrm{Ant}_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ \mathrm{Ant}_{n,1} & \mathrm{Ant}_{n,2} & \dots & \mathrm{Ant}_{n,d} \end{bmatrix}, \tag{11}$$

where $\mathrm{Ant}_{i,j}$ represents the j-th dimension's value of i-th ant. n stands for the number of ants, and d denotes the number of variables (dimension). In addition to ants, the positions of antlions are saved in the $\mathbf{M}_{\mathrm{antlion}}$ matrix as given below:

$$\mathbf{M}_{\mathrm{antlion}} = \begin{bmatrix} \mathrm{Antlion}_{1,1} & \mathrm{Antlion}_{1,2} & \dots & \mathrm{Antlion}_{1,d} \\ \mathrm{Antlion}_{2,1} & \mathrm{Antlion}_{2,2} & \dots & \mathrm{Antlion}_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ \mathrm{Antlion}_{n,1} & \mathrm{Antlion}_{n,2} & \dots & \mathrm{Antlion}_{n,d} \end{bmatrix}, \tag{12}$$

where $\mathrm{Antlion}_{i,j}$ represents the value of j-th dimension of i-th antlion. The fitness values of all ants and antlions ($\mathbf{F}_{\mathrm{ant}}$ and $\mathbf{F}_{\mathrm{antlion}}$) in the population are calculated and stored in the following matrices:

$$\mathbf{F}_{\mathrm{ant}} = \begin{bmatrix} f(\mathrm{Ant}_{1,1}, \mathrm{Ant}_{1,2}, \dots, \mathrm{Ant}_{1,d}) \\ f(\mathrm{Ant}_{2,1}, \mathrm{Ant}_{2,2}, \dots, \mathrm{Ant}_{2,d}) \\ \vdots \\ f(\mathrm{Ant}_{n,1}, \mathrm{Ant}_{n,2}, \dots, \mathrm{Ant}_{n,d}) \end{bmatrix}, \tag{13}$$

$$\mathbf{F}_{\mathrm{antlion}} = \begin{bmatrix} f(\mathrm{Antlion}_{1,1}, \mathrm{Antlion}_{1,2}, \dots, \mathrm{Antlion}_{1,d}) \\ f(\mathrm{Antlion}_{2,1}, \mathrm{Antlion}_{2,2}, \dots, \mathrm{Antlion}_{2,d}) \\ \vdots \\ f(\mathrm{Antlion}_{n,1}, \mathrm{Antlion}_{n,2}, \dots, \mathrm{Antlion}_{n,d}) \end{bmatrix}, \tag{14}$$

where f denotes the fitness (objective) function. Main steps of ALO algorithm are given in Algorithm 1 [22].

Algorithm 1 Pseudo code of original ALO algorithm.

```
Initialize the first population of ants and antlions randomly.
Calculate the fitness of ants and antlions
Find the elite antlion
while the end criterion is not satisfied do
  for every ant do
    Select an antlion by roulette wheel
    Update c and d using Eqs. (6) and (7).
    Create a random walk and normalize it using Eqs. (1) and (3).
    Update the position of ant using Eq. (9).
  end for
  Calculate the fitness of all ants.
  Replace an antlion with its corresponding ant if it becomes fitter by Eq. (10).

  Update elite if an antlion becomes fitter than the elite.
end while
return elite
```

3. Improved Antlion Optimizer (IALO)

Roulette wheel method is a selection method used in most heuristic algorithms. The whole wheel is divided randomly into certain parts, the size of these parts is determined by the magnitude ratios, the wheel part with great fitness value is more likely to win [29]. In the original ALO algorithm, a roulette wheel operator is utilized to select an antlion during optimization and every ant walks randomly around this antlion selected by the roulette wheel. This mechanism gives high chances to the better antlions to hunt ants [22]. The roulette wheel strategy is summarized in Algorithm 2.

Algorithm 2 Roulette Wheel Selection.

STEP1: Roulette wheel get split up randomly in size of population.
STEP2: The cumulative sum of the fitness values of each individual in the population is calculated.
STEP3: The ratio of each individual's fitness value to the cumulative sum that found in STEP2 is winning probability of that individual.

For the optimization problems with the negative fitness values, the first antlion in the population is always selected by roulette wheel selection method and this affects to random walk mechanism negatively. This problem is solved by using the absolute value of the fitness values to select different antlions using the roulette wheel. The mathematical description regarding the probability of selection of each individual in the IALO algorithm is given below:

$$p_i = \frac{\left| \frac{1}{f(\text{Antlion}_i)} \right|}{\sum_{i=1}^{Np} \left| \frac{1}{f(\text{Antlion}_i)} \right|}, i = 1, 2, \ldots, Np, \tag{15}$$

where Np indicates the number of population size, Antlion_i denotes i-th antlion in the population, f stands for the fitness value, and p_i is the probability of selection of i-th antlion.

The random walk of the original ALO algorithm gives the model of the ants' movements by the maximum iteration number, and this model influences the run time of the algorithm. Therefore, the second improvement is made about random walks, the size of the random walks is reduced, and recalculated at each iteration. Also some variables are calculated in the same way at each iteration in the loop, this is unnecessary. These variables have been taken out of the loop because it causes unnecessary time loss. Thus, the positions of the ants around the antlion selected by the roulette wheel and the elite antlion are updated. As given in Eq. 16, the distance of the random walk size is reduced as 20 % of the maximum iteration number ($It_{\max}$)

$$X(t) = [0, \text{cumsum}(2r(t_1) - 1), \ldots, \text{cumsum}(2r(t_n) - 1)], n = 1, 2, \ldots, It_{\max}/5. \tag{16}$$

Ants and antlions are merged in the same population and sorted by the fitness values in the elite selection mechanism of original ALO algorithm. In the proposed IALO, there is no sorting mechanism regarding fitness values of ants and antlions, the fitness values of antlions are compared with the fitness values of the ants at the end of iterations. If the fitness value of ant is better than the fitness value of antlion, then ant becomes antlion. In Algorithm 3, the pseudo code of the proposed new selection mechanism of IALO algorithm is given.

Algorithm 3 IALO selection mechanism.

```
for every antlion do
  Compare ant's fitness and antlion's fitness for each individual.
  if ant fitness is better than antlion fitness then
    Replace antlion position with ant position
  end if
  Do not change Antlion position
end for
```

The novelty of setting trap of antlion is changing conditions as shown Eqs. 17–20, where ζ is randomly chosen number in $[0, 1]$.

$$\left.\begin{aligned} c_i^t &= \text{Antlion}_i^t + c^t \\ d_i^t &= \text{Antlion}_i^t + d^t \end{aligned}\right\} \text{if } 0.75 < \zeta < 1, \tag{17}$$

$$\left.\begin{aligned} c_i^t &= \text{Antlion}_i^t - c^t \\ d_i^t &= \text{Antlion}_i^t - d^t \end{aligned}\right\} \text{if } 0.5 < \zeta < 0.75, \tag{18}$$

$$\left.\begin{aligned} c_i^t &= -\text{Antlion}_i^t + c^t \\ d_i^t &= -\text{Antlion}_i^t + d^t \end{aligned}\right\} \text{if } 0.25 < \zeta < 0.5, \tag{19}$$

$$\left.\begin{aligned} c_i^t &= -\text{Antlion}_i^t - c^t \\ d_i^t &= -\text{Antlion}_i^t - d^t \end{aligned}\right\} \text{otherwise.} \tag{20}$$

In the original ALO, if the ants go out of bounds, they are placed over the bounds and returned to the search area. As the last improvement on ALO algorithm, the ants that are outside of the search space are relocated randomly in the search space. If ants exceed the boundaries, boundary check in Eq. 21 bring them back inside into the search space.

$$\text{Ant}_i^t = b_{\text{low}} + \text{rand} \times (b_{\text{up}} - b_{\text{low}}) \quad \text{if} \quad \left(\text{Ant}_i^t > b_{\text{up}}\right) \text{OR} \left(\text{Ant}_i^t < b_{\text{low}}\right), \tag{21}$$

where rand is a randomly chosen number in interval of $[0, 1]$, b_{low} is lower boundary, b_{up} is upper boundary. In Algorithm 4, the pseudo-code of the proposed IALO algorithm is given in detail.

Algorithm 4 Pseudo-code of IALO algorithm.

```
Initialize the positions of antlions in the population randomly.
Calculate the fitness values of antlions using objective function.
Save the best antlion according to the fitness values.
repeat
  Calculate X(t) using Eq. 16.
  for every ant do
    Select antlion by roulette wheel method for building a trap (Eq. 15).
    Slide randomly walk ants in the trap using Eqs. 17–20.
    Generate ant's random walk route around elite antlion (R_A^t) and selected
    antlion (R_A^t) by roulette wheel.
    Normalize random walks for elite antlion and selected antlion.
    Calculate the ant position (Ant_i^t) using Eq. 9.
    if Ant is out of the search space then
      Relocate the ant in search area (Eq. 21).
    end if
  end for
  Calculate the fitness values of ants.
  for every antlion do
    if Fitness of ant is better than that of antlion then
      antlion hunts ant (update the antlion's position).
    end if
  end for
  Update the elite antlion.
until the stopping criterion is met
return the elite antlion.
```

4. ANFIS Structure

Artificial Neural Networks (ANN) and Fuzzy Inference Systems (FIS) are known as soft computing techniques. FIS is a system that uses fuzzy membership functions

to make a decision. There are many applications of FIS in different areas. However, FIS has not the learning capacity. ANN has a strong ability to learn thanks to the weighted connections between the neurons. For these reasons, the advantage of ANN and FIS can be integrated into a neuro-fuzzy approach. ANFIS presented by Jang [8] is an intelligent neuro-fuzzy structure used in modelling nonlinear functions and estimating chaotic time series. ANFIS structure including the input and output of the linguistically specified rules, is connected to a set of parameters in each layer by the adaptive neural network nodes. The ANFIS model consists of five layers, which can also be described as a multilayer neural network, as shown in Fig. 3. ANFIS model generally uses a hybrid learning algorithm and depending on the number and type of membership function selected, it is constructed by the hybrid learning algorithm. ANFIS is more effective than ANN and FIS because of the fact that it includes the features of ANN and Fuzzy systems.

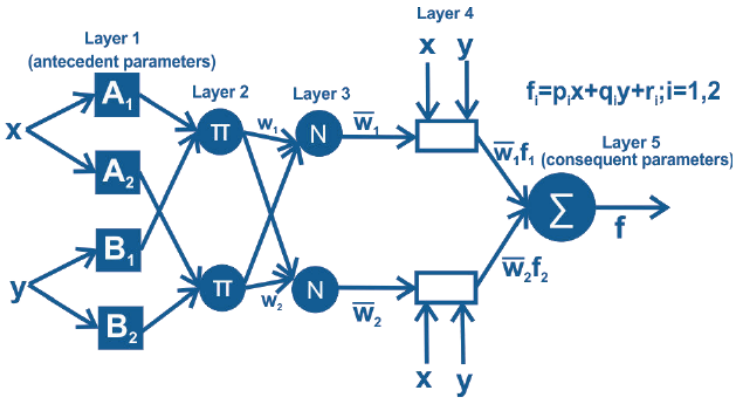


Fig. 3 *General ANFIS structure with two inputs.*

To give the mathematical equations of five layers in the ANFIS structure including two inputs and one output, we assume two fuzzy sets. The membership functions of these fuzzy sets have the shape of a two-dimensional Gaussian distribution. The centers of Gaussian membership functions are given as a_i parameters. b_i parameters determining the width of these fuzzy sets in the x and y directions. In Eq. 22 and Eq. 23, the membership functions of these fuzzy sets are calculated for data (x,y).

Layer 1:

$$O_{1,i} = \mu_{A_i}(x) = \exp\left(-\frac{(x-a_i)^2}{2b_i^2}\right); i = 1,2, \tag{22}$$

$$O_{1,i} = \mu_{B_{i-2}}(y) = \exp\left(-\frac{(y-a_i)^2}{2b_i^2}\right); i = 3,4, \tag{23}$$

a_i and b_i in this layer are called antecedent parameters. μ_{A_i} and μ_{B_i} denote membership function. x and y represent the external inputs. i denotes the index number.

Layer 2:

$$O_{2,i} = w_i = \mu_{A_i}(x)\,\mu_{B_i}(y)\,. \tag{24}$$

The output of each node stands for the firing strength w_i of the rule, generally the algebraic product T-norm operator is applied as the node function.
Layer 3:

$$O_{3,i} = \overline{w}_i = \frac{w_i}{w_1 + w_2}, i = 1, 2. \tag{25}$$

In each node, the ratio of that node rule firing strength (w_i) to sum of all the firing strength ($w_1 + w_2$) is found. The output of this layer is called normalized firing strengths.
Layer 4:

$$O_{4,i} = \overline{w}_i f_i = \overline{w}_i \left(p_i x + q_i y + r_i\right). \tag{26}$$

Each node in this layer is the adaptive node with their node functions. The parameters p_i, q_i, r_i in this layer stand for the consequent parameters.
Layer 5:

$$O_{5,1} = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}. \tag{27}$$

The single node in the last layer is known as fixed node that calculates the sum of the all previous layers' outputs, where f_i are rules.

5. IALO performance test results

5.1 ANFIS parameter learning by IALO algorithm

In this section, ANFIS parameters for modelling of dynamic systems have been optimized by the proposed algorithm and the other well known algorithms, then tested and performance evaluations have been carried out. The parameters of the algorithms used in this study are given in Tab. I and the calculation mechanism is shown as block diagram in Fig. 4.

In this study, ANFIS learning for modelling of dynamical systems is carried out using heuristic algorithms. Dynamical systems are summarized in Tab. II. For the first and second data sets, a cosine and sine functions are utilized as much as the iteration size, and for other data sets, the random number generator is used in the ranges given in this table. In Tab. III, parameter numbers of ANFIS for each dynamical system are given. Detailed information about Tab. II can be obtained from [14].

Algorithms	Parameters
PSO [16]	Constriction factor=0.7298,learning coefficients = 2.05
ABC [12]	limit cycle=100
SA [30]	Temperature=current iteration/maximum iteration number
DE [34]	Crossover probability=0.5, differential weight=0.8, differential strategy=DE/rand/1/bin
TACO [6]	Vaporing=0.1, number of bit=18
ALO [22]	No parameter
IALO	random walk size =Max Iter/5

Tab. I *Parameters of heuristic algorithms used in ANFIS training.*

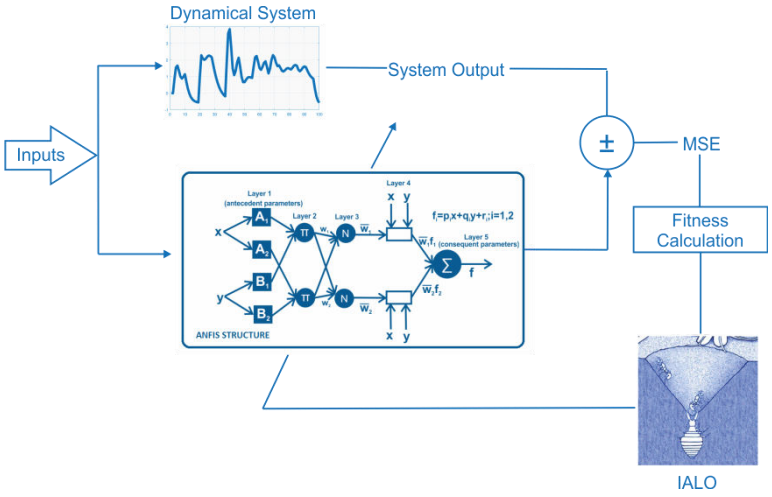


Fig. 4 *ANFIS training by IALO algorithm.*

Sys.No.	Dynamic System	Training Set	Testing Set
1	$y(k)=\frac{y(k-1)y(k-2)(y(k-1)+2.5)}{1+y^2(k-1)+y^2(k-2)}+u(k)$ [24]	$u(k)=\cos(2\pi k/100)$	$u(k)=\sin(2\pi k/25)$
2	$y(k+1)=\frac{y(k)}{1+y^2(k)}+u^3(k)$ [24]	$u(k)=\cos(2\pi k/100)$	$u(k)=\sin(2\pi k/25)$
3	$y(k+1)=y(k)+u(k)e^{-3\lvert y(k)\rvert}$ [1]	$[-1,1]$	$[-1,1]$
4	$y(k+1)=\frac{24+y(k)}{30}y(k)-0.8\frac{u^2(k)}{1+u^2(k)}y(k-1)+0.5u(k)$ [26]	$[-5,5]$	$[-5,5]$
5	$y(k+1)=0.5(\frac{y(k)}{1+y^2(k)}+(1+u(k))u(k)(1-u(k)))$ [31]	$[-2,2]$	$[-2,2]$

Tab. II *Dynamic systems for training and testing.*

Sys.No.	Inputs	# Membership Func.	# Rules	# Total Parameters
1	$u(k), y(k-2), y(k-1)$	6 {2,2,2}	8	44
2	$u(k), y(k), y(k-1)$	6 {2,2,2}	8	44
3	$u(k), y(k)$	4 {2,2}	4	20
4	$u(k), y(k), y(k-1)$	6 {2,2,2}	8	44
5	$u(k), y(k)$	4 {2,2}	4	20

Tab. III *Parameter numbers of ANFIS used for modelling of dynamic systems.*

In the optimization of ANFIS parameters, the squared errors are obtained from ANFIS output and system output at the end of each iteration. The mean of these squared errors is used as fitness value of each individual in used heuristic algorithms. In this study, each heuristic algorithm was run 30 times to train ANFIS. The population size is calculated by the following equation given in [5], where round is the function that rounds to the nearest decimal or integer and D is the dimension of the problem (total parameters in Tab. III).

$$\text{Population Size} = \text{round}\left(10 + 2\sqrt{D}\right). \tag{28}$$

5.2 Statistical analysis

To evaluate the performance of the IALO algorithm, we have used some statistical metrics, such as mean squared error (MSE), statistical best(SB), and statistical worst(SW). Their formulas are given in Eqs. 29–31, where $\hat{x}_i$ is the prediction value produced by ANFIS on training at i-th iteration, x_i is corresponding target value. Also f is the optimal solution's fitness value of each run and m is the number of run.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{x}_i - x_i\right)^2, \tag{29}$$

$$\mathrm{SB} = \mathrm{Min}_{i=1}^{m} f, \tag{30}$$

$$\mathrm{SW} = \mathrm{Max}_{i=1}^{m} f. \tag{31}$$

In this study, we have utilized the Mann-Whitney test, which is a non-parametric statistical test for two independent groups [3]. In the Mann-Whitney test, it is assumed that IALO algorithm is worse than the compared algorithms as the null hypothesis. The alternative hypothesis considers that the compared algorithm is significantly worse than IALO algorithm.

5.3 Validation of the proposed model

This sub-section presents non-parametric statistical test results to show the significance of the results of the proposed IALO algorithm. This test is unpaired Mann-Whitney test of medians. The p-values less than 0.05 can be regarded as strong proof against the null hypothesis. To show the significant differences between the MSE results of proposed IALO algorithm versus other heuristic algorithms, Tab. IV summarizes the p-values obtained by a one-tailed variant of Mann-Whitney test with 5 % degree. p-values ≤ 0.05 are shown in bold face. As can be seen from the Mann-Whitney test results, IALO algorithm shows a significant improvement over TACO, PSO, DE, and SA, with a level of 5 % significance.

IALO	ABC	DE	PSO	TACO	SA	ALO
System 1	9.2936e-01	**2.8730e-02**	**7.3215e-11**	**7.0334e-05**	**2.8730e-02**	9.7316e-01
System 2	6.6325e-01	**8.0311e-07**	**1.3049e-10**	**5.3328e-08**	**2.3186e-03**	4.6760e-01
System 3	9.9999e-01	9.9991e-01	**6.0283e-11**	**2.6325e-05**	3.6970e-01	9.9152e-01
System 4	1.0000e+00	1.0000e+00	**7.3173e-11**	**1.5051e-07**	**5.6139e-03**	9.9995e-01
System 5	1.0000e+00	**1.4457e-03**	**2.3428e-08**	**1.0029e-04**	4.0364e-01	9.9651e-01

Tab. IV *p-values of the Mann-Whitney test with 5 % significance.*

5.4 ANFIS results using IALO algorithm

In this sub-section, we present the ANFIS training and test results for five dynamic systems using the proposed IALO and other heuristic algorithms. Fig. 5 shows the boxplot of the fitness values for ANFIS training results with 30 runs of five dynamic systems.

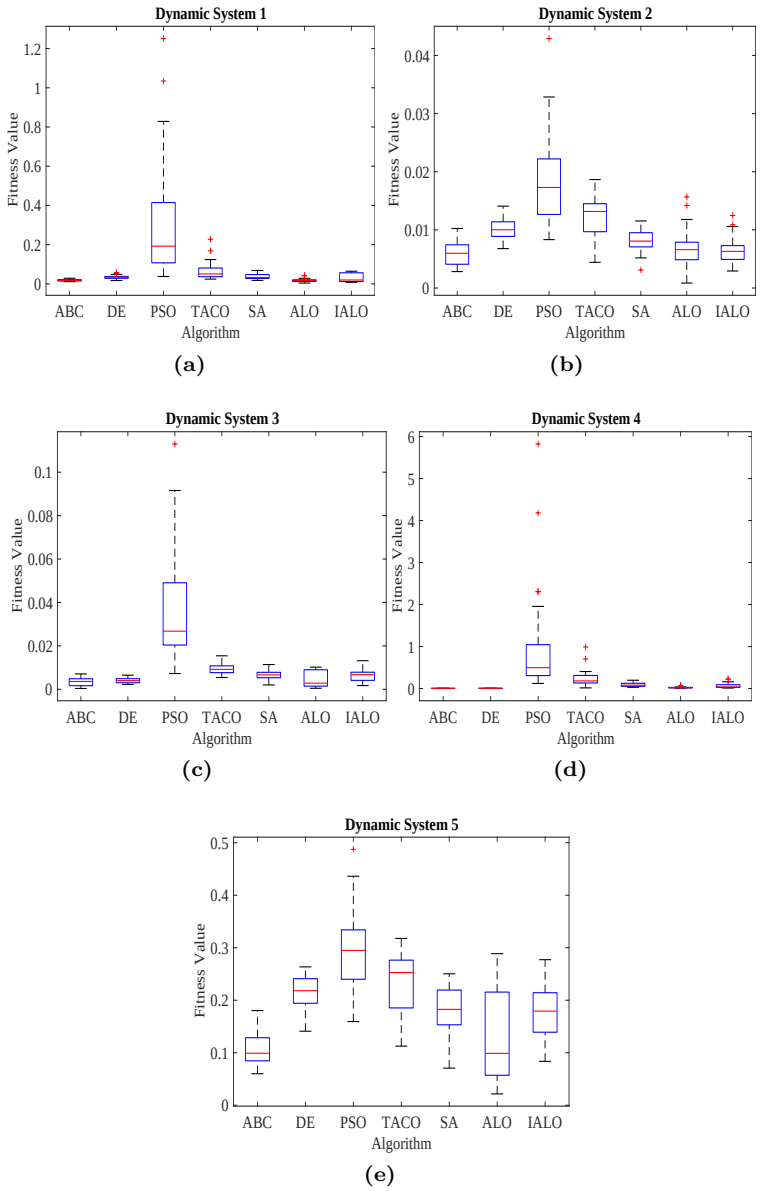


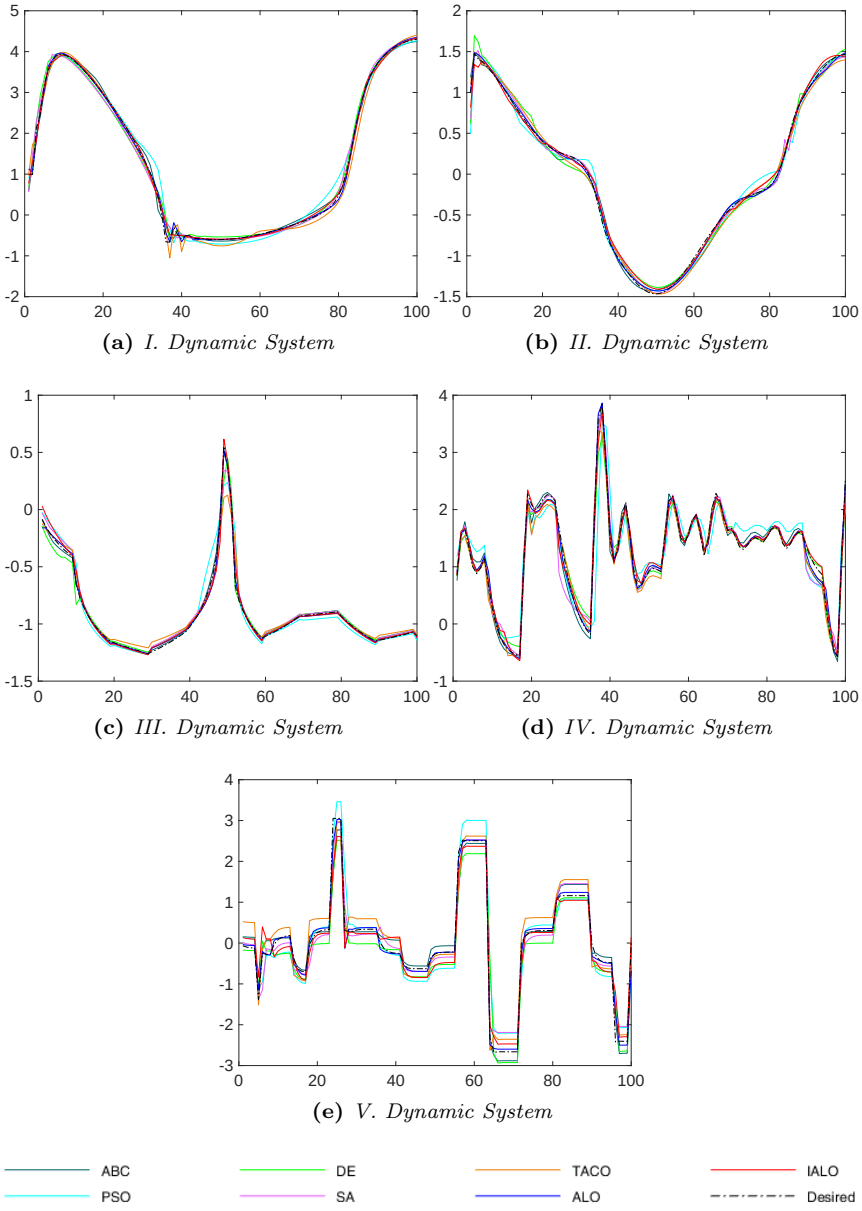
Fig. 5 *Boxplots of fitness values in ANFIS training results for dynamic systems.*

According to these figures, the IALO algorithm has competitive results in comparison with the other heuristic algorithm.The PSO algorithm has the worst performance among the other algorithms. The detailed results of all algorithms are given in Tab. V. In this table, *R., Av.R.,Av.Cum.R* refer to rank, average rank and average cumulative rank respectively. The table organized in two parts, one of which is *Train*, while the other is *Test* for each heuristic optimization algorithm.

Algorithms		Metrics	Sys. 1	R.	Sys. 2	R.	Sys. 3	R.	Sys. 4	R.	Sys. 5	R.	Av. R.	Av. Cum. R.
PSO	*Train*	*Time*	2.866	4	2.596	2	1.535	2	2.659	3	1.554	3	2.8	5.971
		Best	0.033	7	0.007	7	0.004	7	0.144	7	0.125	6	6.8	
		Worst	5.201	7	0.063	7	0.129	7	10.441	7	0.345	7	7.0	
		Mean	0.840	7	0.019	7	0.026	7	1.622	7	0.250	7	7.0	
		St. Dev.	1.338	7	0.010	7	0.026	7	2.431	7	0.060	5	6.6	
	Test	*Best*	0.082	6	0.027	4	0.353	7	0.282	6	0.106	5	5.6	
		Worst	4.781	7	0.282	7	0.187	5	14.423	7	0.240	4	6.0	
ABC	*Train*	*Time*	1.370	1	1.418	1	0.804	1	1.345	1	0.810	1	1.0	1.686
		Best	0.006	3	0.002	1	0.001	2	0.015	3	0.030	2	2.2	
		Worst	0.028	1	0.010	1	0.006	1	0.064	2	0.173	1	1.2	
		Mean	0.016	1	0.005	1	0.003	1	0.038	2	0.098	1	1.2	
		St. Dev.	0.005	1	0.002	2	0.001	2	0.012	2	0.037	3	2.0	
	Test	*Best*	0.007	1	0.012	1	0.319	6	0.038	2	0.072	4	2.8	
		Worst	0.034	1	0.048	2	0.067	2	0.147	1	0.137	1	1.4	
SA	*Train*	*Time*	2.627	3	2.643	3	1.550	4	3.574	4	1.848	4	3.6	4.000
		Best	0.009	4	0.003	4	0.004	5	0.027	5	0.097	5	4.6	
		Worst	0.061	3	0.011	3	0.011	4	0.185	4	0.246	2	3.2	
		Mean	0.037	4	0.008	4	0.007	5	0.106	5	0.180	4	4.4	
		St. Dev.	0.013	4	0.002	3	0.002	3	0.040	4	0.035	2	3.2	
	Test	*Best*	0.033	3	0.032	5	0.096	5	0.339	7	0.156	7	5.4	
		Worst	0.475	6	0.046	1	0.077	4	0.727	5	0.196	2	3.6	
DE	*Train*	*Time*	2.586	2	2.656	4	1.537	3	2.634	2	1.472	2	2.6	3.114
		Best	0.025	5	0.007	6	0.002	3	0.011	2	0.142	7	4.6	
		Worst	0.065	5	0.015	4	0.008	2	0.060	1	0.274	4	3.2	
		Mean	0.039	5	0.010	5	0.005	3	0.039	3	0.210	5	4.2	
		St. Dev.	0.008	3	0.002	1	0.001	1	0.010	1	0.027	1	1.4	
	Test	*Best*	0.051	5	0.015	2	0.014	1	0.032	1	0.126	6	3.0	
		Worst	0.097	2	0.056	4	0.067	3	0.174	2	0.217	3	2.8	
TACO	*Train*	*Time*	16.893	6	16.581	6	6.797	6	16.59	6	6.728	6	6.0	5.600
		Best	0.029	6	0.006	5	0.004	6	0.048	6	0.067	3	5.2	
		Worst	0.169	6	0.018	6	0.016	5	0.522	6	0.338	6	5.8	
		Mean	0.077	6	0.013	6	0.009	6	0.217	6	0.217	6	6.0	
		St. Dev.	0.034	6	0.003	6	0.003	5	0.13	6	0.067	6	5.8	
	Test	*Best*	0.442	7	0.047	7	0.021	2	0.153	5	0.046	2	4.6	
		Worst	0.301	5	0.069	5	0.621	6	0.952	6	0.404	7	5.8	
ALO	*Train*	*Time*	18.636	7	18.894	7	7.983	7	19.666	7	7.995	7	7.0	3.429
		Best	0.005	2	0.002	3	0.001	1	0.005	1	0.017	1	1.6	
		Worst	0.033	2	0.011	2	0.009	3	0.067	3	0.251	3	2.6	
		Mean	0.017	2	0.006	3	0.004	2	0.029	1	0.125	2	2.0	
		St. Dev.	0.007	2	0.002	4	0.003	4	0.017	3	0.068	7	4.0	
	Test	*Best*	0.041	4	0.034	6	0.075	4	0.084	3	0.017	1	3.6	
		Worst	0.114	3	0.053	3	0.016	1	0.688	4	0.257	5	3.2	
IALO	*Train*	*Time*	3.775	5	3.869	5	2.186	5	3.692	5	2.036	5	5.0	4.200
		Best	0.004	1	0.002	2	0.003	4	0.015	4	0.082	4	3.0	
		Worst	0.064	4	0.017	5	0.019	6	0.224	5	0.312	5	5.0	
		Mean	0.026	3	0.006	2	0.006	4	0.090	4	0.164	3	3.2	
		St. Dev.	0.020	5	0.003	5	0.004	6	0.074	5	0.050	4	5.0	
	Test	*Best*	0.021	2	0.022	3	0.044	3	0.118	4	0.057	3	3.0	
		Worst	0.224	4	0.076	6	1.465	7	0.541	3	0.274	6	5.2	

Tab. V *Performance results of all algorithms on ANFIS training and testing.*

In the measurement column, *Best* and *Worst* are refers to the fitness values of best solution and worst solution among 30 runs respectively. *St.Dev.* is standard deviation of all solutions' fitness values, *Average* is average of them. Fig. 6 and Fig. 7 show training and test results of ANFIS model having the best parameters obtained from 30 runs with 500 iterations, for five dynamical systems.

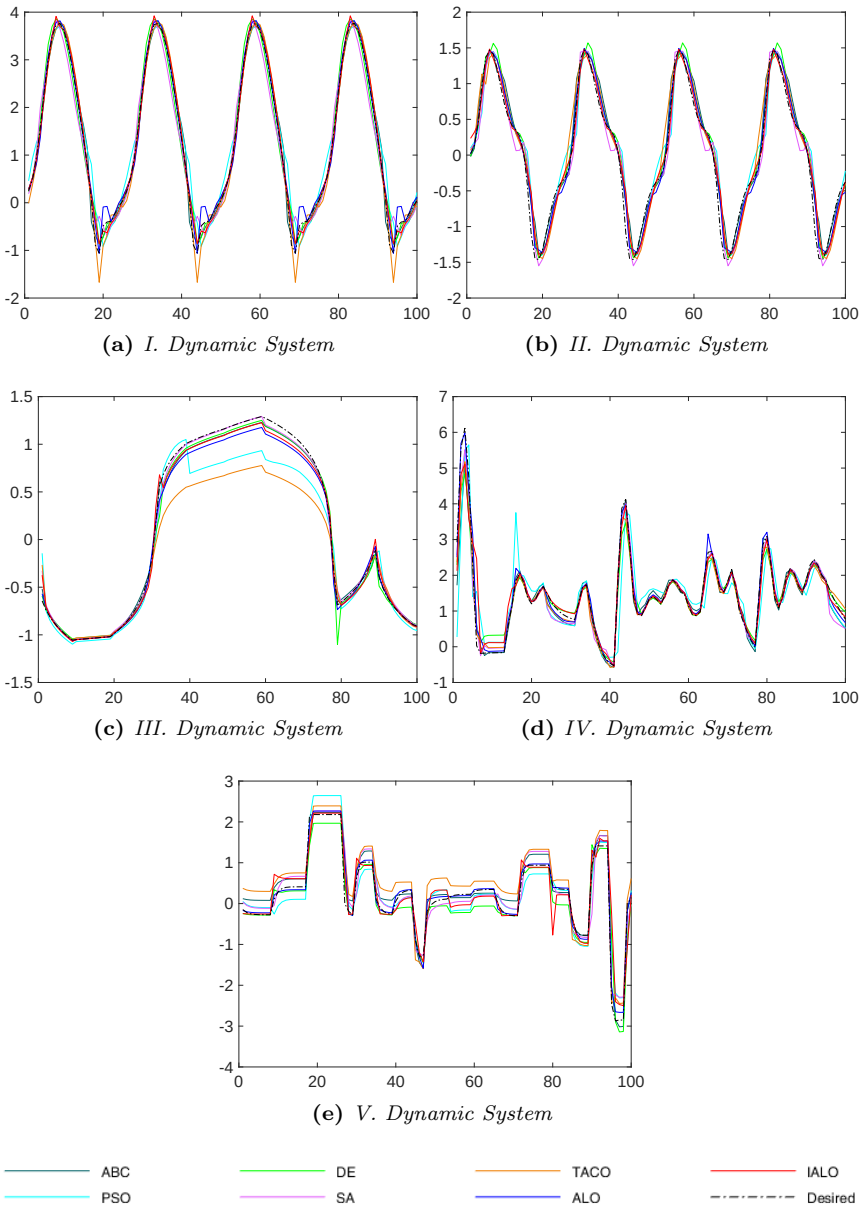


(a) *I. Dynamic System* **(b)** *II. Dynamic System*

(c) *III. Dynamic System* **(d)** *IV. Dynamic System*

(e) *V. Dynamic System*

Fig. 6 *Comparison performance of ANFIS models for training phase.*



(a) *I. Dynamic System* (b) *II. Dynamic System*

(c) *III. Dynamic System* (d) *IV. Dynamic System*

(e) *V. Dynamic System*

Fig. 7 *Comparison performance of ANFIS models for test phase.*

6. Conclusion

The most important lack of the ALO algorithm is that it has a long runtime because of the standard random walking modelling used in it. This paper proposed an improved ALO algorithm to overcome deficiencies in the original ALO algorithm. This proposed IALO algorithm includes some improvements on the random walking procedure, selection method and boundary checking mechanism. IALO algorithm was adapted for optimizing antecedent and consequent parameters of ANFIS model. To evaluate the performance of IALO algorithm, we used five dynamic systems from the literature. The ANFIS training and test results were compared with a variety of well-known heuristic algorithms using some statistical methods.

During ANFIS parameters optimization, IALO and other algorithms were run 30 times for generating meaningful statistical results. It is clearly evident that IALO algorithm is statistically better than DE, PSO, TACO, and SA algorithms according to the p-values in one-tailed Mann-Whitney test. The results of the ANFIS parameter optimization show that the IALO algorithm has competitive results among the well-known heuristic algorithms and is worth applying to different problems.

In the future studies regarding ALO, the opposition-based ALO, the chaotic system based ALO would be developed to increase the performance of the ALO algorithm more. Besides, the proposed various ALO algorithms would be implemented to different optimization problems, such as quadratic assignment problems, parallel scheduling problems, etc. The performance of the developed ALO algorithms would be compared with other heuristic algorithms.

Conflict of Interest

The authors declare that there is no conflict of interest regarding the publication of this article.

References

[1] BABUSKA R. *Fuzzy System, Modeling and Identification*. International Series in Intelligent Technologies. New York: Springer, 1998, doi: 10.1007/978-94-011-4868-9.

[2] DASGUPTA D. *Artificial immune systems and their applications*. Springer Science & Business Media, 2012.

[3] DERRAC J., GARCIA S., MOLINA D., HERRERA F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*. 2011, 1(1), pp. 3–18, doi: 10.1016/j.swevo.2011.02.002.

[4] GUPTA E., SAXENA A. Performance evaluation of antlion optimizer based regulator in automatic generation control of interconnected power system. *Journal of Engineering*. 2016, pp. 1–14, doi: 10.1155/2016/4570617.

[5] HANSEN N., ROS R., MAUNY N., SCHOENAUER M., AUGER A. Impacts of invariance in search: When cma-es and pso face ill-conditioned and non-separable problems. *Applied Soft Computing*. 2011, 11(8), pp. 5755–5769, doi: 10.1016/j.asoc.2011.03.001.

[6] HIROYASU T., MIKI M., ONO Y., MINAMI Y. Ant colony for continuous functions. *The Science and Engineering, Doshisha University*. 2000, 20.

[7] HLAVICA J., PRAUZEK M., PETEREK T., MUSILEK P. Assessment of Parkinson's disease progression using neural network and ANFIS models. *Neural Network World*. 2016, 26(2), pp. 111–128, doi: 10.14311/NNW.2016.26.006.

[8] JANG J.S.R. Anfis: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*. 1993, 23, pp. 665–685, doi: 10.1109/21.256541.

[9] JIANG H.M., KWONG C.K., IP W.H., WONG T.C. Modeling customer satisfaction for new product development using a pso-based anfis approach. *Applied Soft Computing*. 2012, 12(2), pp. 726–734, doi: 10.1016/j.asoc.2011.10.020.

[10] KAMBOJ V.K., BHADORIA A., BATH S.K. Solution of non-convex economic load dispatch problem for small-scale power systems using antlion optimizer. *Neural Computing and Applications*. 2017, 28(8), pp. 2181–2192, doi: 10.1007/s00521-015-2148-9.

[11] KARABOGA D., KAYA E. Training anfis using artificial bee colony algorithm. In: *Innovations in Intelligent Systems and Applications (INISTA)* (I. I. S. on IEEE, ed.), 2013, pp. 1–5.

[12] KARABOGA D., AKAY B. Artificial bee colony (abc) algorithm on training artificial neural networks. In: *IEEE 15th Signal Processing and Communications Applications*, 2007, pp. 1–4.

[13] KARAKUZU C. Parameter tuning of fuzzy sliding mode controller using particle swarm optimization *Int J Innovative Comput Inform Control*. 2010, 6, pp. 4755–4770.

[14] KARAKUZU C. On the performance of newsworthy meta-heuristic algorithms based on point of view fuzzy modelling. *Turkish Journal of Electrical Engineering & Computer Sciences*. 2017, 25, pp. 4706–4721, doi: 10.3906/elk-1705-337.

[15] KAYA S., GUNEY K., YILDIZ C., TURKMEN M. Anfis models for synthesis of open supported coplanar waveguides. *Neural Network World*. 2013, 23(6), pp. 553–569, doi: 10.14311/NNW.2013.23.033.

[16] KENNEDY J., EBERHART R. Particle swarm optimization. In: *Neural Networks, Proceedings., IEEE International Conference on*, 4, 1995, pp. 1942–1948.

[17] KILIC H., YUZGEC U. Improved antlion optimization algorithm via tournament selection and its application to parallel machine scheduling. *Computers & Industrial Engineering*. 2019, 132 pp. 166–186, doi: 10.1016/j.cie.2019.04.029.

[18] KILIC H., YUZGEC U. Tournament selection based antlion optimization algorithm for solving quadratic assignment problem. *Engineering Science and Technology, an International Journal*. 2019, 22(2), pp. 673–691, doi: 10.1016/j.jestch.2018.11.013.

[19] KILIC H., YUZGEC U., KARAKUZU C. A novel improved antlion optimizer algorithm and its comparative performance. *Neural Computing and Applications*. 2018, doi: 10.1007/s00521-018-3871-9.

[20] KUMAR V., VENKATESH K., TIWARI R. P. A neurofuzzy technique to predict seismic liquefaction potential of soils. *Neural Network World*. 2014, 24(3), pp. 249–266, doi: 10.14311/NNW.2014.24.015.

[21] MEI R.N.S. M., SULAIMAN M.H., ZURIANI M. Antlion optimizer for optimal reactive power dispatch solution. *Journal of Electrical Systems*. 2015, 3, pp. 68–74.

[22] MIRJALILI S. The antlion optimizer. *Advances in Engineering Software*. 2015, 83, pp. 80–98, doi: 10.1016/j.advengsoft.2015.01.010.

[23] NAIR S.S., RANA K.P.S., KUMAR V., CHAWLA A. Efficient modeling of linear discrete filters using antlion optimizer. *Circuits, Systems, and Signal Processing*. 2017, 36, pp. 1535–1568, doi: 10.1007/s00034-016-0370-z.

[24] NARENDRA K.S., PARTHASARATHY K. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*. 1990, 1(1), pp. 4–27, doi: 10.1109/72.80202.

[25] NISCHAL M.M.,MEHTA S. Optimal load dispatch using antlion optimization. *Int.J Eng Res Appl*. 2015, 5(8), pp. 10–19.

[26] OUSSAR Y., RIVALS I., DREYFUS L. Training wavelet networks for nonlinear dynamic input output modeling. *Neurocomputing*. 1998, 20, pp. 173–188, doi: 10.1016/S0925-2312(98)00010-1.

[27] PETROVIC M., PETRONIJEVIC J., MITIC M., VUKOVIC N.,PLEMIC A., MILJKOVIC Z.,BABIC B. The antlion optimization algorithm for flexible process planning. *Journal of Production Engineering*. 2015, 18(2), pp. 65–68.

[28] RAJU M., SAIKIA L., SINHA N. Automatic generation control of a multi-area system using antlion optimizer algorithm based pid plus second order derivative controller. *International Journal of Electrical Power and Energy Systems*. 2016, 80, pp. 52–63,doi: 10.1016/j.ijepes.2016.01.037.

[29] RAZALI N.M., GERAGHTY J., *et al.* Genetic algorithm performance with different selection strategies in solving tsp. In: *Proceedings of the world congress on engineering*, 2, 2011, pp. 1134–1139.

[30] RUTENBAR R.A. Simulated annealing algorithms: an overview. *in IEEE Circuits and Devices Magazine*. 1989, 5, pp. 19–26, doi: 10.1109/101.17235.

[31] SASTRY P.S., SANTHARAM G., UNNIKRISHNAN K.P. Memory neuron networks for identification and control of dynamical systems. *in IEEE Transactions on Neural Networks*. 1994, 5, pp. 306–319, doi: 10.1109/72.279193.

[32] SATHEESHKUMAR R., SHIVAKUMAR R. Antlion optimization approach for load frequency control of multi-area interconnected power systems. *Circuits and Systems*. 2016, 7(9), pp. 2357-2383.

[33] SIMSEK H., CEMEK B., ODABAS M.S., RAHMAN S. Estimation of nutrient concentrations in runoff from beef cattle feedlot using adaptive neuro-fuzzy inference systems. *Neural Network World*. 2015, 25(5), pp. 501–518, doi: 10.14311/NNW.2015.25.025.

[34] STORN R., PRICE K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*. 1997, 11, pp. 341–359, doi: 10.1023/A:1008202821328.

[35] TUNG N.S., CHAKRAVORTY S. Antlion optimizer based approach for optimal scheduling of thermal units for small scale electrical economic power dispatch problem. *International Journal of Grid and Distributed Computing*. 2016, 9(7), pp. 211–224, doi: 10.14257/ijgdc.2016.9.7.22.

[36] YAO P., WANG H. Dynamic adaptive antlion optimizer applied to route planning for unmanned aerial vehicle. *Soft Computing*. 2017, 21(18), pp. 5475–5488, doi: 10.1007/s00500-016-2138-6.

[37] YETILMEZSOY K., OZKAYA B., CAKMAKCI M. Artificial intelligence-based prediction models for environmental engineering. *Neural Network World*. 2011, 21(3), pp. 193–218, doi: 10.14311/NNW.2011.21.012.

[38] ZANGENEH A.Z., MANSOURI M., TESHNEHLAB M., SEDIGH A.K. Training anfis system with de algorithm. In: *Advanced Computational Intelligence (IWACI) Fourth International Workshop on IEEE*, 2011, pp. 308–314.

[39] Organic Garden Info Page, 2019 [viewed 2019-07-13]. Available from: http://www.organicgardeninfo.com/antlion.html