International Journal of Concrete Structures and Materials

RESEARCH

Open Access

A Hybrid Particle Swarm Optimization with Dragonfly for Adaptive ANFIS to Model the Corrosion Rate in Concrete Structures

Gholam Reza Khayati[1], Zahra Rajabi[1*], Maryam Ehteshamzadeh[1] and Hadi Beirami[1,2]

Abstract

The use of reinforced concrete is common in marine structures. Failure of reinforcement due to corrosion has detrimental impacts on nearly all of these structures. Hence, proposing an accurate and reliable model was imperative. The goal of this paper is to develop a new hybrid model by combining Particle Swarm Optimization (PSO) with Dragonfly Algorithm (DA) for Adaptive Neuro-Fuzzy Inference System (ANFIS) to predict the corrosion current density (C_{11}) of marine reinforced concrete. The neuro-fuzzy-based methods have emerged as suitable techniques for encountering uncertainties associated with the corrosion phenomenon in marine structures. To the best of our knowledge, this is the first research that predicts the C_{11} through a model integrating fuzzy learning, neural learning rules, and meta-heuristics. 2460 data are collected from 37 regions in Persian Gulf. The input parameters are age, concrete repairing history, height above the sea level, distance from sea, concrete compressive strength, rebar diameter, concrete cover depth, concrete electrical resistivity, chloride ion concentration and pH. The proposed rules for the estimation of C_{11} based on collected dataset are assessed based on the several metrics such as R^2, efficiency, mean absolute percentage error (MAPE), and median of absolute error (MEDAE). According to the results, ANFIS-PSO–DA enables to predict C_{11} by R^2 (0.92), MAPE (1.67), MEDAE (0.14), and EF (0.97). The results of sensitivity analysis revealed that concrete compressive strength and pH are the most effective parameters on the corrosion current density of reinforced concrete.

Keywords: modeling by improved ANFIS, corrosion current density, corrosion rate, marine reinforced concrete

1 Introduction

The most accepted reason for the destruction of reinforced concrete (RC) structures is the corrosion of the steel rebar inserted in them. Corrosion can be activated by the entrance of any carbon dioxide or chloride ions into the concrete structure (Qiao et al., 2015; Taffese & Sistonen, 2017). The CO_2 dissolves the protective layer on the rebar surface by a decrease in pH of surrounding concrete of reinforcement to below 9. The existence of chlorine ions above the critical concentration will also destabilize the protective film by attainment the surface of the steel rebars and occurrence of pitting corrosion (Doi et al., 2020; Paul et al., 2018; Rodrigues et al., 2020). These phenomena lead to the corrosion of the steel bar and consequently a reduction in its cross-sectional area, manifestation of cracks, and spalling in the concrete cover (Li et al., 2020). Thus, the mechanical properties of the rebar were reduced (Sun et al., 2018; Zhou et al., 2020) and this had a negative impact on the performance, safety, and durability of the reinforced concrete structure, which can cause the failure of the entire structure (Angst, 2018).

The total cost of corrosion damage in the world as reported by 2013 NACE impact study is about 4% of gross domestic product (GDP) (Salami et al., 2020). For example, in the case of a RC structure, the maintenance

*Correspondence: zr.rajabi95@gmail.com
[1] Department of Materials Science and Engineering, Shahid Bahonar University of Kerman, P. O. Box No, 76135-133 Kerman, Iran
Full list of author information is available at the end of the article
Journal information: ISSN 1976-0485 / eISSN 2234-1315

and repair cost of US highway bridges is about 8 billion US$ annually, while indirect costs are ten times higher due to the traffic delays and loss of efficiency (Hamidane et al., 2020). Therefore, predicting the service life and designing of the sustainable RC is a major challenge to avoid the cost of premature damage and extend the service life of such structures. In addition, this prediction can be utilized for maintenance and timely repair planning (Cai et al., 2020; Salami et al., 2020). Estimation and prediction rely gradually on analytical models based on in-service experience and observations. In general, the communication between the inferences attained from the experimental laboratory and the factual in-field behavior is weak (Melchers, 2020). In this study, we employed field data gathered from the corrosive area of the marine Persian Gulf to estimate the corrosion rate of steel reinforcement. Indeed, one of the significant factors that controls the destruction rate of RC structures is the corrosion rate.

In the recent decade, soft computing has become one of the most well-known intelligence techniques due to the lower cost and time consumption in different engineering sciences (Cai et al., 2019; Chung et al., 2018; Heo et al., 2018; Ryu et al., 2020; Shafaei and Khayati, 2020). Combining fuzzy logic with neural networks presents adaptive network-based fuzzy inference systems (ANFIS) to determine the superlative relationship among the input and output parameters in various processes. Fuzzy logic applies human reasoning in a mathematical framework and so can play a fundamental role in the relationship modeling of parameters. In ANFIS the training to find the adaptive parameters is very critical, which must be investigated. Therefore, artificial intelligence techniques and evolutionary methods are employed to determine the optimized parameters of the model.

Many researchers have utilized artificial intelligence in modeling of corrosion rate of steel rebar and service life prediction reinforced concrete structures (Anoop et al., 2002; Dey et al., 2019; Lv et al., 2020; Parthiban et al., 2005; Roxas and Lejano, 2019a; Sadowski and Nikoo, 2014; Salami et al., 2020; Taffese & Sistonen, 2017; Topçu et al., 2009). For example, Anoop et al. (2002) demonstrated the success of the fuzzy sets to predict the service life of reinforced concrete based on corrosion of reinforcement. Sadowski and Nikoo (2014) suggested a combination model based on ANN and imperialist competitive algorithm (ICA) to forecast the corrosion current density of steel in concrete. They employed ICA to optimize of ANN weights and compared ICA-ANN model with genetic algorithm. Their results revealed the high ability and accuracy of ICA-ANN model in forecasting of steel corrosion in comparison to the genetic algorithm. Roxas and Lejano (2019b) evaluated the performance of artificial neural network (ANN) in predicting the corrosion current density of steel rebar. Their results indicated a good prediction capability of neural network procedure. Lv et al. (2020) proposed a method based on particle swarm optimization support vector machine (PSO-SVM) and the grid search support vector machine (GS-SVM) for the prediction of steel corrosion rate of reinforcement. The authors found that the accuracy of PSO-SVM model is more than GS-SVM in the estimation of corrosion rate in steel.

Ansarinezhad and Shahbazian (2016) predicted corrosion rate of 3C steel by comparing hybrid ANFIS-PSO model with ANFIS-GA and SVR. Their results revealed that all the models have the capacity of predicting corrosion rate of steel. However, ANFIS-PSO model has a higher accuracy compared to ANFIS-GA and SVR. This is due to a combination advantage of both fuzzy logic with ANN and additionally benefit of inserting PSO algorithm.

To the best of our knowledge, different artificial intelligence approaches were used to predict the corrosion rate of steel. The main challenges of most literatures are the lack of appropriate operational data for modeling and the lack of use of appropriate strategies to distinguish effective interaction between selected features. The main contributions of current study can be considered as the usage of a sufficient field dataset and propose a new hybrid model based on the combination of PSO, dragonfly (DA) and ANFIS (PSO–DA–ANFIS) to model the corrosion rate of steel in concrete. In this regard, two main purposes are followed: (i) improvement of ANFIS performance for determining the corrosion current density as a criterion for corrosion rate using five optimization algorithms, namely, genetic algorithm (GA), particle swarm optimization (PSO), biogeography-based optimization (BBO), bat algorithm, hybrid particle swarm optimization (HPSO), improved genetic algorithm (IGA), hybrid dragonfly algorithm and particle swarm optimization (DA–PSO). (ii) Comparing the operation of available models, for predicting the corrosion current density in RC structures. The experimental results proved the ability of mentioned techniques in enhancing ANFIS accuracy for predicting the corrosion current density.

2 Background

Adaptive Neuro-Fuzzy Inference System (ANFIS) model typically consists of input–output variables, Fuzzy Inference System (FIS), which is used the Takagi–Sugeno rule types (Benzaouia & El Hajjaji, 2014; Lendek et al., 2010) and an Artificial Neural Network (ANN). In an ANFIS approach, adaptive and feed-forward network extracts fuzzy rules from inputs. Then, a hybrid learning procedure applies the parameters of fuzzy membership functions and tries to find correlations among the inputs and output according to the expert knowledge. Fig. 1

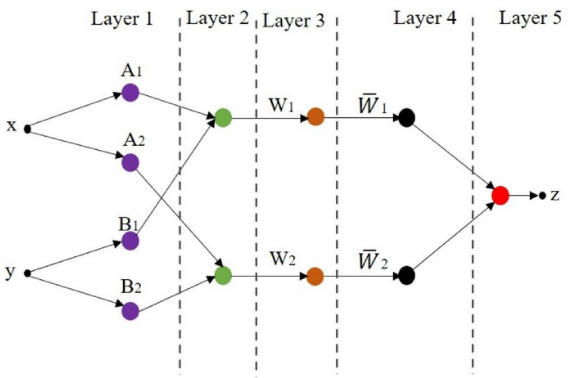


Fig. 1 ANFIS structure (Karaboga & Kaya, 2019a).

indicates the learning process of ANFIS with several nodes.

For example, in Fig. 1, two inputs (x, y) and one output (z) are assumed. Then, a set with two different if–then fuzzy rules for a first-order Sugeno fuzzy model is expressed by Eq. (1):

$$\begin{aligned} &\text{Rule1}: \text{if } x \text{ is } A1 \text{ and } y \text{ is } B1 \text{ then } Z1 = P1x + q1y + r1,\\ &\text{Rule2}: \text{if } x \text{ is } A2 \text{ and } y \text{ is } B2 \text{ then } Z2 = P2x + q2y + r2. \end{aligned} \tag{1}$$

The linguistic $A1$ and $B1$ variables perform the entries evaluation and a linear combination of the inputs with a constant value (r) defines the results of each rule (Karaboga & Kaya, 2019b).

2.1 Particle Swarm Optimization (PSO)

Generally, PSO is a meta-heuristic technique that is inspired by the movement of flocks and used for solving different optimization problems and with low calculations and acceptable results. Compared to existing optimization algorithms, PSO has a high learning speed, takes up less memory and is easy to execute (Bahiraei et al., 2021; Patwal et al., 2018). This algorithm is a population-based search technique in which every capability solution is signified as a particle in a populace (Pousinho et al., 2012). Each particle is considered as a solution and randomly moves in the search space. In PSO, particles move toward their best neighbors. In other words, each particle updates its position based on its best position and the best position adjacent to its neighbor. PSO randomly generates a particle set. If $x_i^t = \left(x_{i1}^t, x_{i2}^t, \ldots x_{in}^t\right)$ and $v_i^t = \left(v_{i1}^t, v_{i2}^t, \ldots v_{in}^t\right)$ show the position and velocity of the ith particle for tth iteration, respectively, then the particle updates its position for the $t+1$ iteration based on Eqs. (2) and (3) (Alarifi et al., 2019):

$$v_i^{t+1} = \omega \cdot v_i^t + c_1 \cdot r_1(p_i^t - x_i^t) + c_2 \cdot r_2 \cdot (g_i^t - x_i^t) \text{ with } -v_{\max} \le v_i^{t+1} \le v_{\max}, \tag{2}$$

$$x_i^{t+1} = (x_i^t + v_i^{t+1}), \tag{3}$$

where x_i^t shows the previous position of ith particle. p_i^t and g_i^t indicate the best position of ith particle and the best position found by particles, respectively. The two parameters r_1 and r_2 are random numbers in [0–1]. In addition, the cognitive coefficient, social coefficient, and inertia weight are defined by c_1, c_2, and ω, respectively. In standard PSO, the inertia weight is obtained as Eq. 4 (Basser et al., 2015):

$$\text{Interia weight } \omega = \omega_{\max} - \left(\frac{\omega_{\max} - \omega_{\min}}{\text{Itr}_{\max}}\right) \times \text{Itr}, \tag{4}$$

where $\omega_{\max}$ and $\omega_{\min}$ signify the initial and final inertia weights, and also $\text{Itr}_{\max}$ and $\text{Itr}_{\min}$ are the maximum number of iteration and the current number of iteration.

2.2 Genetic Algorithm (GA)

GA is a heuristic optimization method based on Darwinian survival of the fittest and natural selection. It is inspired by nature and biological evolution. The algorithm begins by random initializing a population of individual solutions (chromosomes). Then GA operates on this population to produce better and better generation of it. The simple genetic algorithm follows stages: initialize population; calculate fitness, selection, crossover and mutation (Mirjalili, 2019).

It uses mutation and crossover operators for recombination and generating a population that has more properties related to the previous iteration. The fitness function is used for chromosome evaluation. Equation 5 indicates crossover formula (Katoch et al., 2021):

$$R = (G + 2\sqrt{g})/3G, \tag{5}$$

where G represents the total number of evolutionary generation set by populace and g indicates the number of generations.

3 Problem Explanation, Material, and Methods

The data are collected from electric transmission tower foundations of 37 regions in the Persian Gulf. These foundations have a great impact on the stability and performance of the towers. Various examples of corrosion affected zones of these structures are shown in Fig. 2a–c. As illustrated the corrosion of steel rebar causes the concrete cover cracks.

For investigating the corrosion behavior, different parameters were measured as construction date or age

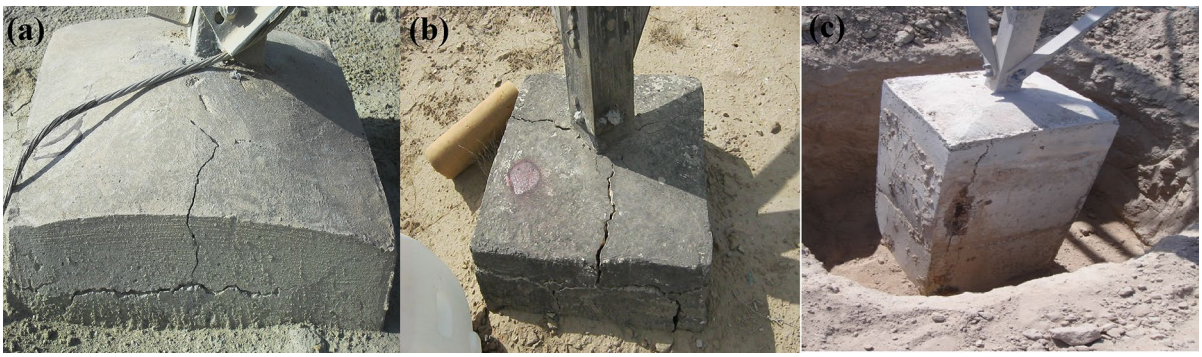


Fig. 2 Corrosion-affected zones of electric transmission tower foundations.

Fig. 3 The measurement of corrosion rate by galvanostatic pulse technique (Beirami & Ehteshamzadeh, 2016).

of concrete (C_1, year), concrete repairing history (C_2; 0: NO, 1: yes), height above the sea level (C_3, m), distance from the sea (C_4, km), concrete compressive strength (C_5, N/mm^2), rebar diameter (C_6, mm), concrete cover depth (C_7, mm), concrete electrical resistivity (C_8, Ω m), chloride ion concentration (C_9, ppm) in concrete, alkalinity of concrete (C_{10}) and corrosion current density (C_{11}, μA/cm^2) of the rebar. The C_6 and C_7 were evaluated with an ultrasonic test. EN 13791 and ASTM C114 standards were used to determine the C_5 and C_9, respectively. The corrosion rate or C_{11} and C_8 were measured by galvanostatic pulse technique (Fig. 3).

ANFIS model adjusts two structural parameters (i.e., antecedent and consequent) based on the gradient-based methods. The main weakness of gradient-based methods is the slow convergence rate and the solution may trap local optimality (Salleh & Hussain, 2016). Metaheuristic optimization techniques (e.g., PSO and GA) can be used to overcome the issue of gradient-based methods. The optimal neuro-fuzzy system steps are defined as follows:

1. Define the train and test data;
2. Design a fuzzy system;
3. Tune the parameters of the fuzzy system based on meta-heuristic technique and error function;
4. Determine the best values of the ANFIS model.

3.1 PSO and GA Improvement Methods

PSO and GA are popular due to their simplicity and strong global search capability and the optimal answer is obtained in a relatively small number of iterations. Similar to most of the optimization techniques, GA and PSO algorithms can fall in the local optima. Especially for complex problem, the immature convergence of these algorithms have been observed. Thus, several modifications of PSO and GA are developed to enhance the ability to find global optimal value and jump out of local optimal solutions (Garg, 2015, 2018; Patwal et al., 2018; Serani et al., 2015).

An effective way is a hybridization with other meta-heuristics. For example, Garg (2016) presented a PSO–GA method where at first particles are updated based on standard PSO, and then some of them are selected for performing the crossover and mutation to increase the diversity of the population. Nevertheless, as the dimensionality of the problem increases, the likelihood of being trapped in the local optimization of the PSO–GA method increases. To solve this issue, in this paper a chaotic function is used in the searching step of particles. The details of the proposed HPSO are explained in Subsect. 3.1.1. Another hybridization version of PSO is introduced by Rafie et al. (Pellegrini et al., 2020) that uses the ability of the dragonfly algorithm to improve the local search capability.

Many works have been done to improve the searchability and diversity for genetic algorithms by modifying the genetic operators (selection, crossover, and mutation) (Metawa et al., 2017). For example, Song & Kusiak (2010) introduced a new ranking group selection operator that helps GA to generate a more feasible solution. In

addition, the standard GA algorithm randomly initializes chromosome swarm and may fall into the local optimum during searching. To address this weakness, a chaotic initialization method is used.

3.1.1 Hybrid Particle Swarm Optimization (HPSO)

It can be seen that the inertia weight ω in Eq. (4) affects the convergence speed. As iterations number increases, ω decreases and leads to a decrease in the accuracy and speed of convergence in PSO. For setting the search step of particles so that they can better explore search space, a chaotic version of inertia weight is used. We consider the sin map (Griffin, 2013) function. The sine interval is defined between zero and one. The inertia weight ω is obtained based on Eq. (6) (Wang et al., 2020):

$$\omega = \overline{\omega}^{t} = A \cdot \sin(\pi \overline{\omega}^{t-1}),\ \overline{\omega}^{t} \in (0,1)\ , 0 < A \leq 1,\ k = 1,2,\ldots,\max \text{iter}, \tag{6}$$

where k is the current iteration number.

After updating particles using Eqs. (2), (3), and (6), some particles are randomly selected to perform crossover (Prügel-Bennetf, 2001) and mutation (Hassanat et al., 2019) by the roulette wheel selection (Yu & Author, 2016). New populations are called *Cpopulation* and *Mpopulation*. The particles in *Cpopulation* and *Mpopulation* are evaluated, and then the appropriate particles are selected for the next iteration.

3.1.2 Hybrid Particle Swarm Optimization with Dragonfly (PSO–DA)

The hybrid algorithm, named PSO–DA, uses the dragonfly algorithm (DA) and PSO to solve the optimization problem. In general, PSO has good exploitation due to the best experience of the particles and so shows a quick converge rate. Nevertheless, PSO may fall in the local optima and show a quickly converge exploration ability. While DA improves the exploration ability and stochastic behavior using the Levy flight. Therefore, a hybrid algorithm (PSO–DA) combines the exploration of DA and the exploitation of PSO. In the first phase, DA defines the population of dragonflies to determine the region of the global solution. Then, the best position of DA is replaced with the global best position of PSO. Now, PSO performs the exploitation phase to determine the expected optimal solution. Therefore, the velocity of PSO is defined by Eq. (7) (Khunkitti et al., 2018):

$$\begin{aligned} v_i^{t+1} &= \omega \cdot v_i^t + c_1 \cdot r_1 (x_{p\text{best}_i}^t - x_i^t) \\ &\quad + c_2 \cdot r_2 \cdot (x_{\text{DA}}^{t+1} - x_i^t) \text{ with} \\ &\quad - v_{\max} \leq v_i^{t+1} \leq v_{\max}, \end{aligned} \tag{7}$$

where x_{DA}^{t+1} shows the best position of DA for iteration $t+1$.

3.1.3 Improved Genetic Algorithm (IGA)

In this modification of GA, a chaotic system [e.g., logistic map (Liu et al., 2013)] has been employed to generate the initial population. In other words, IGA replaces the standard initialization (i.e., uniformly distributed random numbers) with the chaotic system that shows more randomness and less regular behavior. In the chaotic initial population, each individual is defined based on Eq. (8) (Wang et al., 2020):

$$c_{i+1} = \mu c_i (1 - c_i),\ c_i \in [0,1],\ i = 0,1,\ldots,D-1 \tag{8}$$

where c_i indicates a set of chaotic numbers, μ shows the control factor, D denotes the dimension of the problem, and $c_0 \notin \{0, 0.25, 0.5, 0.75, 1\}$. Firstly, it defines a D-dimensional chaotic vector $c = (c_0, c_1, \ldots, c_{D-1})$, where c_0 is obtained by random manner, and $c_1, \ldots, c_{D-1}$ are created according to Eq. (8).

Secondly, it maps the chaotic vector (c) to search space $[\text{lb}, \text{ub}]^D$ based on Eq. (9) (Tang et al., 2010):

$$x_i = \text{lb}_i + c_i(\text{ub}_i - \text{lb}_i),\ i = 0,1,\ldots,\ D-1, \tag{9}$$

where lb_i and ub_i are the upper and lower bounds for the ith dimension, respectively. Therefore, an individual $x = (x_0, x_1, \ldots, x_{D-1})$ is generated.

3.1.4 Modified Genetic Algorithm (MGA)

In the GA, the selection process has a great impact on the overall performance of the genetic algorithm. Since the mutation and crossover operators should choose one or more agents and the selection operator determines which agent can be selected as input to these two operators. Instead of using common selection methods where they require repeated comparison of the individual's fitness, an alternative approach named ranking group selection (RGS) is employed (Rahimzadeh et al., 2021). RGS is as follows.

The size of the population is considered a multiple of 4 and the individuals are arranged based on fitness function *F(x)*. For maximum and minimum value problems, descending and ascending order of *F(x)* is necessary, respectively. Assume that $X = \{X_1, X_2, \ldots, X_n\}$ is the initial population and the arranged population is $X^R = \{X_1^R, X_2^R, \ldots, X_n^R\}$, hence it satisfies $F(X_1^R) \leq F(X_2^R) \leq, \ldots, \leq F(X_n^R)$. X^R is divided into four sets

(i.e., $X^1 = \left\{X^R_1, \ldots, X^R_{n/4}\right\}$, $X^2 = \left\{X^R_{(n/4)+1}, \ldots, X^R_{n/2}\right\}$, $X^3 = \left\{X^R_{(n/2)+1}, \ldots, X^R_{3n/4}\right\}$, and $X^4 = \left\{X^R_{(3n/4)+1}, \ldots, X^R_n\right\}$) to enhance the diversity of the paired individuals. When two elements must be selected from four elements, there are six cases [i.e., (X^1, X^2), (X^1, X^3), (X^1, X^4), (X^2, X^3), (X^2, X^4), and (X^3, X^4)] is feasible. Now, RGS presents the paired groups of individuals as $X^A = \{X^1, X^1, X^1, X^2, X^2, X^3\}$and $X^B = \{X^2, X^3, X^4, X^3, X^4, X^4\}$. Fig. 4 indicates the RGS process when the population size is 8.

In Fig. 4, the population is first sorted and divided into four sets (i.e., X^1, X^2, X^3, and X^4). Then, the crossover is performed by selecting X^R_1 in X^A and X^R_3 in X^B, X^R_2 in X^A and X^R_4 in X^B, and so on. The individuals of X^A are better than the individuals of X^B in terms of fitness value.

3.2 Optimize ANFIS with PSO and GA Variants

In Fig. 1, ANFIS with two layers consists of adaptive parameters and other layers have fixed parameters. The adaptive parameters can be grouped as consequent and premise parameters. In the first layer, there are premise parameters that correlate to the membership functions like the Gaussian function (Eq. 10) (Sarkheyli et al., 2015):

$$\mu_i(x) = e^{\frac{-(x-c_i)^2}{2a_i^2}}. \tag{10}$$

In Eq. (10), a_i and c_i show the premise parameters.

In this layer, the membership functions number is a function of linguistic terms number of inputs. For Gaussian function, the total number of premise parameters that must be optimized is obtained by Eq. (11) (Sarkheyli et al., 2015):

$$\mathrm{TNP} = 2 \times \sum_{i=1}^{n} L_i, \tag{11}$$

where n indicates the number of inputs and L_i shows the linguistic terms for input i. In the fourth layer, the consequent parameters of the fuzzy rules are determined. In Eq. (12), the total number of consequent parameters is determined (Sarkheyli et al., 2015):

$$\mathrm{TNC} = r \times (n+1). \tag{12}$$

As can be seen, the effective and optimized finding of the premise parameters of layers 1 and 4 plays a key role in finding the best ANFIS model.

3.2.1 *Initialization*

Firstly, the proposed algorithm creates a basic fuzzy system according to the received training data. Secondly, the optimization algorithms determine the best values based on predefined objective functions, which is more explained in details in Subsect. 3.2.2. The general frameworks of the optimized ANFIS by PSO-based and GA-based variants are illustrated in Figs. 5 and 6.

Fig. 5 shows the steps of PSO–DA and HPSO that have similar steps like initialization particles and loading dataset. But PSO–DA is a hybrid method where two methods particle swarm optimization and dragonfly are combined. In addition, HPSO adds two operators (i.e., crossover and mutation).

Fig. 6 indicates the steps of MGA and IGA. The difference between them can be seen in the initialization step, where IGA initializes its individuals with a chaotic

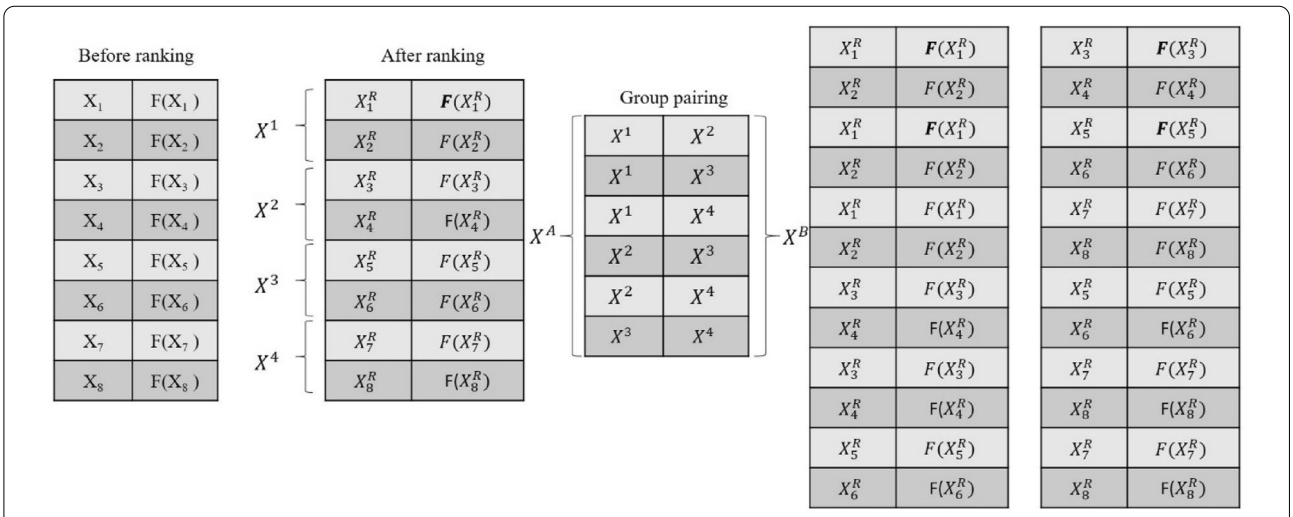


Fig. 4 An example of the RGS process.

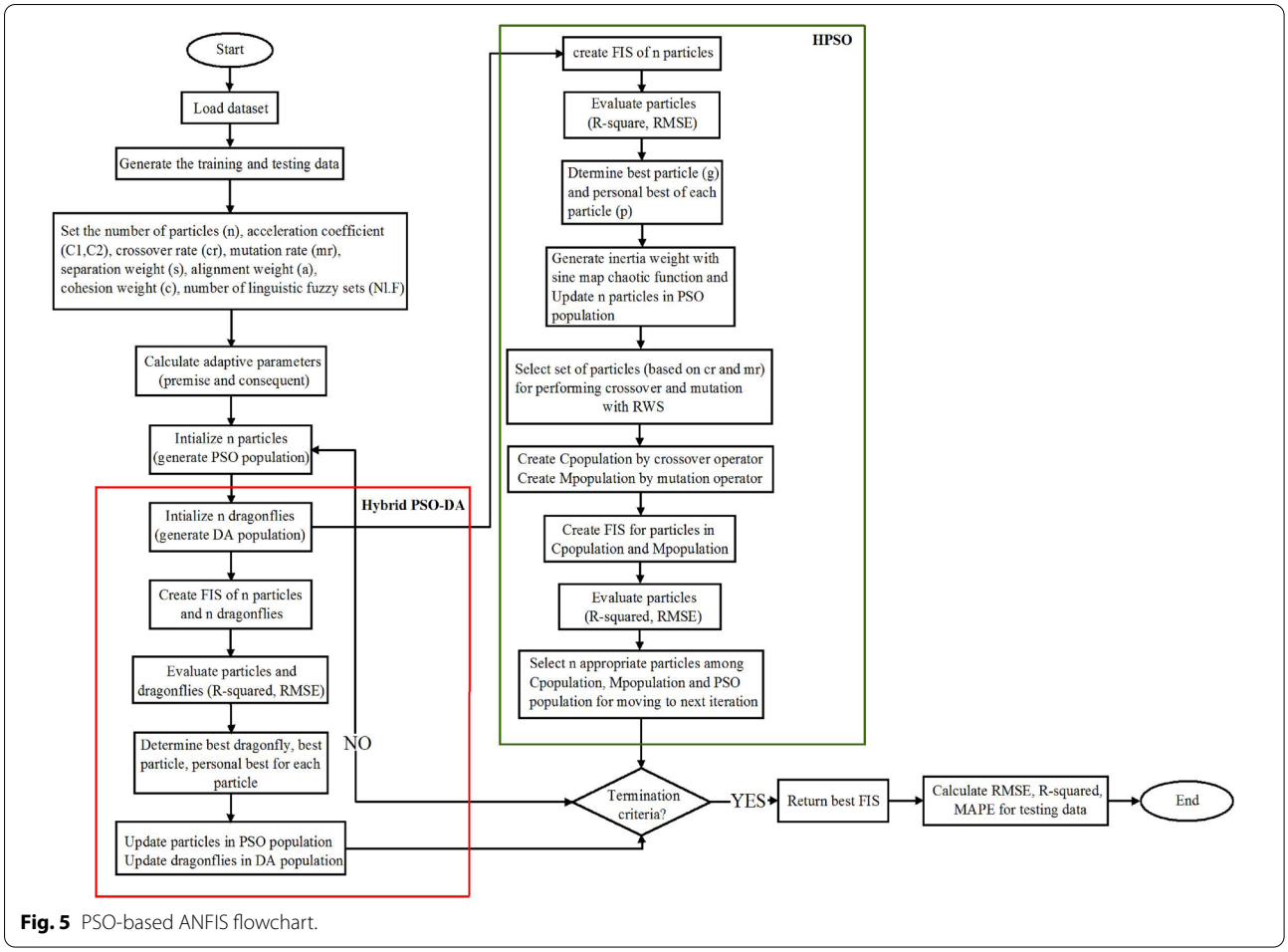


Fig. 5 PSO-based ANFIS flowchart.

function and MGA considers uniformly distributed random numbers. In the following steps, two algorithms perform crossover and mutation like standard GA. But MGA selects individuals by RGS technique (see Sect. 3.1.4) and IGA selects them by roulette wheel selection.

3.2.2 Individual Evaluation

It evaluates the position of each individual based on the objective function *F*. In this paper, two factors are used for evaluating each position. Firstly, the square of coefficient of determination (R^2) determines the percentage change in response due to the input variable *X*. Secondly, the root mean squared error (RMSE) shows the size of the errors in the regression model. Here, the objective function maximize R^2 and minimize the RMSE value, respectively. R^2 and RMSE are defined by Eqs. 13 and 14, respectively (Somu et al., 2020):

$$R^2 = 1 - \frac{\text{SSE}}{\text{SST}} = 1 - \frac{\sum_{i=1}^{n}(y_i - \widehat{y}_i)}{\sum_{i=1}^{n}(y_i - \overline{y}_i)}, \tag{13}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{y}_i\right)}, \tag{14}$$

where y_i indicates the observed amount and $\widehat{y}_i$ shows the predicted amount for the *i*th case. In addition, $\overline{y}_i$ is the average of observed values. The adopted fitness function combines the two criteria into one value by setting a weight factor in Eq. (15) (Somu et al., 2020):

$$F = \text{minimize}(\text{RMSE} + w \times (1 - R^2)), \tag{15}$$

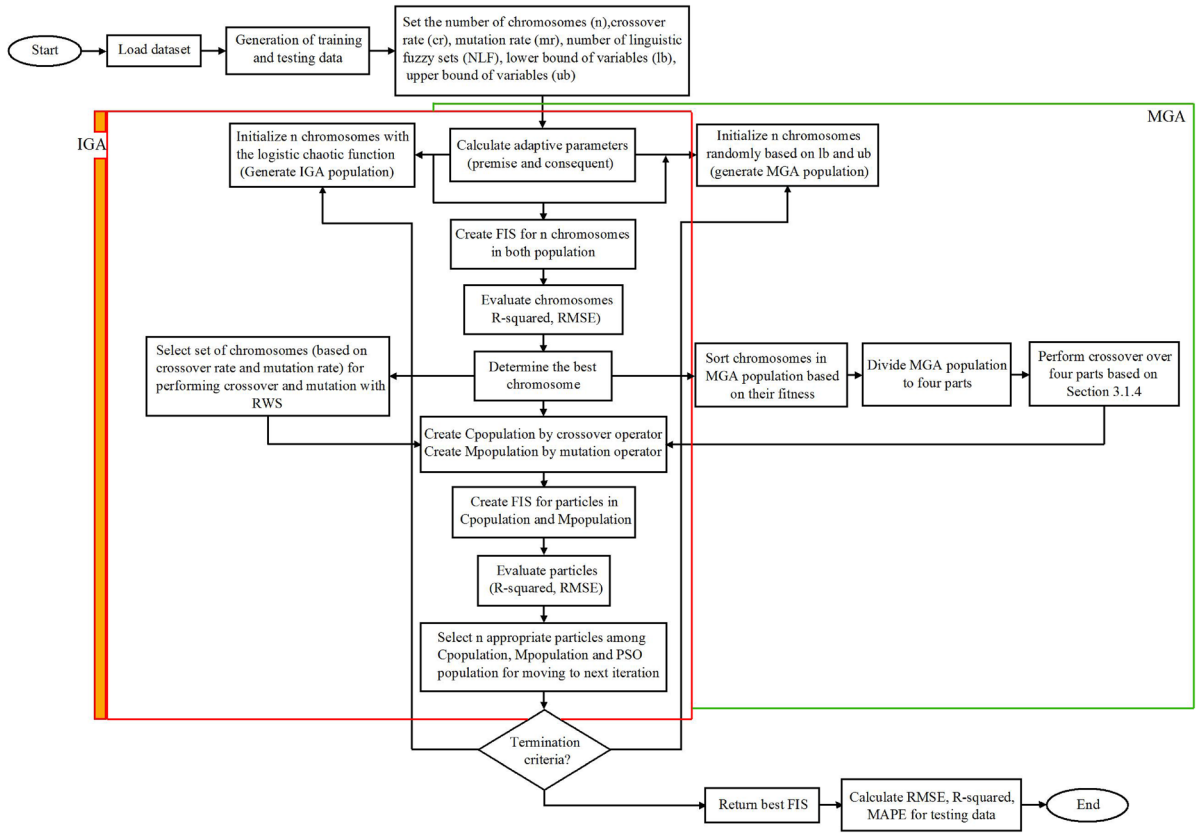


Fig. 6 GA-based ANFIS.

where w shows the weight factor, which has value in [0, 1]. The factor w is used to control the importance of the coefficient of determination and the size of errors. We divide the original dataset into two parts (i.e., training and testing) based on a tenfold cross-validation approach. The training part is applied to train a model based on the optimization process and the testing part evaluates the selected features.

4 Discussion of Results

This section is divided into two parts. Firstly, the performance of GA and PSO variants, which are explained in Sect. 3, is tested based on several test functions. In addition, the optimization results obtained by GA, PSO, and their variants including (i.e., HPSO, PSODA, IGA, and MGA) and other strategies (i.e., BBO and Bat) are discussed. Secondly, the ANFIS created by each algorithm is evaluated. Moreover, the sensitivity analysis is performed for the appropriate algorithm, which indicates in Sects. 4.1 and 4.2.

Table 1 Parameter setting.

Methods	Parameters	Values
GA, IGA	Mutation rate	0.33
	Crossover rate	0.67
PSO, HPSO, PSODA	Inertia weight	2
	Best global experience	2.2
	Best personal experience	2.4
	w-damp	0.98
Bat	Minimal and maximal values of the acoustic frequency ($[F_{\max}, F_{\min}]$)	[1,− 1]
	Pulse intensity attenuation coefficient (γ)	0.95
	Pulse frequency increase factor (δ)	0.05
	Maximum pulse frequency (R^0)	0.75
	Maximum pulse loudness (A)	0.25
BBO	Habitat modification probability	1
	Immigration probability limits	[0,1]
	Step size	1
	Max immigration (I) and Max emigration (E)	1
	Mutation probability	0.005

The setting of parameters of different methods in this paper is shown in Table 1 (Guo et al., 2020; Hayyolalam et al., 2020; Hong et al., 2019).

4.1 General Experimental Setting

1. Test problems and dimension setting: To verify the performance of compared algorithms, ten test functions from CEC2005 (Liang et al., 2013) benchmark problems are used. According to their diverse characteristics, the problems can be divided into two kinds of optimization problems (Liang et al., 2013):

   - Unimodal problems (i.e., F1–F5).
   - Multimodal problems (**benchmark**).

   Table 2 shows the details of the test functions used.

2. Experimental platform: All the experiments are run on a PC with intel core i5-10210U 2.11 GHz CPU and 8 GB memory.
3. Performance metrics: In this paper, convergence curve, trajectory, and distance to the best solution (Δ_x) are used to investigate the performance of each algorithm. The convergence curve shows the best solution obtained by each algorithm throughout the iteration. Trajectory illustrates the location of the first agent at the first dimension in each algorithm. It is used to show whether the agent changes its location to explore the search space or exploit a specific location. The distance-based metric is used to show the algorithm-found solutions and analytical optima. It is calculated by Eq. (16) (Serani et al., 2016):

$$\Delta_x = \sqrt{\frac{1}{N_{dv}} \sum_{j=1}^{N_{dv}} \left(\frac{x_{j,\min} - x^*_{j,\min}}{R_j} \right)^2}, \tag{16}$$

where Δ_x is a normalized Euclidean distance between the analytical optima ($x^*_{\min}$) and the optimal position that is obtained by the method ($x_{\min}$). $R_j = |u_i - l_j|$ shows the range of variable j.

4.1.1 Analysis of Compared Methods on Benchmark Functions

In this subsection, the behavior of different methods for several test functions has been examined. Figs. 7 and 8 show the trajectory and convergence curve of different methods, respectively.

From Fig. 7, it can be seen that most methods face abrupt fluctuations in less than 50% of the time (under 250 iterations) for Unimodal functions (i.e., F1–F5). This behavior is logical because Unimodal functions have one

Table 2 Test functions description.

Function	Dim	Range	$f_{\min}$
$F_1(x) = \sum_{i=1}^{n} x_i^2$	10	[− 100,100]	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	10	[− 10,10]	0
$F_3(x) = \sum_{i=1}^{n} (\sum_{j-1}^{i} x_j)^2$	10	[− 100,100]	0
$F_4(x) = \max_i \{\lvert x_i \rvert, \quad 1 \le i \le n\}$	10	[− 100,100]	0
$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2) + (x_i - 1)^2]$	10	[− 30,30]	0
$F_6(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$	10	[− 600,600]	0
$F_7(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6} \right)^{-1}$	2	[− 65.53,65.53]	1
$F_8(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1 (b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[− 5,5]	0.0003
$F_9(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[− 5,5]	− 1.0316
$F_{10}(x) = \left[1 + (x_1 + x_2 + 1)^2 \left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2 \right) \right] \times \left[30 + (2x_1 - 3x_2)^2 \times \left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2 \right) \right]$	2	[− 2,2]	3

global optimal. In multimodal functions, this fluctuation is increased up to half of the time since the number of local optimal increases in these types of functions and hence subsequently the exploration activity is increased too.

We can observe that, unlike most algorithms where their fluctuations are gradually decreased throughout iterations, HPSO and Bat do not converge appropriately. The sine-map function that is used in HPSO and the frequency parameter that is used in the Bat algorithm for setting the value of weight can change the size of the search step irregularly. Therefore, agents take a long step in some iterations and their locations are very differently changed from their previous location. Then,

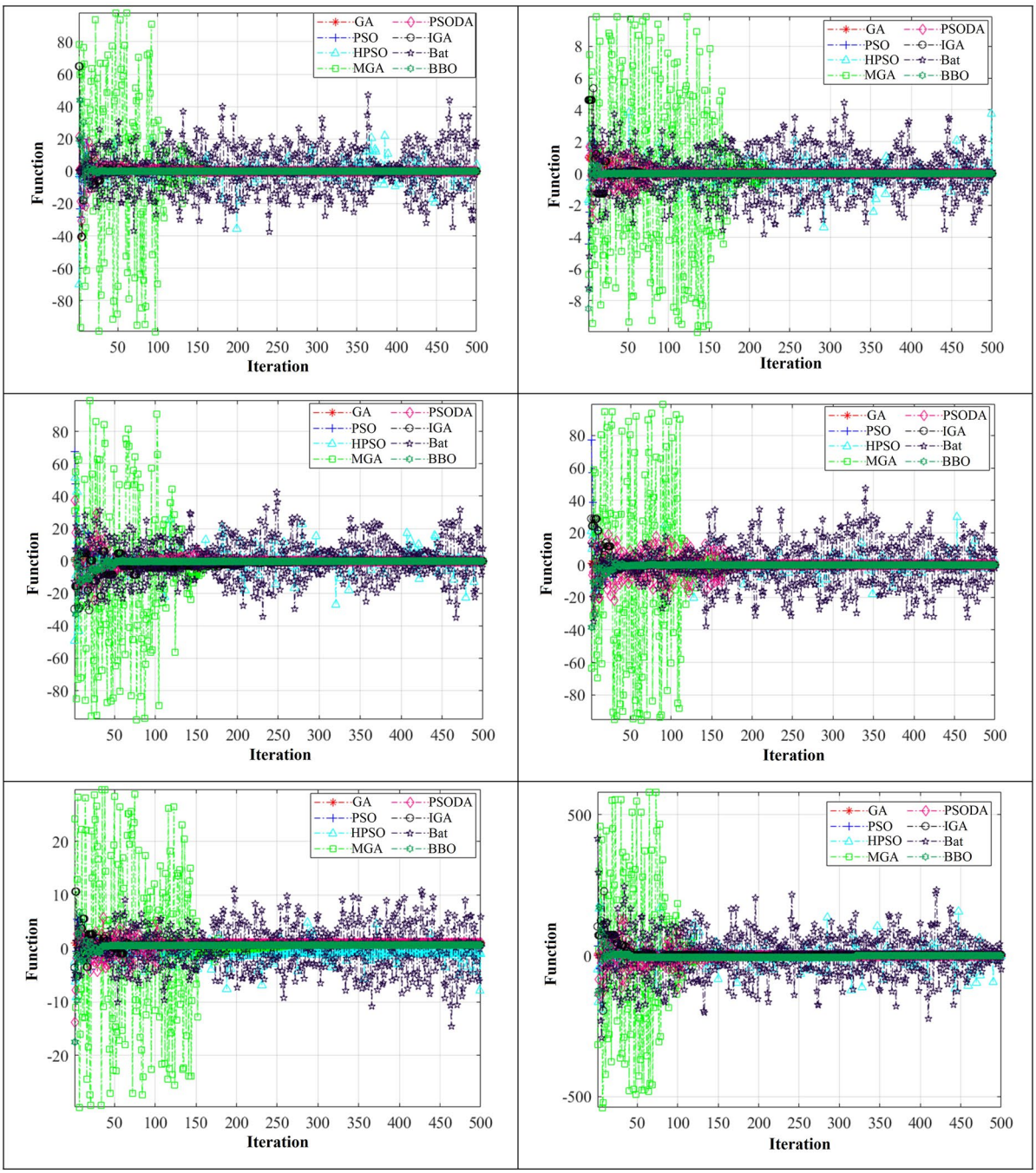


Fig. 7 The trajectory of different compared methods.

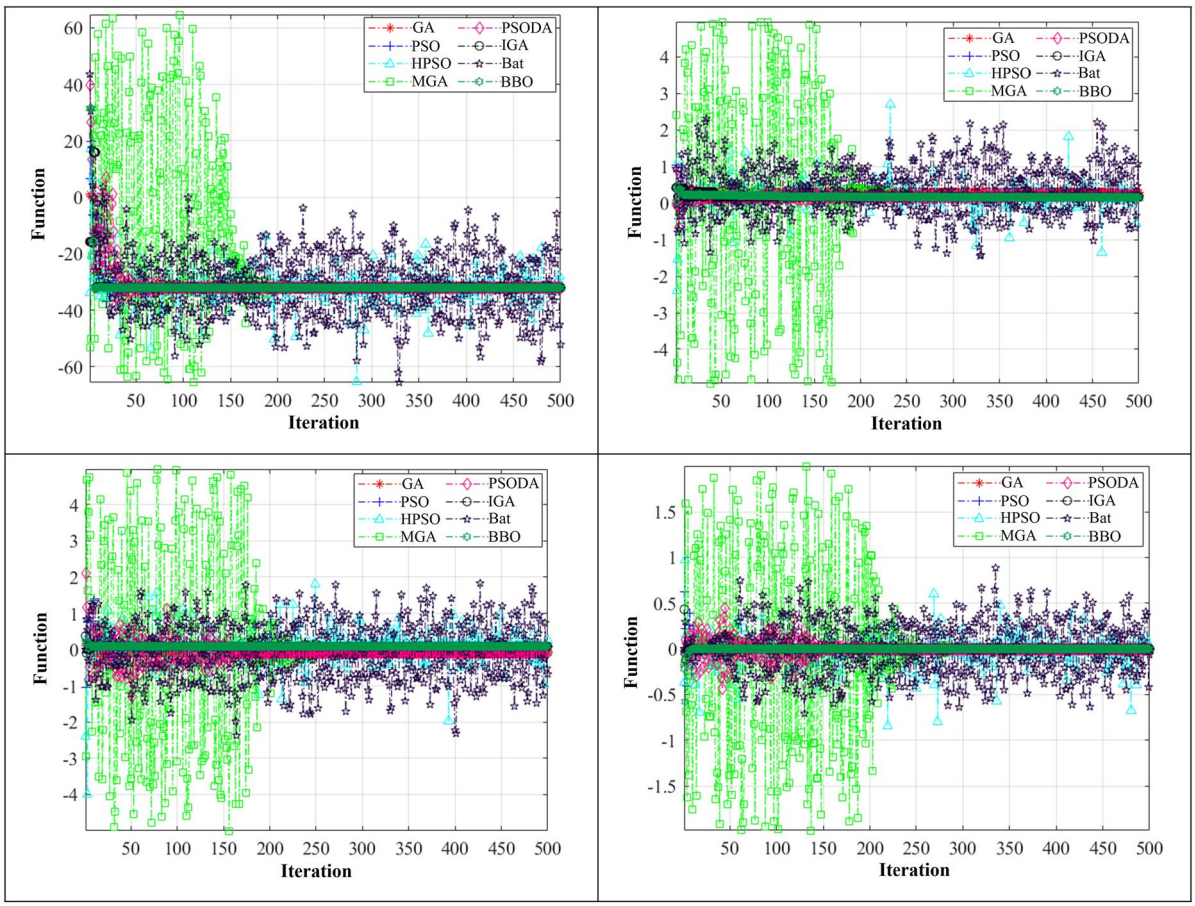


Fig. 7 continued

they take a small step search to move near from current location with small change.

However, using a chaotic function increases the exploration ability of HPSO but causes the HPSO algorithm to face unbalancing between exploration and exploitation activity. To remove this drawback, evolutionary operators (e.g., crossover and mutation) are added but it cannot help HPSO to converge appropriately.

In Fig. 7, it seems that PSODA and MGA are two algorithms that have successfully provided a balance between exploration and exploitation. We can see that range of PSODA fluctuations are smaller than MGA. It is because the best solution is determined by DA in each iteration of PSODA. The DA has a good exploration ability and when it is combined with PSO, it can significantly improve the searchability. In general, it seems that PSODA and MGA are the best ones among other variants and can converge to the global optimal and escape from the local optimal.

Fig. 8 shows the best value obtained by each algorithm for different test functions. It can be seen that PSODA reaches the best value in Unimodal functions except in F1, which achieves the second-best result.

Fig. 9 indicates the average fitness of different methods. It shows the performance of agents in a method as a team unlike the convergence curve (Fig. 8), where it considers only the best agent in each iteration and plots its behavior. In general, it is expected that all agents eventually behave like the best agent and have a downward movement from the first iteration to the last one. It can be seen from Fig. 9 that HPSO agents don't follow this term in functions F4 and F7, while the behavior of the best agent in Fig. 8 shows the downward movement. On the other hand, in the both of Figs. 8 and 9, the formation of BBO has a downward movement and all agents try to help each other to reach the best position. From the structure of HPSO and BBO and their results in Figs. 8 and 9, it can be concluded that if all agents in HPSO get an

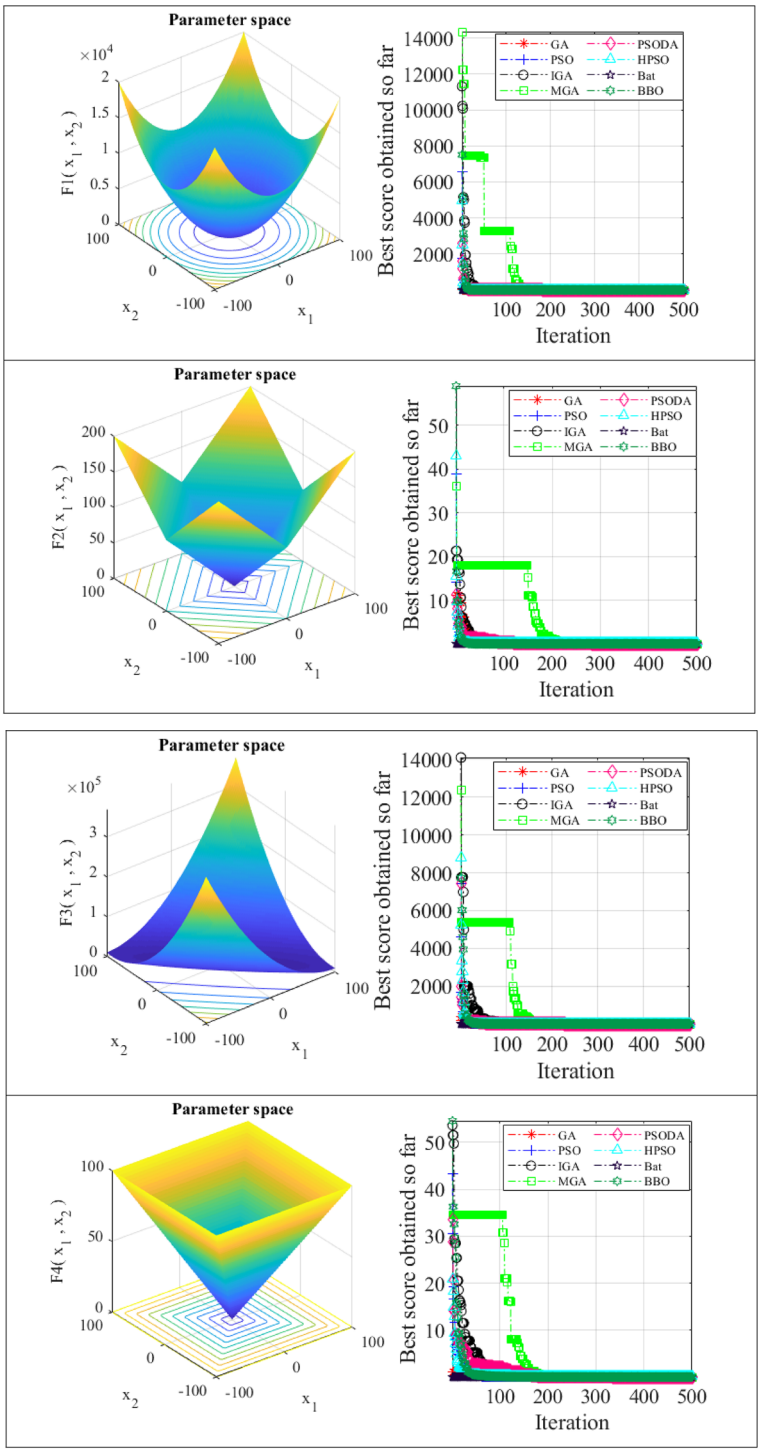


Fig. 8 Convergence curve of different compared methods.

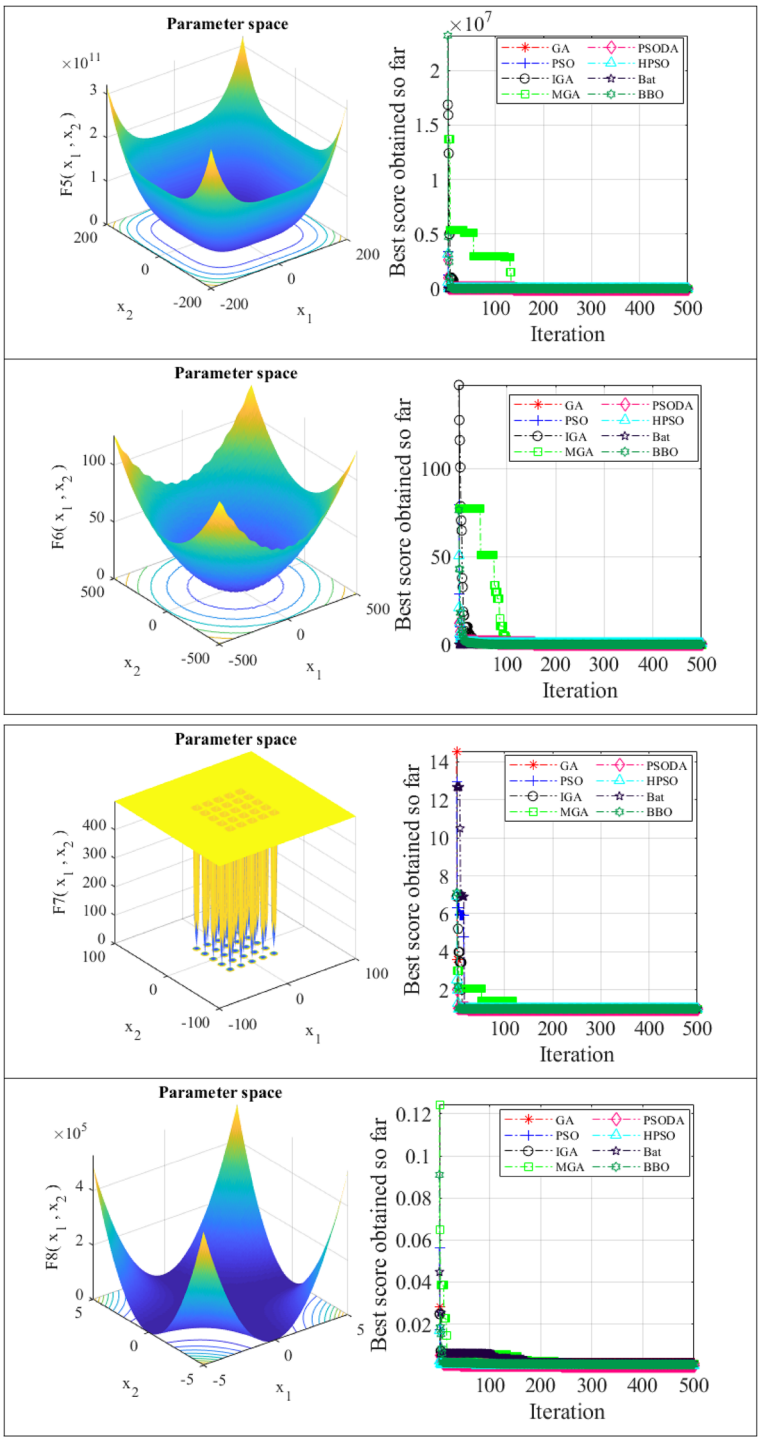


Fig. 8 continued

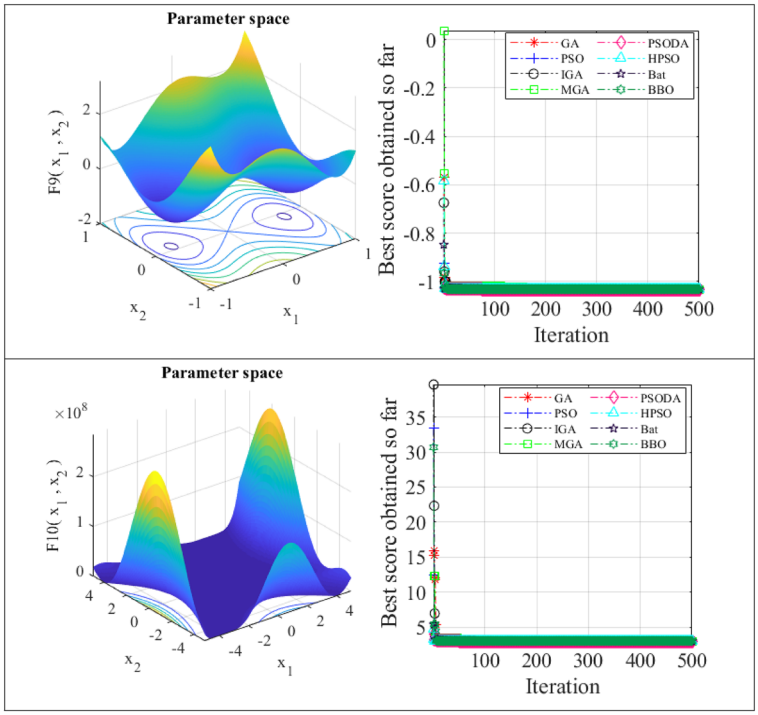


Fig. 8 continued

equal chance to perform crossover and mutation, most likely this algorithm can find a better solution. This result shows that PSODA has more exploitation ability compared to other methods and the RGS technique embedded in PSODA significantly improves the exploitation ability. In multimodal functions, all algorithms except HPSO converge to a near-optimal solution for F7 and F9. The HPSO suffers from an unbalancing between exploration and exploitation. The results of IGA and HPSO prove that using strategies that radically increase the exploration (e.g., chaotic functions) regardless of exploitation it's not appropriate and can take the algorithm away from converging to the optimal solution.

Table 3 shows the average and standard deviation of objective function achieved by the compared methods for different test functions. It can be interpreted that the Bat algorithm achieved the worst result compared to PSO, especially in multimodal functions. The standard Bat algorithm bears many benefits, along with the significant advantage is the fact that it can produce extremely fast convergence at an extremely major stage by transferring from adventure to exploitation. Nevertheless, when we permit the algorithm to switch to the exploitation stage much too immediately, it may result in stagnation after a certain first stage. Therefore, the nature of the Bat algorithm is such that it is expected to work better in Unimodal functions, which have one global optimal, compared to Multimodal functions, which have several local optimal.

The distance between the best solution was found by different methods and the analytical best solution of each test function can be seen in Fig. 10. In addition, the average and standard deviation of this distance value obtained by the compared methods after 30 runs can be seen in Table 4. It can be seen that the best solution that the PSODA reaches to them is near to the analytical best solution more than BBO's best solution. This result indicates that the quality of solutions obtained by PSODA is higher than BBO.

One of the main weaknesses of BBO is that the migration operator, an immigration habitat, and an emigration habitat are selected, respectively, based on the immigration rate and emigration rate first, and then features are migrated from the emigration habitat to the immigration habitat. On one hand, any habitat has a chance to be an emigrating habitat. In other words, BBO uses global topology in migration operations. It needs to check $(ps-1)$ habitats for a given immigration habitat, which is computationally expensive. On the other hand, high-quality habitat has more chance to be an emigration habitat whereas poor habitat is easy to be an immigrating habitat. That is to say, the direction of migration operation is asymmetrical, which decreases population diversity. Unlike the BBO, in PSODA by using the ability of the

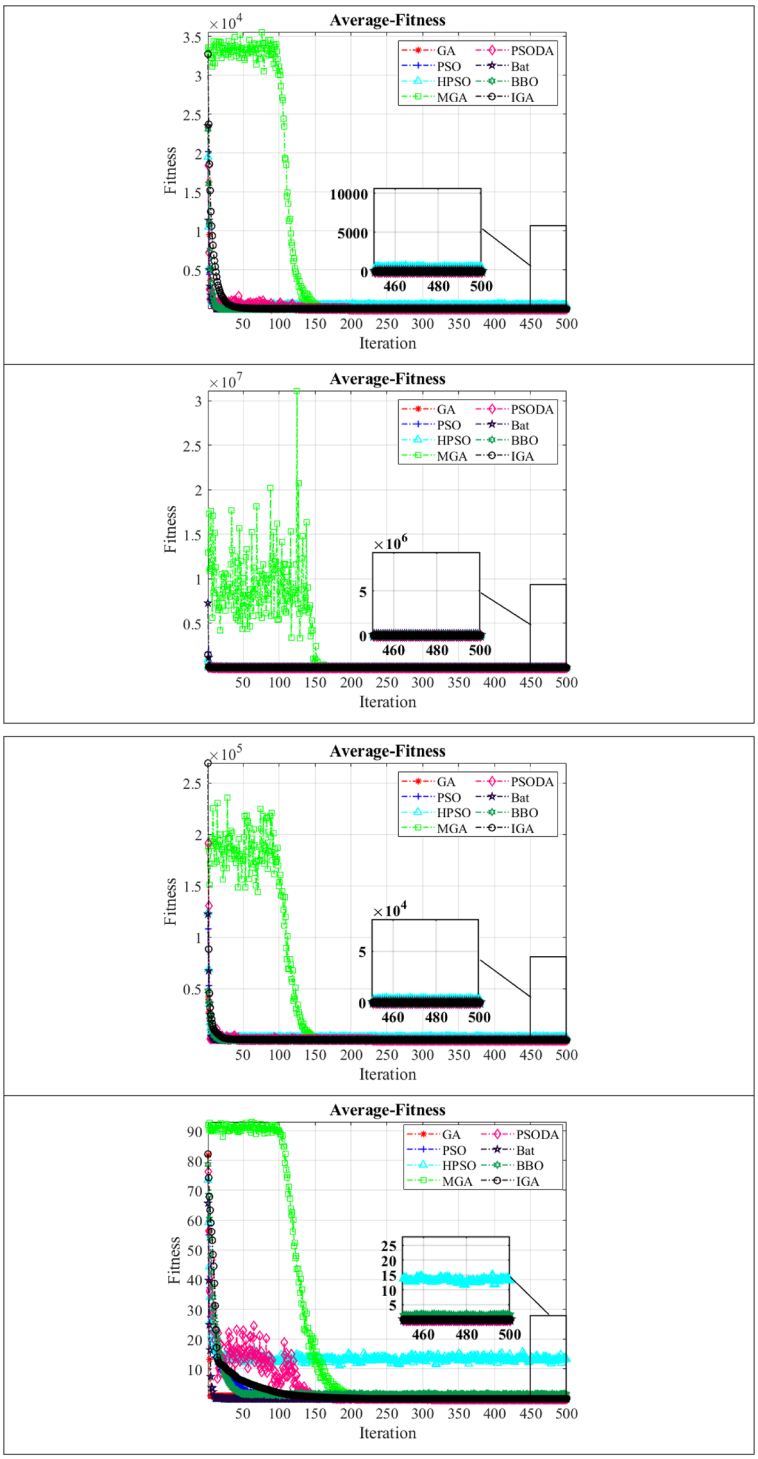


Fig. 9 The average fitness of different compared methods.

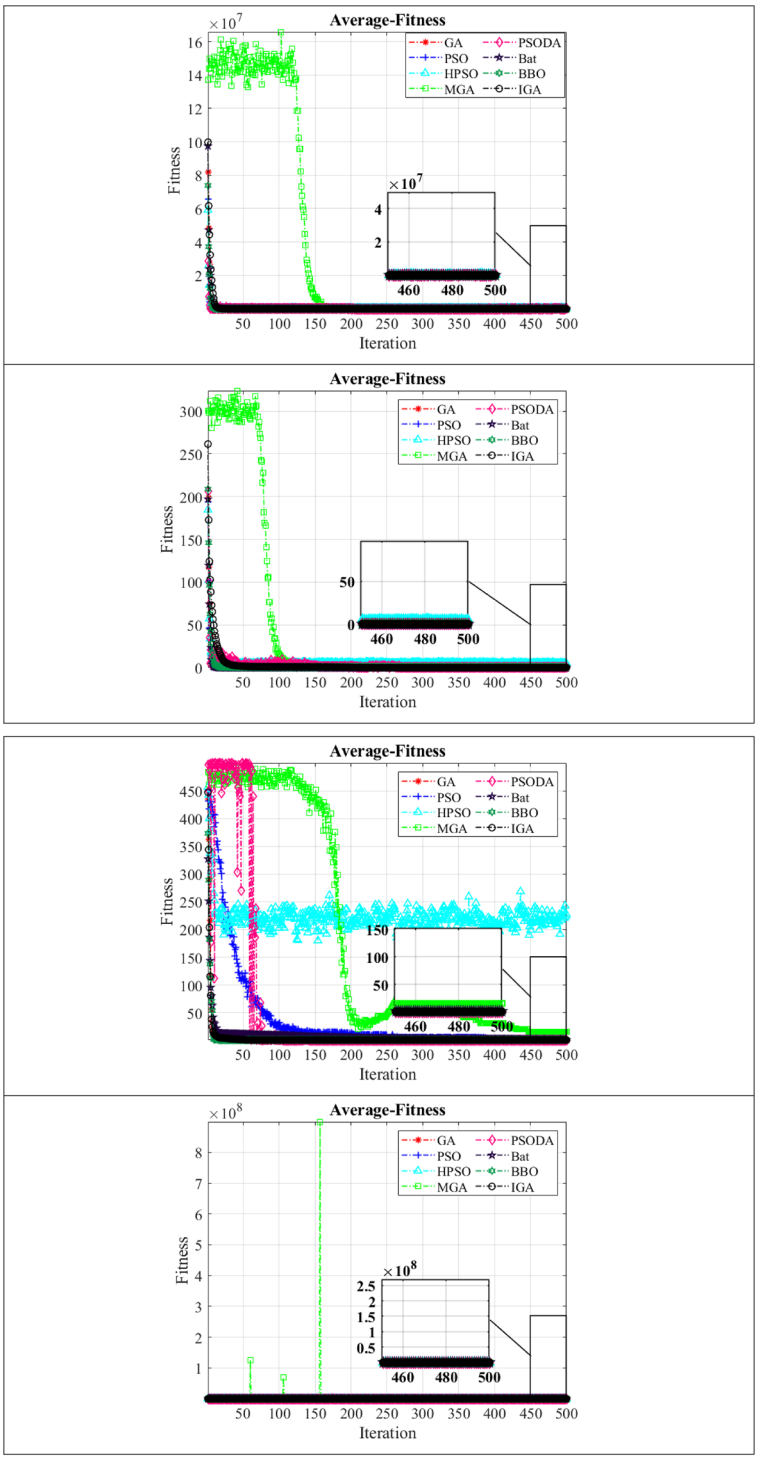


Fig. 9 continued

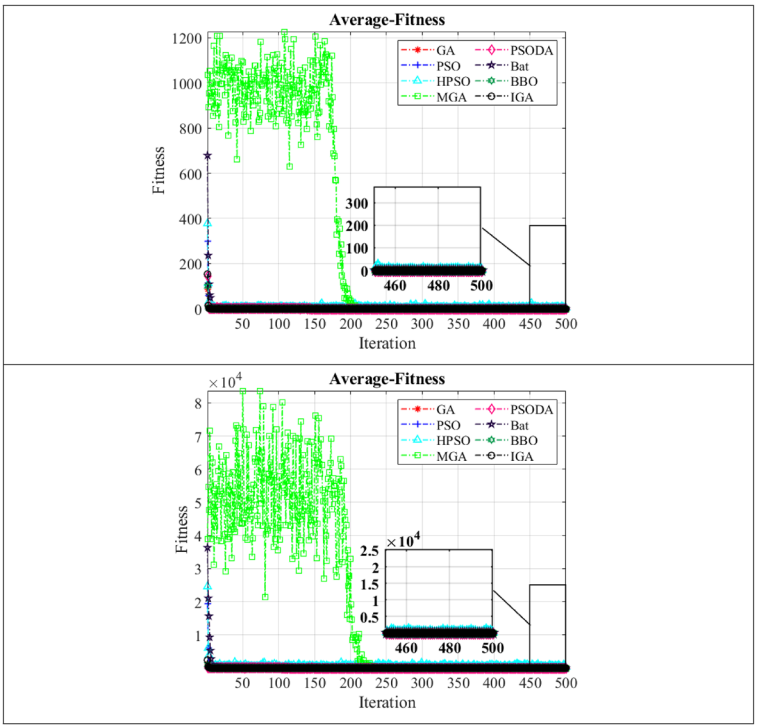


Fig. 9 continued

Table 3 Average and standard deviation objective function values obtained by different methods.

Test functions	*F1*		*F2*		*F3*		*F4*		*F5*	
Metric	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**
GA	4.19E−07	2.11E−07	2.96E−04	4.68E−05	2.91E−10	3.31E−10	4.73E−07	5.16E−07	6.18E−02	5.81E−02
PSO	1.88E−08	4.11E−08	3.71E−06	1.88E−06	3.85E−12	4.15E−13	5.13E−10	2.19E−11	4.89E−02	2.69E−03
IGA	3.17E−07	1.06E−08	2.11E−04	4.33E−04	1.63E−08	3.28E−09	1.59E−08	4.49E−08	7.50E−03	1.18E−04
MGA	1.08E−10	3.27E−10	4.13E−10	2.18E−11	5.22E−18	1.87E−18	**4.28E−13**	2.05E−14	7.11E−03	4.10E−04
PSODA	**2.12E−14**	1.90E−15	**5.08E−15**	3.44E−15	**7.18E−21**	5.36E−22	3.38E−12	4.19E−13	5.27E−03	5.76E−03
HPSO	4.26E−05	4.39E−05	5.28E−04	6.18E−04	5.39E−05	2.17E−05	2.88E−05	7.23E−05	5.29E−02	2.14E−02
Bat	3.15E−04	7.29E−04	1.43E−03	5.32E−04	3.14E−03	3.12E−03	3.95E−04	6.29E−04	8.66E−02	1.03E−02
BBO	4.26E−10	1.93E−10	4.55E−08	1.17E−08	5.07E−15	4.89E−15	8.45E−10	5.61E−10	**4.76E−03**	3.37E−03
Test functions	***F6***		***F7***		***F8***		***F9***		***F10***	
GA	2.62E−13	3.07E−13	1.4348	3.22E−01	**0.0003**	3.21E−06	− 1.0317	3.29E−09	3.0018	4.11E−02
PSO	6.02E−14	2.40E−15	1.4902	2.69E−01	**0.0003**	1.09E−13	− 1.0318	4.86E−21	**3.0000**	6.30E−07
IGA	6.25E−14	4.84E−14	1.2075	1.39E−01	**0.0003**	2.19E−08	− 1.0316	1.09E−04	3.0015	5.38E−03
MGA	**0**	0	**9.9E−01**	3.23E−08	**0.0003**	6.38E−19	− 1.0317	2.18E−07	**3.0000**	5.03E−20
PSODA	**0**	0	**9.9E−01**	3.18E−10	**0.0003**	4.11E−11	**− 1.0316**	6.37E−18	**3.0000**	2.68E−10
HPSO	3.27E−10	1.92E−10	1.5724	5.50E−01	0.0006	2.25E−05	− 1.0318	5.03E−10	**3.1000**	1.003
Bat	4.18E−12	3.23E−12	1.8349	3.89E−01	0.0076	1.94E−06	− 1.0318	4.89E−07	3.1024	5.25E−01
BBO	**0**	0	9.81E−01	4.28E−06	**0.0003**	3.29E−04	**− 1.0316**	9.23E−01	3.0958	1.0183

Bold values in the table are related to the maximum or minimum of the obtained results

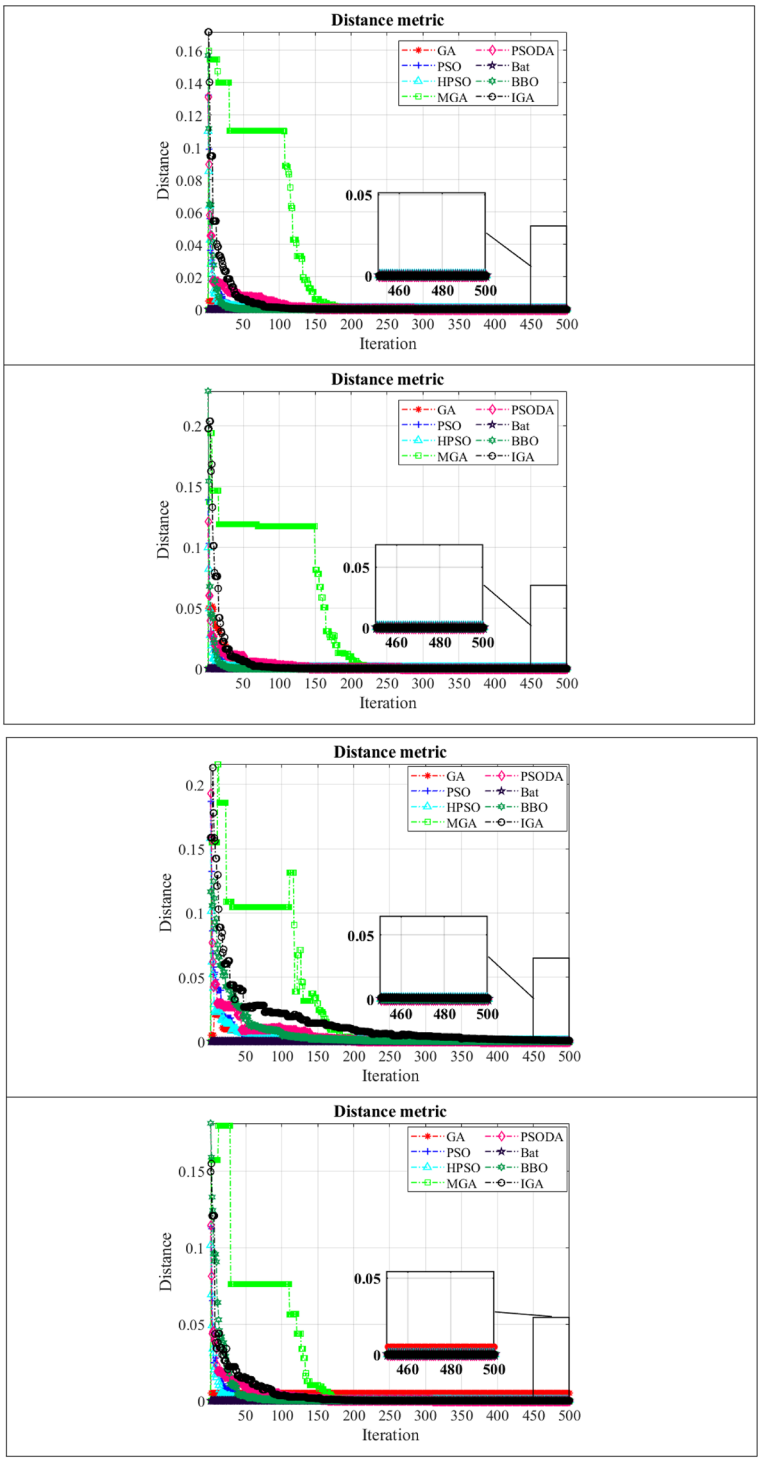


Fig. 10 Distance metric.

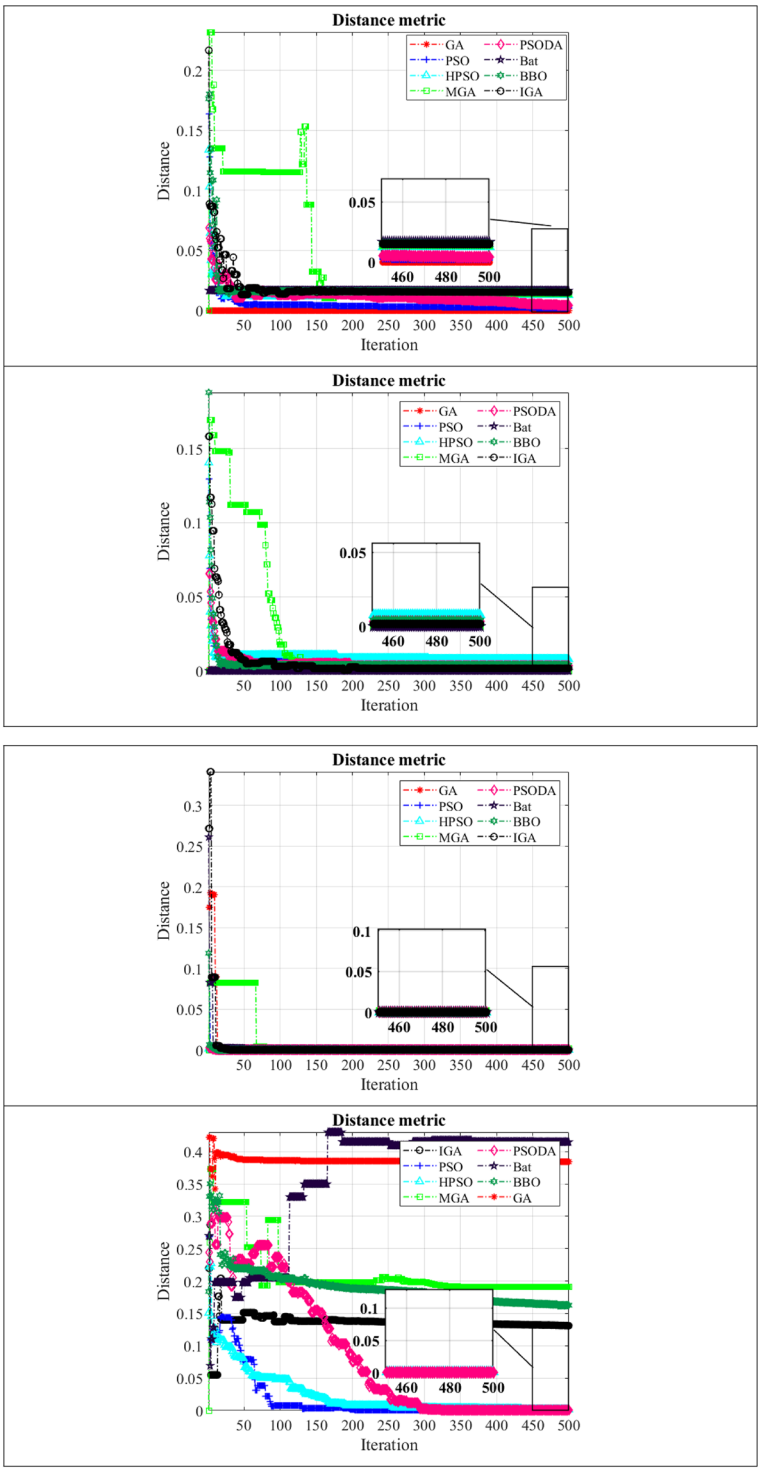


Fig. 10 continued

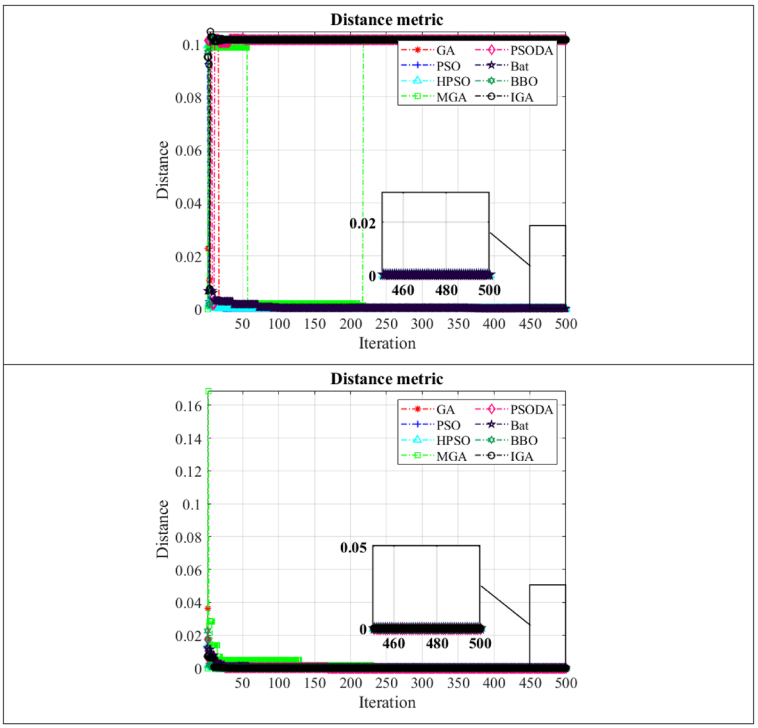


Fig. 10 continued

Table 4 Average and standard deviation of Δ_x obtained by the different methods.

Test functions	***F1***		***F2***		***F3***		***F4***		***F5***	
Metric	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**
GA	0.0054	2.19E−04	0.0094	4.37E−06	0.0007	1.82E−12	0.0046	4.52E−04	0.0002	6.78E−05
PSO	0.0052	4.28E−05	0.0089	4.87E−09	0.00074	1.05E−09	0.0033	3.79E−04	0.0031	5.38E−04
IGA	0.0052	3.80E−18	0.0152	2.49E−11	0.00071	5.61E−12	0.0035	2.83E−05	0.0245	5.76E−05
MGA	0.0045	1.76E−07	0.0007	3.98E−06	0.0007	4.79E−10	0.0018	5.46E−05	0.0197	2.59E−03
PSODA	0.0039	5.38E−04	0.0008	7.30E−18	0.0006	4.02E−06	0.0018	3.11E−04	0.0011	2.42E−04
HPSO	0.0062	2.88E−06	0.0128	4.18E−03	0.00088	5.29E−08	0.0047	2.56E−08	0.0024	4.95E−04
Bat	0.0070	1.69E−05	0.0174	5.31E−03	0.00095	3.18E−06	0.0057	1.57E−03	0.1729	2.88E−02
BBO	0.0050	5.22E−13	0.0011	1.70E−06	0.0007	2.89E−09	0.0020	1.89E−06	0.0239	6.88E−04
Test functions	***F6***		***F7***		***F8***		***F9***		***F10***	
GA	0.0043	3.49E−05	0.0012	2.68E−06	0.3804	4.20E−01	0.0042	1.19E−04	0.0010	7.20E−07
PSO	0.0082	4.39E−04	0.0018	3.19E−09	0.0005	5.07E−12	0.0065	5.22E−06	0.0010	7.44E−07
IGA	0.0086	5.09E−05	0.0025	2.33E−04	0.1253	2.48E−01	0.0089	4.51E−07	0.0010	5.18E−08
MGA	0.0001	2.95E−06	0.0008	3.36E−10	0.2239	4.37E−01	0.0005	5.11E−13	0.0006	4.87E−14
PSODA	0.0001	2.99E−06	0.0008	5.73E−13	0.0003	8.65E−12	0.1544	1.72E−01	0.0006	4.06E−14
HPSO	0.0012	4.11E−04	0.0029	4.20E−08	0.0007	5.38E−15	0.0079	7.37E−04	0.0010	3.95E−05
Bat	0.0078	4.77E−05	0.0037	2.75E−06	0.4107	1.08E−01	0.1563	1.09E−01	0.0011	3.49E−05
BBO	0.0094	5.68E−04	0.0006	4.29E−11	0.1864	5.62E−01	0.0003	6.37E−09	0.0006	2.76E−09

Table 5 The properties of ANFIS structure.

Parameter	Value
Number of input	10
Layer 1: number of membership function	140
Layer 2: number of nodes	140
Layer 3: number of nodes	140
Layer 4: number of rules	18
Layer 5: number of output	1
Membership function	Gaussian

DA algorithm in searching the different locations of solution space a good diversity in population can be created.

The compared approaches [i.e., ANFIS-MGA (Song et al., 2019), ANFIS-IGA (Hong et al., 2013), ANFIS-GA and ANFIS-PSO (Moayedi et al., 2020), ANFIS-HPSO (Chen et al., 2018), ANFIS-PSO–DA (Khunkitti et al., 2018)] are applied and evaluated to model constructed based on C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9 and C_{10} as input parameters and C_{11} as output ones. For estimating the corrosion current density with the trained ANFIS models, the *k*-fold cross-validation technique has been used.

In *K*-fold cross-validation, the original dataset is divided into *K* sub-sets. Then, one sub-set is considered as validation data to test the model and the remaining sub-sets are applied for the training phase. This process is repeated *K* times so that each section is selected once as a validation set. The final result will be the average of *K* folds.

Table 5 summarizes the factors employed for comparisons of training algorithms of the ANFIS network. Considering the 2460 dataset, it has 10 features and so the number of input variables for ANFIS is equal to 10. On the other hand, each input variable has 14 membership functions and the membership functions are Gaussian. This type of membership function has two parameters for optimization and so, in total $2 \times (10 \times 14) = 280$ premise parameters exist. There are 14 rules in the ANFIS model, each rule has 11 parameters for optimization (based on Eq. (15)), and so there are $11 \times 14 = 154$ consequent parameters. In general, the optimization problem has $280 + 154 = 434$ parameters, which must optimize during the finding of best values as well as the best ANFIS model (or training process). In the following subsections, the result of meta-heuristic methods in terms of RMSE,

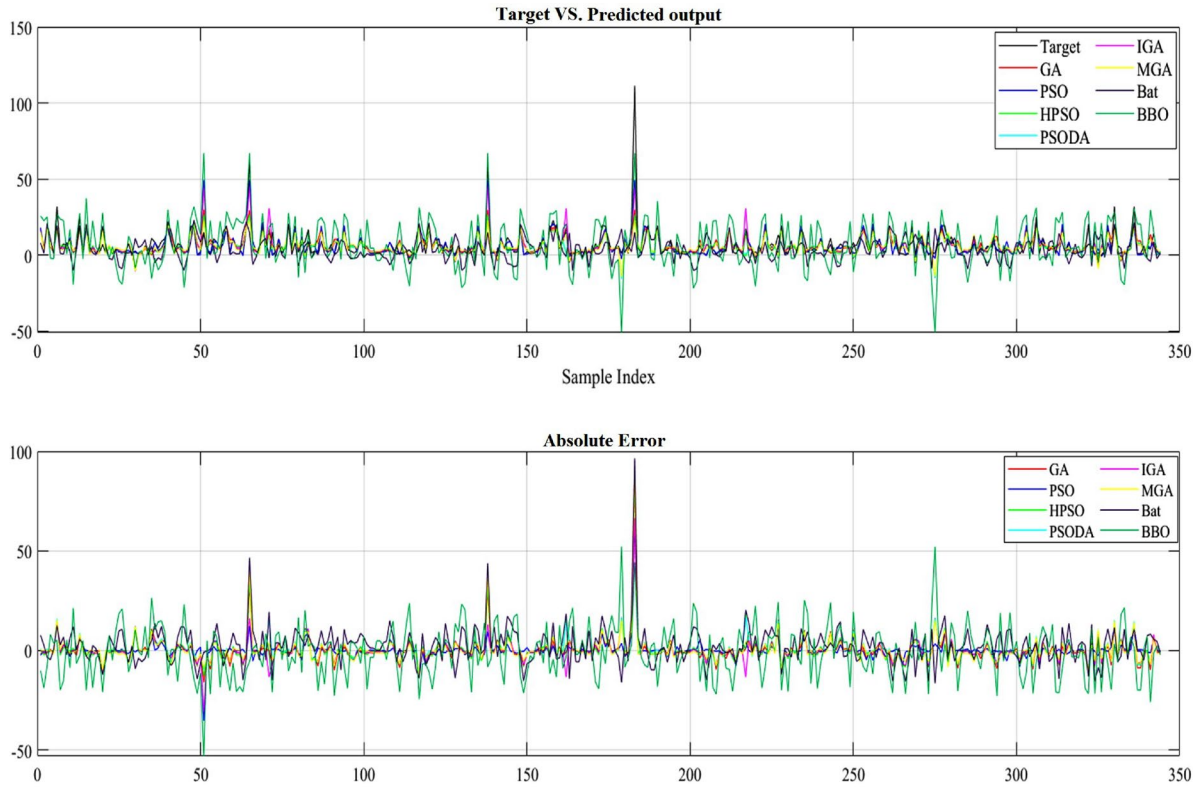


Fig. 11 Performance and absolute error of various approaches through training dataset.

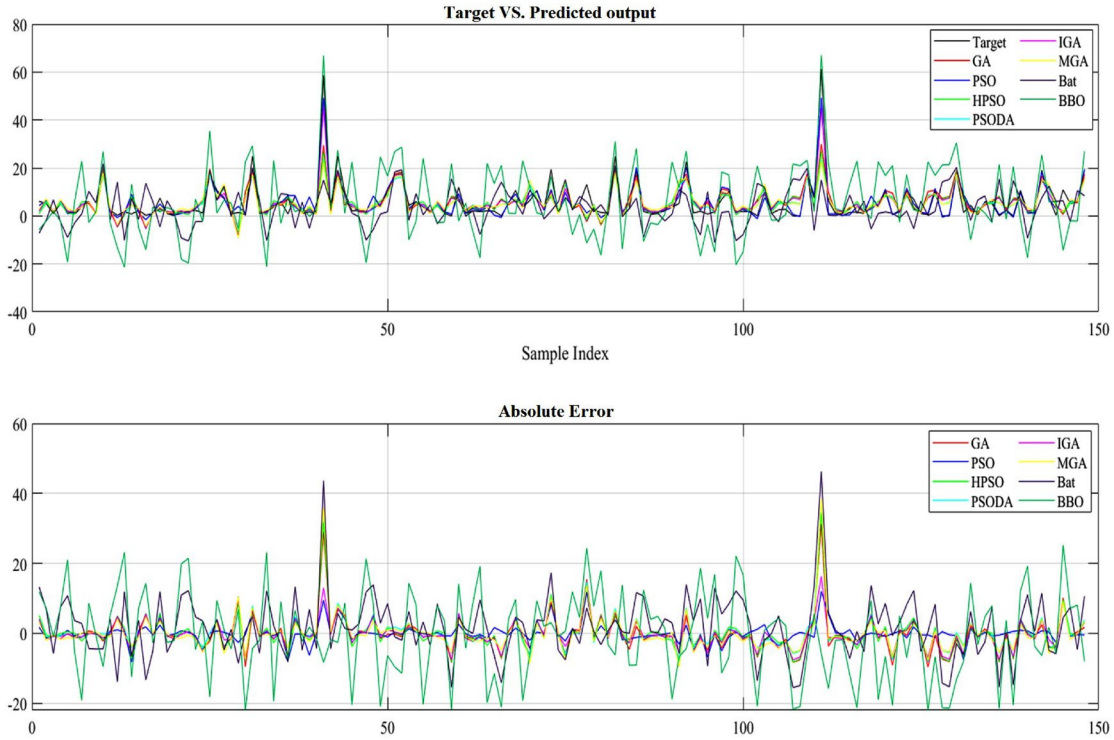


Fig. 12 Performance and absolute error of various approaches through testing dataset.

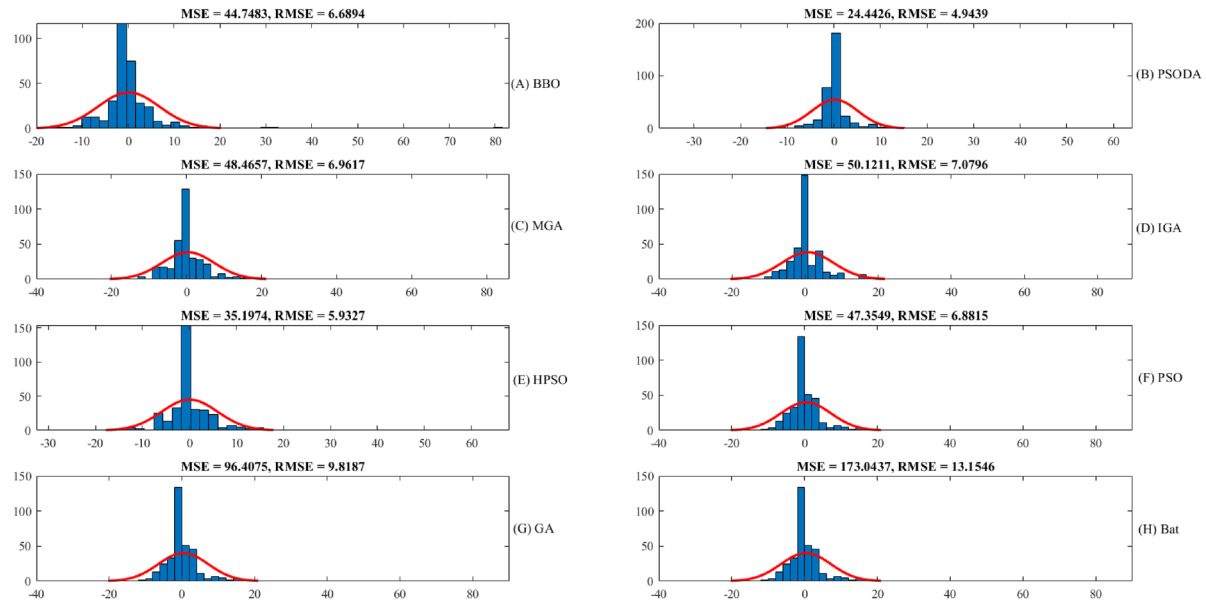


Fig. 13 MSE and RMSE values of different methods for training dataset.

MSE, R^2, VAF, MEDAE, and the efficiency for solving optimization ANFIS parameters' is presented.

4.1.2 Actual Train–Predict Train

Figs. 11 and 12 compare the predictive results that are regularly obtained by different ANFIS models and the actual data for training and testing datasets. The performance on testing and training datasets by different methods is divided into two parts: predict vs. actual value and absolute error part. We can see that using the GA, PSO and their improvements in the training process of ANFIS leads to robust training steps. Although both GA-based and PSO-based models are found satisfactorily capable of predicting corrosion rate, PSO-based models indicate superiority over GA-based in terms of accuracy and on average reduce absolute error by 17%. Among PSO-based, the absolute error of ANFIS-PSO–DA is the smallest, followed by ANFIS-HPSO, then ANFIS-PSO. In GA-based methods, the error value of ANFIS-IGA is the smallest, followed by ANFIS-MGA and ANFIS-GA.

The main reason for this superiority is that the PSO-based method can better control individual search steps with inertia weight compared to GA. While GA focuses only on performing exploitation by crossover operator and exploration by mutation operator and so PSO-based model has a better balance between exploration and exploitation activity by moving individuals towards the best locations (i.e., global best and personal best).

Table 6 Error mean and error std. for different methods.

Method	Train		Test	
	Error mean	Error Std	Error mean	Error Std
GA	0.4586	6.9268	0.2699	5.6155
PSO	0.1002	6.1799	0.0288	4.1593
HPSO	− 0.2043	6.4002	− 0.3394	3.9592
PSO–DA	− 0.0552	5.6300	− 0.1582	2.6873
IGA	− 0.1410	6.7674	− 0.2814	5.1517
MGA	0.1779	6.9048	− 0.0728	5.5206
Bat	0.7239	6.3980	0.2750	4.4894
BBO	− 0.1175	6.0749	0.2107	5.1976

4.1.3 Errors

Theoretically, a predictive model could be evaluated based on error mean and error standard deviation that are obtained based on Eqs. (17) and (18), respectively (Bemani et al., 2020):

$$\text{Error mean} = \frac{\sum_{i=1}^{n} (y_i - \widehat{y_i})}{n}, \tag{17}$$

$$\text{Error StD} = \sqrt{\frac{\sum_{i=1}^{n} (E_i - \overline{E})}{n-1}}, \tag{18}$$

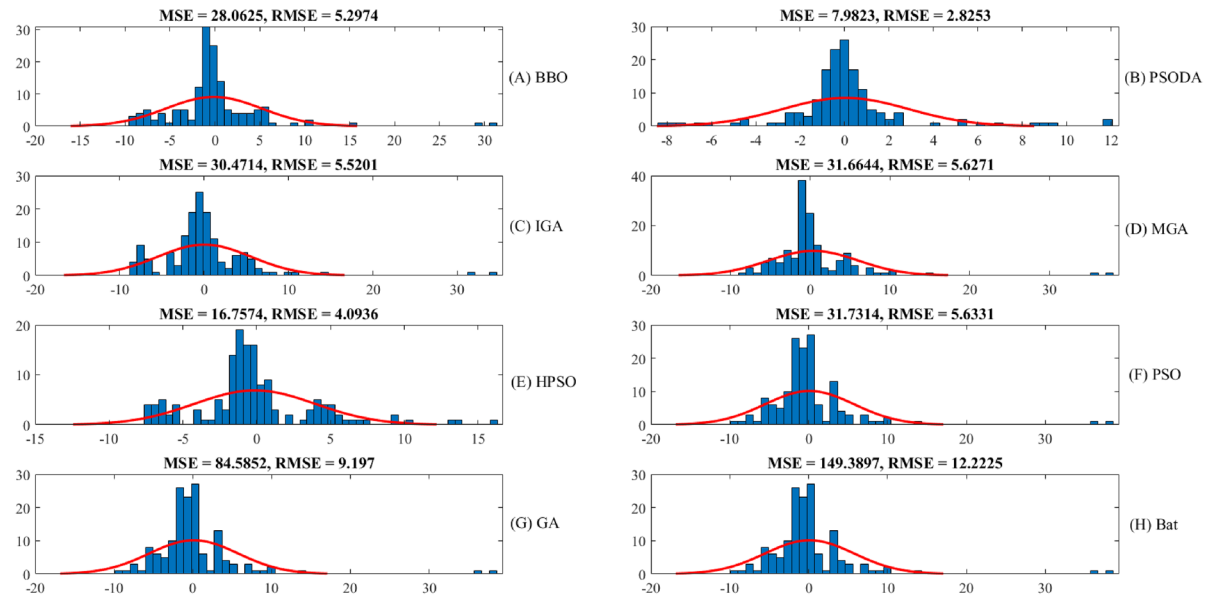


Fig. 14 MSE and RMSE values of different methods for testing dataset.

where E represents the error of predicting a value and $\overline{E}$ indicates the mean error. The RMSE parameter indicates how much output of each meta-heuristic model is close to the real value. As it can be seen in Fig. 13, RMSE values of ANFIS-IGA, ANFIS-PSO–DA, ANFIS-MGA, ANFIS-GA, ANFIS-HPSO, and ANFIS-PSO have been calculated for the training steps 6.6586, 5.5479, 6.8101, 6.8564, 6.2857, and 6.3529, respectively. This shows that compared to other models, ANFIS-PSO–DA has the best performance while ANFIS-GA has the worst result for training step.

The ANFIS-PSO improves RMSE by 8% compared to ANFIS-MGA, on the other hand, ANFIS-MGA with changing the selection strategy from the standard selection techniques [i.e., roulette wheel selection (RWS) and tournament selection (TS)] to RGS improves the RMSE by 9.2% compared to standard GA. Therefore, this method gets better result compared to the standard GA but it cannot create suitable diversity in population over PSO method due to the lack of exploitation performance.

We know that determining the best model based on RMSE for training step is not enough, and it is necessary to check the RMSE level for testing step. According to the results shown in Fig. 14, the values of RMSE for ANFIS-GA, ANFIS-PSO–DA, ANFIS-MGA, ANFIS-A, ANFIS-HPSO and ANFIS-PSO are 5.1419, 2.6828, 5.5024, 5.603, 3.9554, and 4.141, respectively. Therefore, the best performance is achieved by ANFIS-PSO–DA and the worst model is ANFIS-GA.

The error distribution plots in Figs. 13 and 14 show that most of the errors are concentrated near zero, specifically for PSO–DA.

Table 6 illustrates the standard deviation error and mean error achieved by compared methods for both training and testing datasets. It can be seen that HPSO, PSO–DA, and IGA predict output value higher than the actual value. In addition, PSO–DA achieves the low value for standard deviation error and so shows good stability and performance.

4.2 Comparison of Other Methods

There are some common metrics, i.e., mean absolute percentage error (MAPE), square coefficient of determination (R^2), the variance accounted for (VAF), and median of absolute error (MEDAE) for performance evaluation of different models. The square coefficient of determination (R^2) is obtained based on Eq. (13). The MAPE, Efficiency, VAF, and MEDAE are defined as Eqs. (19)–(21), respectively (Umrao et al., 2018):

$$\text{MAPE} = \frac{1}{n}\sum_{j=1}^{n}\left|\frac{y_j - \widehat{y}_j}{y_j}\right| \times 100, \tag{19}$$

$$\text{Efficiency} = \frac{\sum_{j=1}^{n}\left(y_j - \overline{y}\right)^2 - \sum_{j=1}^{n}\left(\widehat{y}_j - y_j\right)^2}{\sum_{j=1}^{n}\left(y_j - \overline{y}\right)^2}, \tag{20}$$

$$\text{VAF} = \left[1 - \frac{\text{var}(y - \widehat{y})}{\text{var}(y)}\right], \tag{21}$$

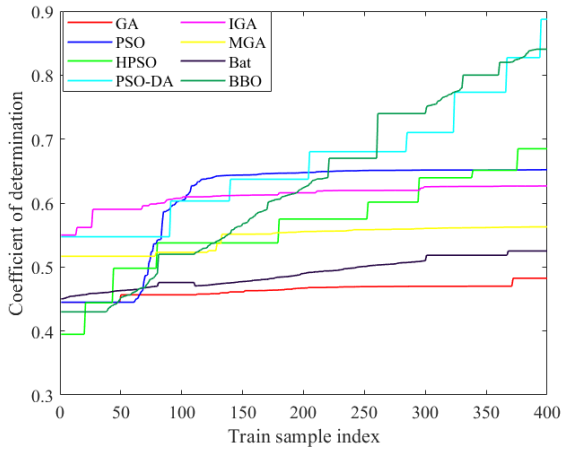
$$\text{MEDAE} = \text{median}(y - \widehat{y}), \tag{22}$$

where y is the actual output for $\widehat{y}$ that is predicted by the networks. $\overline{y}$ is the mean of the whole (actual) outputs and n is the number of outputs.

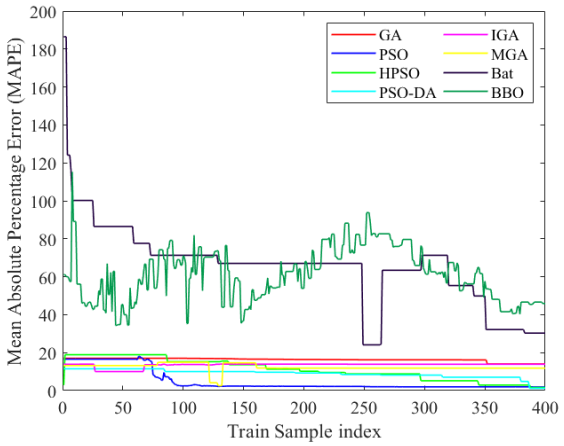
PSO–DA achieves the highest value for R^2 and efficiency. Also, it on average improves these indices compared to GA, MGA, IGA, HPSO, and PSO by 35%, 29%, 25%, 20%, and 22.5%, respectively. In addition, it obtains the lowest error among the other methods and on average reduces the error by 5% and 3% compared to PSO and HPSO, respectively. The main reason for this superiority is improved local search in PSO–DA with dragonfly levy flight technique compared to PSO and HPSO. This technique helps PSO–DA to explore wide search problem areas than the two other methods. It can be seen from Fig. 15b that PSO and HPSO achieve the same error in the last iteration while they have a great different error at iteration 100 (i.e., 15 for HPSO and 2.5 for PSO). But as the number of iterations increases, the influence of using the chaotic function in HPSO is more appeared. The chaotic function creates different search steps and so each individual in HPSO can take a small and big step, while an individual in PSO takes a steady step and different steps help HPSO to escape from local optimal.

The performance results of compared methods on the test dataset can be seen in Table 7. It is seen that IGA achieves the best result among all of the GA-based methods (i.e., MGA and GA). This superiority demonstrates the importance of definition for the initial population. In IGA, the population is initialized with a chaotic function, while in GA and MGA it randomly creates population. The chaotic technique helps IGA to a better spread individuals.

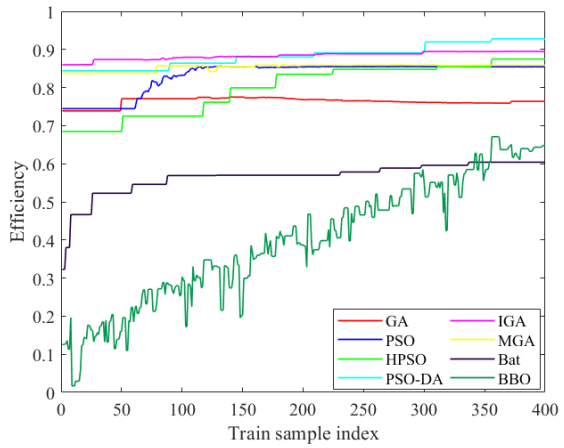
To gain a better insight into the prediction success of models, the actual values were plotted against predicted ones. Figs. 16a–f and 17a–f show the results for the train dataset and test dataset, respectively. As can be observed



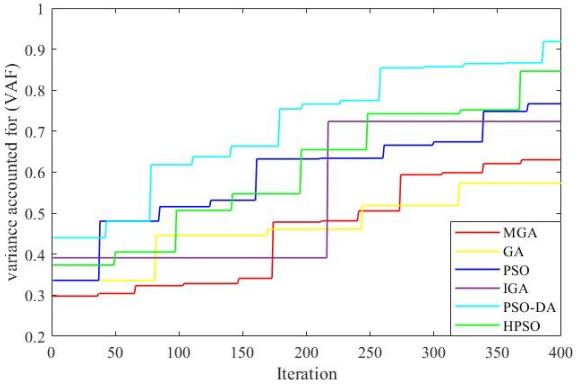
(a) R-squared



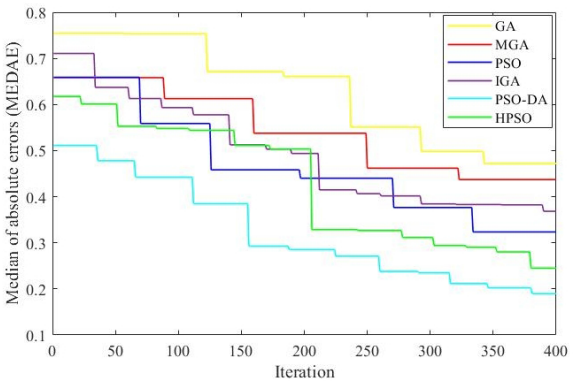
(b) Mean absolute percentage error (MAPE)



(c) Efficiency



(d) Variance accounted for (VAF)



(e) Median of absolute errors (MEDAE)

Fig. 15 Results for training dataset over 400 iterations.

Table 7 Results for the test dataset.

Methods	MAPE	Efficiency	R^2	VAF	MEDAE
GA	16.70	0.82	0.56	0.59	0.83
PSO	15.59	0.86	0.63	0.72	0.33
HPSO	11.15	0.89	0.75	0.77	0.51
PSO–DA	**1.67**	**0.97**	**0.92**	**0.93**	**0.14**
IGA	16.35	0.84	0.58	0.66	0.62
MGA	15.15	0.83	0.59	0.62	0.77
Bat	18.67	0.78	0.58	0.58	0.80
BBO	12.07	0.87	0.79	0.77	0.48

Bold numbers are related to the maximum or minimum of the obtained results

from Fig. 16, for most methods the points are scattered around the fit line (the fit line represents the experimental data). In other words, there is a good adjustment between the outputs of models and the actual values in the train dataset. Fig. 16 shows all methods approximately learn well and can predict unseen data very well. Fig. 17 indicates the performance of compared methods on the test dataset.

It can be seen that GA and MGA can predict output value as well as HPSO and PSO–DA in train dataset. Therefore, it shows that these methods (i.e., GA and MGA) get overfit and cannot be predicted these unseen data very well. In predicting of C_{11}, the ANFIS-PSO–DA model presents a low error in comparison with the other models with having a correlation coefficient close to one.

4.3 PSO–DA ANFIS Information

In this section, the properties of PSO–DA as the best model among compared methods are explained and then the influence of a different number of control parameters (i.e., population size and iteration) and the number of membership functions are investigated. Fig. 18 shows the optimal value of the membership function.

By finding optimal values for consequent parameters, the ANFIS fuzzy rules could be defined as fixed rules for testing the network. Table 8 illustrates 14 rules that are prepared using various combinations of linguistic terms after the training process. It is clear that fuzzy rules can be effective in determining the effect of parameters on

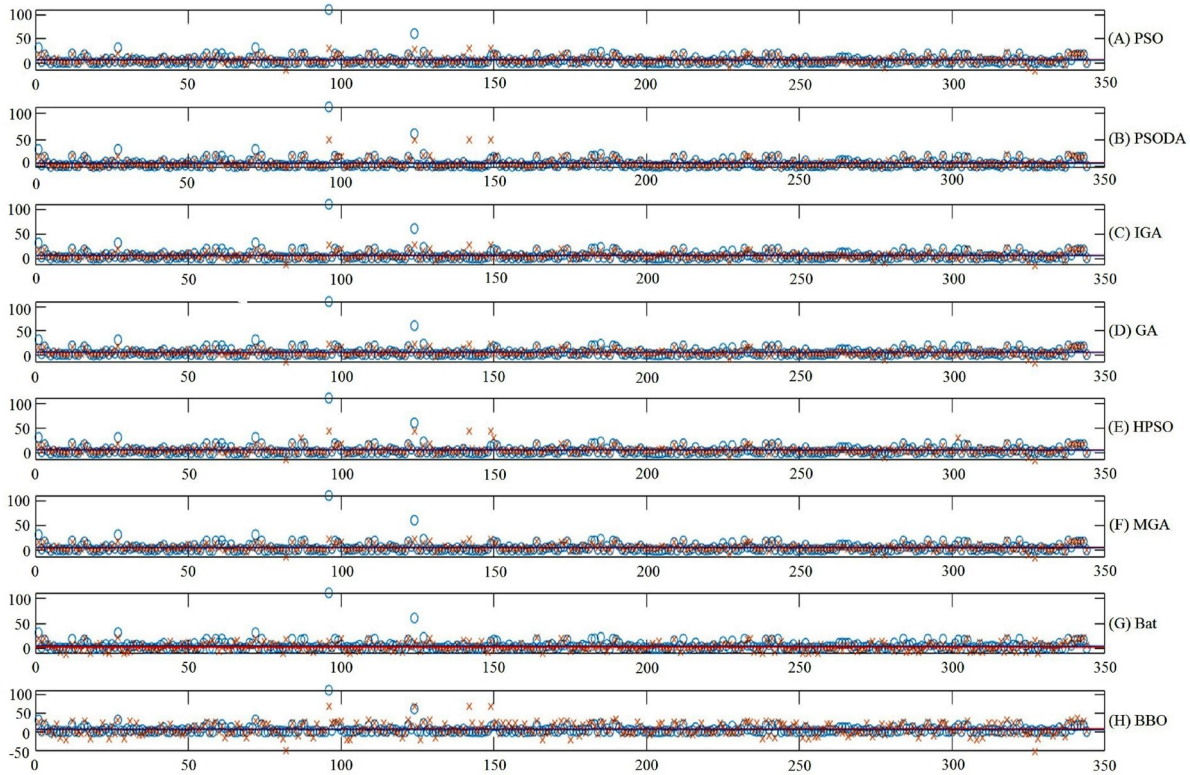


Fig. 16 Regression line plot for the train dataset.

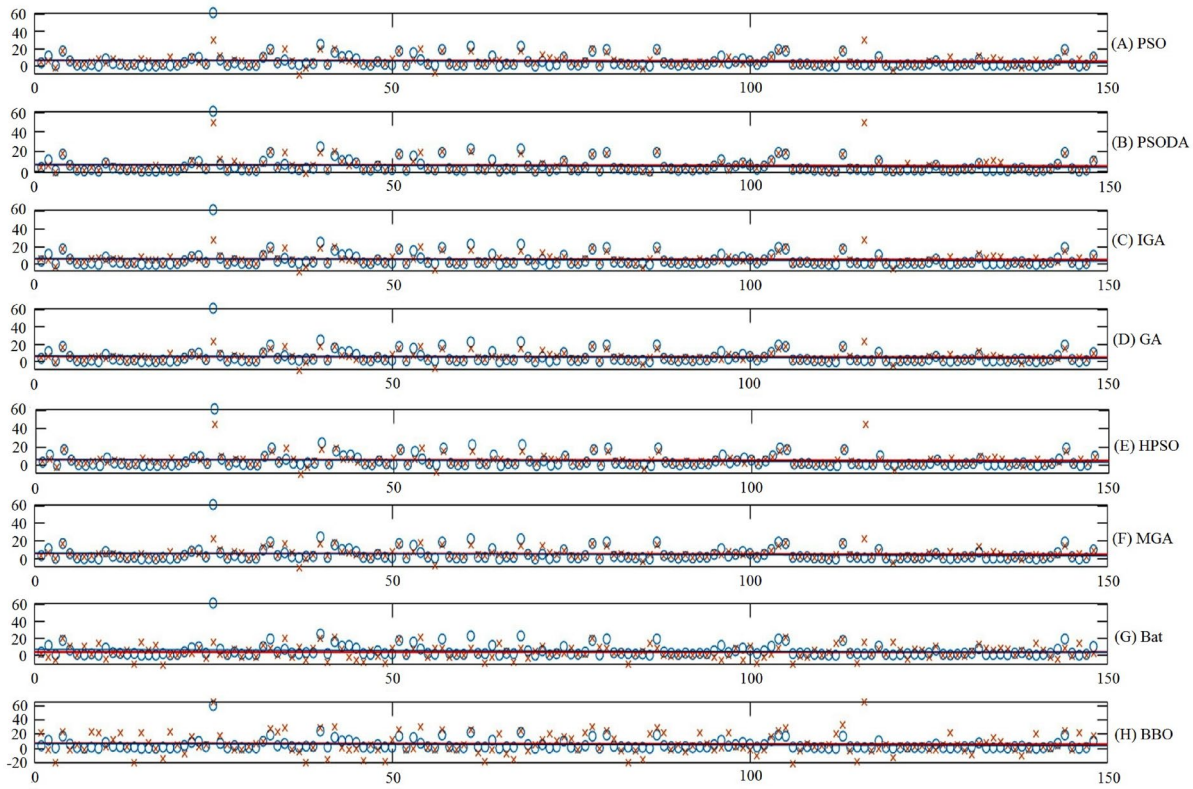


Fig. 17 Regression line plot for test dataset.

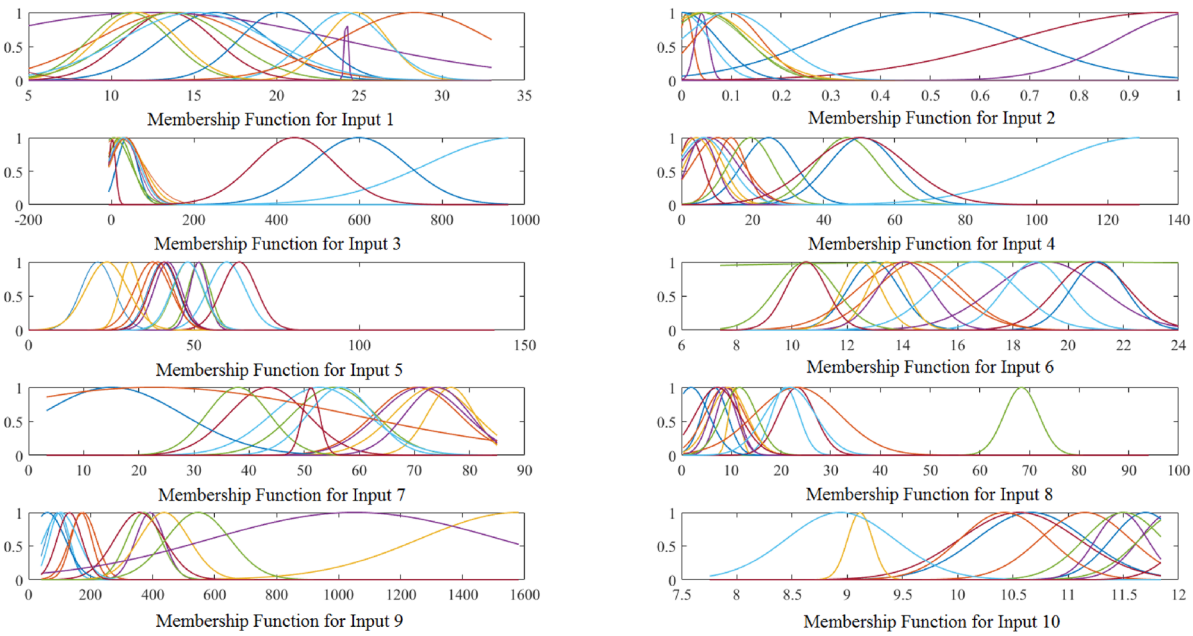


Fig. 18 Gaussian membership functions of input parameters after the training process by PSO–DA.

Table 8 The optimal consequent parameters after training.

Rule numbers	Rules
1	$C_{11(1)} = -0.1155C_1 - 0.0471C_2 - 0.0394C_3 + 0.3621C_4 - 0.1798C_5 + 0.3021C_6 - 0.0451C_7 - 0.11C_8 + 0.0098C_9 - 3.4646C_{10} + 45.1552$
2	$C_{11(2)} = -0.0595C_1 - 0.0327C_2 - 0.0395C_3 + 0.4252C_4 - 0.2229C_5 + 0.6232C_6 - 0.0445C_7 - 0.2647C_8 + 0.01C_9 - 3.4798C_{10} + 45.0383$
3	$C_{11(3)} = -0.0447C_1 - 0.0478C_2 - 0.0393C_3 + 0.3049C_4 - 0.1772C_5 + 0.2875C_6 - 0.0462C_7 - 0.1382C_8 - 0.2404C_9 - 2.8651C_{10} + 36.2256$
4	$C_{11(4)} = -0.0605C_1 - 0.0476C_2 - 0.0408C_3 + 0.3528C_4 - 0.1644C_5 + 0.3031C_6 - 0.0433C_7 - 0.1854C_8 + 0.0116C_9 - 3.6469C_{10} + 45.5615$
5	$C_{11(5)} = -0.0613C_1 - 1.2013C_2 - 0.0007C_3 + 0.2849C_4 - 0.1646C_5 + 0.3281C_6 - 0.0481C_7 - 0.1984C_8 + 0.0254C_9 - 3.8826C_{10} + 46.3025$
6	$C_{11(6)} = -0.0061C_1 - 0.0703C_2 - 0.0409C_3 + 0.3506C_4 - 0.1718_5 + 0.3024C_6 - 0.0513C_7 - 0.1488C_8 + 0.0112C_9 - 3.3384C_{10} + 44.5699$
7	$C_{11(7)} = -0.0614C_1 - 0.3094C_2 - 0.0384C_3 + 0.1063C_4 - 0.0015C_5 + 0.3002C_6 - 0.0454C_7 - 0.1612C_8 + 0.0087C_9 - 3.7786C_{10} + 43.5307$
8	$C_{11(8)} = -0.0591C_1 - 0.1390C_2 - 0.0395C_3 + 0.3467C_4 - 0.1637C_5 + 0.3469C_6 - 0.0451C_7 - 0.1699C_8 + 0.0091C_9 - 3.5540C_{10} + 43.3374$
9	$C_{11(9)} = -0.0599C_1 - 0.0495C_2 - 0.0944C_3 + 0.3234C_4 - 0.1250C_5 + 0.2858C_6 - 0.0468C_7 - 0.0953C_8 + 0.0083C_9 - 3.5426C_{10} + 44.1564$
10	$C_{11(10)} = -0.0539C_1 - 0.0367C_2 - 0.0383C_3 + 0.3950C_4 - 0.1626C_5 + 0.3528C_6 - 0.0450C_7 - 0.1556C_8 + 0.0330C_9 - 3.6017C_{10} + 50.91$
11	$C_{11(11)} = -0.0397C_1 - 0.0467C_2 - 0.0194C_3 + 0.3538C_4 - 0.1689C_5 + 0.2494C_6 - 0.0398C_7 - 0.1519C_8 + 0.0107C_9 - 3.5988C_{10} + 39.5035$
12	$C_{11(12)} = -0.0721C_1 - 0.0497C_2 - 0.0318C_3 + 0.3429C_4 - 0.1622C_5 + 0.2888C_6 - 0.0822C_7 - 0.1854C_8 + 0.0070C_9 - 3.7603C_{10} + 44.7857$
13	$C_{11(13)} = -0.0592C_1 - 0.0453C_2 - 0.0401C_3 + 0.3538C_4 - 0.1591C_5 + 0.2577C_6 - 0.0433C_7 - 0.2697C_8 + 0.0147C_9 - 3.5458C_{10} + 40.6151$
14	$C_{11(14)} = -0.0605C_1 - 0.0494C_2 - 0.0410C_3 + 0.3982C_4 - 0.1391C_5 + 0.3033C_6 - 0.0451C_7 - 0.1764C_8 + 0.0098C_9 - 3.5827C_{10} + 42.2834$

model performance. Fig. 19 represents the data analyzing result for the sensitivity of this experimental case. We can see that C_5 and C_{10} (each of them with around 30%) have more influence on C_{11}, while C_4 (with around 1%) has the least influence on that performance.

4.3.1 Investigation of PSO–DA ANFIS Performance

Three parameters for training ANFIS are available that should be adjusted. They are two common control parameters for PSO–DA (i.e., population size and the number of iteration) and the number of membership functions for ANFIS. The values of these important parameters have a major influence on the solution quality and execution time. Nevertheless, these values are usually determined empirically through pilot runs.

Table 9 shows the influence of setting the different number of membership functions for input variables. Accordingly, the best performance is achieved by setting the number of membership functions to 14. The number of variables to be optimized decreases as the number of membership functions decreases and so the execution time is reduced. Nevertheless, the efficiency and the accuracy of the model are also reduced. On the other hand, with more than 14 membership functions, it does not show significant improvement results while the execution time is increased.

The influence of setting different sizes for population and the different number of iterations is shown in Table 10. It can be interpreted that the PSO–DA shows the best performance by setting maximum iteration to 400 and population size to 50 the PSO–DA. By setting population size lower than 50, the method face to lack of diversity and so convergence to local optimal. Also by setting the number of iterations lower than 400, the model doesn't enough time to explore search space very well and hence convergences to the local optimal.

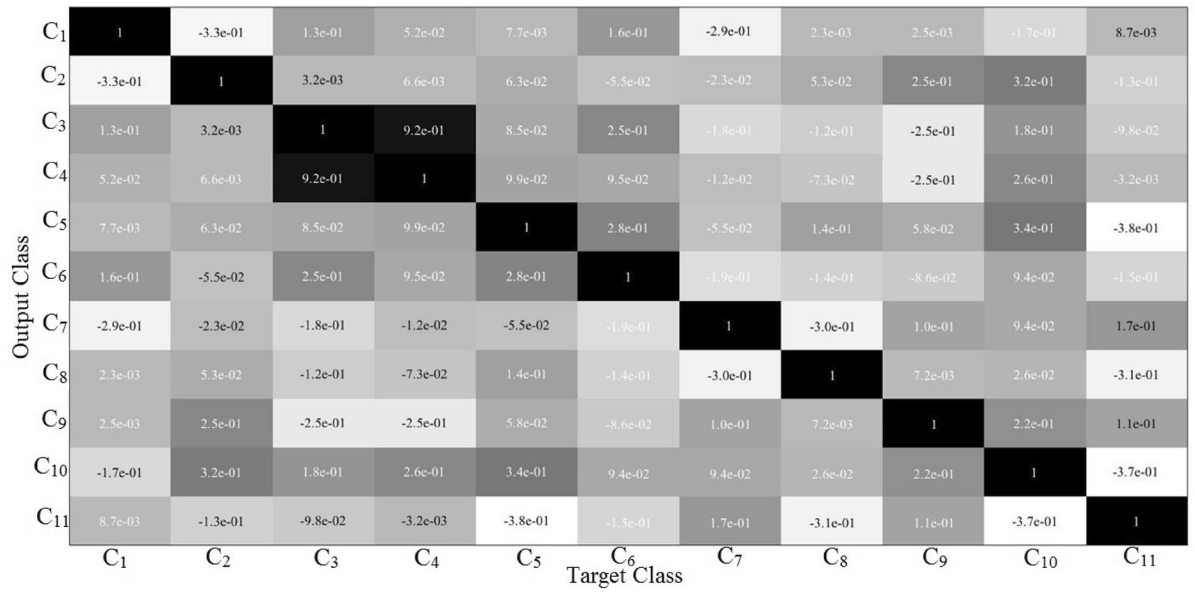


Fig. 19 Relationships between the input parameters and output parameter.

Table 9 Influence of the various number of membership functions on PSO–DA ANFIS.

Number of membership function	MAPE	Efficiency	R^2	RMSE
5	3.56	0.85	0.84	4.44
8	3.60	0.85	0.84	4.51
11	2.09	0.91	0.90	3.52
14	**1.67**	**0.97**	**0.92**	**2.68**
18	1.88	0.94	0.89	2.93

Bold numbers in the table are related to the maximum or minimum of the obtained results

4.3.2 Sensitivity Analysis

The sensitivity analysis is used to rank the effect of each parameter on C_{11} of steel in concrete. The results of this analysis can extend the useful service life of the RC structures in terms of sustainable new structure's design and optimization of renovation and maintenance of such structures (Zhang & Lounis, 2006). As seen from Fig. 19, the importance of each input parameter in determining the corrosion rate of steel in descending order (from top values to low values) is as C_5, C_{10}, C_8, C_7, C_6, C_2, C_9, C_3, C_1, and C_4. According to results, C_{11} is inversely related to C_5, C_{10}, C_8, C_6, C_2, C_3, and C_4, and decreases with increasing their proportion. But C_{11} is directly related to the C_7, C_9 and C_1. In the following, the sensitivity of major input parameters to the C_{11} as well as the service life of RC structures will be discussed.

4.3.2.1 Sensitivity of C_{11} to the Compressive Strength of the Concrete C_{11} depends on C_5 (Stewart & Mullard, 2007) and compressive strength relates to the permeability of concrete. By decreasing the ratio of water to binder in the concrete, the concrete permeability decreases, and consequently the compressive strength of concrete will increase. Concrete with a high compressive strength

Table 10 Influence of various population sizes and iterations on PSO–DA ANFIS.

Method	Population size				Iteration			
	25	50	75	100	100	200	300	400
MAPE	11.76	**1.67**	1.75	1.71	14.83	11.52	7.37	**1.67**
Efficiency	0.85	**0.97**	0.95	0.95	0.82	0.86	0.91	**0.97**
R^2	0.87	**0.92**	0.92	0.91	0.85	0.86	0.85	**0.92**
RMSE	3.43	**2.68**	2.70	2.69	4.68	4.09	3.71	**2.68**

Bold numbers in the table are related to the maximum or minimum of the obtained results

extends the time for chlorine ions to penetrate. As a result, the incubation time to start the corrosion and service life of structures will be longer (Topçu et al., 2009).

4.3.2.2 Sensitivity of C_{11} to C_{10} The rate of steel corrosion decreases with increasing pH. Corrosion of concrete reinforcement occurs very slowly in normal circumstances due to the high pH within the concrete. In fact, high pH of the pore solution and passivation of the reinforcement can change by occurring some unfavorable chemical processes, which leads to an increase in the corrosion rate of the steel (Garcés et al., 2011). The main reason for reducing the pH of the pore solution is the carbonation of the concrete, leading to the cracking

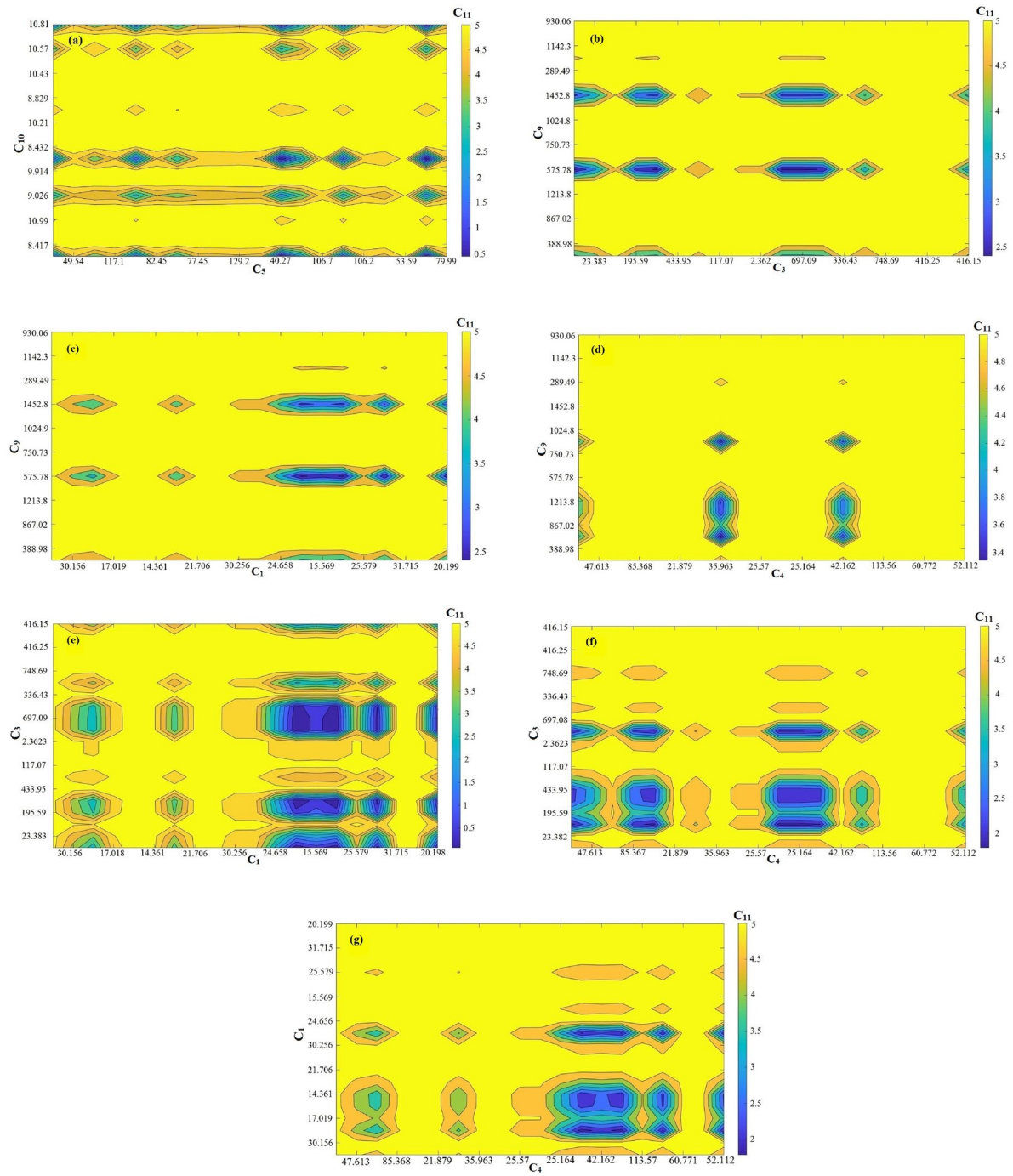


Fig. 20 Binary changes for some of selecting parameters: **a** C_{10}–C_5, **b** C_9–C_3, **c** C_9–C_1, **d** C_9–C_4, **e** C_3–C_1, **f** C_3–C_4, and **g** C_1–C_4.

and dissolution of the passive layer of the rebar. In fact, carbonation reduces hydroxyl ions in the pore solution of alkaline concrete and thus leads to a decrease in concrete pH (Otieno et al., 2019).

4.3.2.3 Sensitivity of C_{11} to C_8 The electrical resistance of a substance is its ability to resist the flow of free electrical charges when exposed to an electric field (Alhajj et al., 2019). Concrete resistivity (ρ, Ω m), also known as electrical resistivity, is mainly affected by the water to cement ratio, the aggregates size, binder type, and conditions of the concrete preparation (Rodrigues et al., 2020). Concrete electrical resistivity is one of the important parameters for evaluating steel reinforcement corrosion. It is now generally confirmed that as the concrete resistivity increases under the normal environmental circumstances, the corrosion current density and consequently the corrosion rate decreases (Hornbostel et al., 2013; Wang et al., 2019).

4.3.2.4 Sensitivity of C_{11} to C_7 and C_6 Concrete cover acts as a fence against the penetration of aggressive materials, which are necessary to initiate and stabilize the corrosion process (Muthulingam & Rao, 2014). Corrosion current density depends on the thickness of the concrete cover (Wang et al., 2019), and as the thickness of the cover increases, more corrosion is necessary to start the cracking in the surface of the concrete (Otieno et al., 2019). Because of the presence of thicker concrete cover, more tensile energy is required to the manifestation of cracks in the concrete (Zhao et al., 2011). It should be noted that the influence of cover depth on the corrosion rate of steel can be different after the depassivation of steel. In this case, due to a better electrical potential circulation in concrete with a thicker cover, the corrosion rate of the reinforcement will increase (Pour-Ghaz et al., 2009).

In general, a larger diameter of steel bars generates more tensile energy in concrete. As a result, a crack can initiate more easily in a larger diameter steel bar on the concrete cover than the smaller one (Zhao et al., 2011). But after the corrosion propagation, the diameter of the rebar decreases with increasing the corrosion current density according to Eq. 23 (Molina et al., 1993):

$$\varphi(t) = \varphi(i) - 0.023 I_{\text{corr}} t, \tag{23}$$

where $\phi(t)$ is the diameter of rebar (mm) at time t, $\phi(i)$ is the rebar diameter (mm) at the start of service life, I_{corr} is corrosion current density ($\mu A/cm^2$), t is the time passed (years) from the start of the propagation period and 0.023 is the conversion factor for uniform corrosion.

4.3.2.5 Sensitivity of C_{11} to C_9 Concrete can be contaminated with chlorine ions through exposure to a marine environment, the use of deicing salts, or even during the manufacturing of concrete. Chloride penetration generally takes place by suction and diffusion via the capillary pores. Therefore, the time to the corrosion initiation is highly related to the diffusion coefficient of chloride in concrete. Diffusion is influenced by the size distribution of the pores in the concrete, which is related to the ratio of water to cement. The presence of defects such as cracks in the steel–concrete interface provides easier and faster paths for chlorine penetration (Rodrigues et al., 2020). The existence of chloride ions above the critical level (more than 0.4% compared to the mass of cement) in RC structures, around the steel reinforcement, makes the passive state unstable and thus causes the corrosion process to begin (Medeiros et al., 2013). In fact, the existence of chloride ions destabilizes the passive layer of reinforcement by the occurrence of pitting corrosion (Li et al., 2020). Thus, as the chloride concentration increases, the corrosion rate will increase.

4.4 Comparison of the Effect of Selected Variables on C_{11}

Observing how the operating parameters change and their effect on the corrosion rate can provide a good view of the interaction between the parameters. As an example, Fig. 20a–g shows these binary changes for some of selecting parameters. It should be noted that the values of other parameters are considered to draw each graph equivalent to the sample with the lowest corrosion rate. As shown in the guide bar, the blue region indicates the conditions with less corrosion current density, and yellow indicates the conditions with higher corrosion rates. These diagrams are a good guide for selecting the best parameter values in specific operating conditions.

5 Summary and Conclusions

Effective construction, maintenance of concrete reinforced structures located at coastal areas has an effective role in sustainable development. Therefore, accurate prediction of the corrosion rate of these structures is very important. In the first step, Particle Swarm Optimization with Dragonfly Algorithm (PSO–DA) is introduced to enhance stochastic behavior using the Levy flight. In the second step, the PSO–DA algorithm is utilized to optimize Adaptive-Network-Based Fuzzy Inference System (ANFIS). Finally, a novel hybrid model is proposed for forecasting corrosion current density, based on the combination of PSO–DA and ANFIS.

A real test case dataset including 2460 cases collected from 37 regions in the Persian Gulf is considered to investigate the effectiveness of the proposed hybrid modeling technique. The input parameters of the model included the concrete characteristics and the operational condition of the structure. Different meta-heuristic optimization techniques (i.e., GA, PSO, HPSO, IGA, PSO–DA, BBO, and Bat) are used to predict C_{11} as a criterion of corrosion rate. From the experiment results, the following conclusions are drawn.

1. The convergence curve and trajectory for 10 Unimodal and Multimodal problems proved the PSO–DA can converge to the global optimal and escape from local optimal;
2. The proposed method (PSO–DA) can improve the performance of ANFIS and find the optimal values of adaptive modeling parameters;
3. The optimal ANFIS topology based on PSO–DA can provide a balance between the exploitation/exploration abilities through modeling and optimizing;
4. Comparison of the prediction results of the employed models shows that the predictive performance of the proposed model is high and the error level is low ($R^2 = 0.92$, MAPE = 1.67, MEDAE = 0.14, EF = 0.97);
5. The proposed model fulfills the goal of the present study to predict the corrosion current density of reinforced concrete structures in the coastal areas by reducing the error of generalization and optimizing the performance of the ANFIS model;
6. The efficient rules are constructed by various combinations of linguistic terms for the prediction of corrosion current density;
7. A useful model was proposed by as new approach to the current international corrosion management models;
8. Scientific and industrial aspects were considerd to predict the corrosion current density in marine concrete structures;
9. The sensitivity analysis revealed that C_5, C_{10}, C_8, C_7, C_6, C_2, C_9, C_3, C_1, and C_4 are the effective factors on the C_{11} in reinforced concrete structures. Knowing the extent and order of effectiveness of the parameters, played an important role in improving the performance of such structures.
10. Proposed rules can be used as criteria for risk based inspection, enhanced the documentation and data management, and considered as a new approach to expressing the importance of recording quality with industrial data and documentation.

For future work, parallel computing will be applied in the ANFIS model to reduce the computational time. Also, we would like to consider the impact of other meta-heuristics algorithms in ANFIS training. Further research to be undertaken to use a complex machine learning approach (e.g., deep learning) for corrosion rate prediction.

Acknowledgements
Not applicable.

Authors' contributions
GRK designed the model and the computational framework and analyzed the data. ZR performed literature survey regarding RC structure. GKH and ZR wrote the manuscript. ME supervised the work. HB performed the measurements. All authors read and approved the final manuscript.

Authors' information
Gholam Reza Khayati is an Associate Professor in the School of Materials Science and Engineering at Shahid Bahonar University. Zahra Rajabi is a Ph.D. student in the school of Materials Science and Engineering at Shahid Bahonar University. Maryam Ehteshamzadeh is a Full Professor in the School of Materials Science and Engineering at Shahid Bahonar University. Hadi Beirami is researcher in Metalnastri Anticorrosion Systems, Via Padova, 3, 20063, Cernusco sul Naviglio, Milano, Italy.

Funding
This project was supported by Niroo Research institute (NRI), info@nri.ac.ir.

Availability of data and materials
The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Department of Materials Science and Engineering, Shahid Bahonar University of Kerman, P. O. Box No, 76135-133 Kerman, Iran. [2]Metalnastri Anticorrosion Systems, Via Padova, 3, Cernusco Sul Naviglio, 20063 Milano, Italy.

Received: 25 August 2021 Accepted: 27 February 2022
Published online: 02 June 2022

References

Alarifi, I. M., Nguyen, H. M., Bakhtiyari, A. N., & Asadi, A. (2019). Feasibility of ANFIS-PSO and ANFIS-GA models in predicting thermophysical properties of Al2O3-MWCNT/Oil hybrid nanofluid. *Materials, 12*(21), 3628. https://doi.org/10.3390/ma12213628

Alhajj, M. A., Palma-lopes, S., & Villain, G. (2019). Accounting for steel rebar effect on resistivity profiles in view of reinforced concrete structure survey. *Construction and Building Materials, 223*, 898–909. https://doi.org/10.1016/j.conbuildmat.2019.07.208

Angst, U. M. (2018). Challenges and opportunities in corrosion of steel in concrete. *Materials and Structures, 51*(1), 4. https://doi.org/10.1617/s11527-017-1131-6

Bahiraei, M., Nazari, S., & Safarzadeh, H. (2021). Modeling of energy efficiency for a solar still fitted with thermoelectric modules by ANFIS and PSO-enhanced neural network: A nano fluid application. *Powder Technology, 385*, 185–198. https://doi.org/10.1016/j.powtec.2021.03.001

Basser, H., Karami, H., Shamshirband, S., Akib, S., Amirmojahedi, M., Ahmad, R., et al. (2015). Hybrid ANFIS-PSO approach for predicting optimum parameters of a protective spur dike. *Applied Soft Computing, 30*, 642–649. https://doi.org/10.1016/j.asoc.2015.02.011

Beirami, H., & Ehteshamzadeh, M. (2016). Applying cathodic prevention to electric transmission tower foundations. *Materials Performance, 55*(10), 30–33.

Bemani, A., Baghban, A., Shamshirband, S., Mosavi, A., Csiba, P., & Várkonyi-Kóczy, A. R. (2020). Applying ANN, ANFIS, and LSSVM models for estimation of acid solvent solubility in supercritical CO2. *Computers, Materials & Continua, 63*, 1175–1204. https://doi.org/10.32604/cmc.2020.07723

Benzaouia, A., & El Hajjaji, A. (2014). Introduction to Takagi-Sugeno fuzzy systems. *Advanced Takagi-Sugeno fuzzy systems* (pp. 1–40). Springer.

Cai, B., Xu, L. F., & Fu, F. (2019). Shear resistance prediction of post-fire reinforced concrete beams using artificial neural network. *International Journal of Concrete Structures and Materials, 13*(1), 46. https://doi.org/10.1186/s40069-019-0358-8

Cai, R., Han, T., Liao, W., Huang, J., Li, D., Kumar, A., & Ma, H. (2020). Prediction of surface chloride concentration of marine concrete using ensemble machine learning. *Cement and Concrete Research, 136*, 106164. https://doi.org/10.1016/j.cemconres.2020.106164

Chen, K., Zhou, F., Yin, L., Wang, S., Wang, Y., & Wan, F. (2018). A hybrid particle swarm optimizer with sine cosine acceleration coefficients. *Information Sciences, 422*, 218–241. https://doi.org/10.1016/j.ins.2017.09.015

Chung, L., Hur, M. W., & Park, T. (2018). Performance evaluation of CFRP reinforced concrete members utilizing fuzzy technique. *International Journal of Concrete Structures and Materials, 12*(1), 78. https://doi.org/10.1186/s40069-018-0313-0

Doi, K., Hiromoto, S., Shinohara, T., Tsuchiya, K., Katayama, H., & Akiyama, E. (2020). Role of mill scale on corrosion behavior of steel rebars in mortar. *Corrosion Science, 177*, 108995. https://doi.org/10.1016/j.corsci.2020.108995

Garcés, P., Saura, P., Zornoza, E., & Andrade, C. (2011). Influence of pH on the nitrite corrosion inhibition of reinforcing steel in simulated concrete pore solution. *Corrosion Science, 53*(12), 3991–4000. https://doi.org/10.1016/j.corsci.2011.08.002

Garg, H. (2015). A hybrid GA-GSA algorithm for optimizing the performance of an industrial system by utilizing uncertain data. In P. Vasant (Ed.), *Handbook of research on artificial intelligence techniques and algorithms* (pp. 620–654). IGI Global. https://doi.org/10.4018/978-1-4666-7258-1.ch020

Garg, H. (2016). A hybrid PSO-GA algorithm for constrained optimization problems. *Applied Mathematics and Computation, 274*, 292–305. https://doi.org/10.1016/j.amc.2015.11.001

Garg, H. (2018). A hybrid GSA-GA algorithm for constrained optimization problems. *Information Sciences, 478*, 499–523. https://doi.org/10.1016/j.ins.2018.11.041

Griffin, J. (2013). *The Sine Map*, (3), 1–9. Retrieved from https://people.maths.bris.ac.uk/~macpd/ads/sine.pdf

Guo, M. W., Wang, J. S., Zhu, L. F., Guo, S. S., & Xie, W. (2020). Improved ant lion optimizer based on spiral complex path searching patterns. *IEEE Access, 8*, 22094–22126. https://doi.org/10.1109/ACCESS.2020.2968943

Hamidane, H., Chateauneuf, A., Messabhia, A., & Ababneh, A. (2020). Reliability analysis of corrosion initiation in reinforced concrete structures subjected to chlorides in presence of epistemic uncertainties. *Structural Safety, 86*, 101976. https://doi.org/10.1016/j.strusafe.2020.101976

Hassanat, A., Almohammadi, K., Alkafaween, E., & Abunawas, E. (2019). Choosing mutation and crossover ratios for genetic algorithms—A review with a new dynamic approach. *Information, 10*, 390. https://doi.org/10.3390/info10120390

Hayyolalam, V., Asghar, A., & Kazem, P. (2020). Black widow optimization algorithm: A novel meta-heuristic approach for solving engineering optimization problems ☆. *Engineering Applications of Artificial Intelligence, 87*, 103249. https://doi.org/10.1016/j.engappai.2019.103249

Heo, S. J., Chunwei, Z., & Yu, E. (2018). Response simulation, data cleansing and restoration of dynamic and static measurements based on deep learning algorithms. *International Journal of Concrete Structures and Materials, 12*(1), 82. https://doi.org/10.1186/s40069-018-0316-x

Hong, W. C., Dong, Y., Zhang, W. Y., Chen, L. Y., & Panigrahi, K. B. (2013). Cyclic electric load forecasting by seasonal SVR with chaotic genetic algorithm. *International Journal of Electrical Power and Energy Systems, 44*(1), 604–614. https://doi.org/10.1016/j.ijepes.2012.08.010

Hong, W., Li, M., Geng, J., & Zhang, Y. (2019). Novel chaotic bat algorithm for forecasting complex motion of floating platforms. *Applied Mathematical Modelling, 72*, 425–443. https://doi.org/10.1016/j.apm.2019.03.031

Hornbostel, K., Larsen, C. K., & Geiker, M. R. (2013). Relationship between concrete resistivity and corrosion rate—A literature review. *Cement and Concrete Composites, 39*, 60–72. https://doi.org/10.1016/j.cemconcomp.2013.03.019

Karaboga, D., & Kaya, E. (2019a). Adaptive network based fuzzy inference system (ANFIS) training approaches: A comprehensive survey. *Artificial Intelligence Review, 52*(4), 2263–2293. https://doi.org/10.1007/s10462-017-9610-2

Karaboga, D., & Kaya, E. (2019b). Adaptive network based fuzzy inference system (ANFIS) training approaches: A comprehensive survey. *Artificial Intelligence Review, 52*, 2263–2293. https://doi.org/10.1007/s10462-017-9610-2

Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: Past, present, and future. *Multimedia Tools and Applications, 80*, 8091–8126. https://doi.org/10.1007/s11042-020-10139-6

Khunkitti, S., Siritaratiwat, A., Premrudeepreechacharn, S., Chatthaworn, R., & Watson, N. R. (2018). A hybrid DA-PSO optimization algorithm for multiobjective optimal power flow problems. *Energies, 11*(9), 2270. https://doi.org/10.3390/en11092270

Lendek, Z., Guerra, T. M., Babuška, R., & De Schutter, B. (2010). Takagi-Sugeno fuzzy models. *Stability Analysis and nonlinear observer design using Takagi-Sugeno fuzzy models* (pp. 5–24). Springer.

Li, C., Chen, Q., Wang, R., Wu, M., & Jiang, Z. (2020). Corrosion assessment of reinforced concrete structures exposed to chloride environments in underground tunnels: Theoretical insights and practical data interpretations. *Cement and Concrete Composites, 112*, 103652. https://doi.org/10.1016/j.cemconcomp.2020.103652

Liang, J., Qu, B., & Suganthan, P. (2013). *Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization; technical report*. Zhengzhou University and Nanyang Technological University.

Liu, Y., Pang, S., & Chen, D. (2013). An unusual chaotic system and its control. *Mathematical and Computer Modelling, 57*(9–10), 2473–2493. https://doi.org/10.1016/j.mcm.2012.12.006

Medeiros, M. H. F., Gobbi, A., Réus, G. C., & Helene, P. (2013). Reinforced concrete in marine environment: Effect of wetting and drying cycles, height and positioning in relation to the sea shore. *Construction and Building Materials, 44*, 452–457. https://doi.org/10.1016/j.conbuildmat.2013.02.078

Melchers, R. E. (2020). Modelling durability of reinforced concrete structures. *Corrosion Engineering Science and Technology, 55*(2), 171–181. https://doi.org/10.1080/1478422X.2019.1710660

Metawa, N., Hassan, M. K., & Elhoseny, M. (2017). Genetic algorithm based model for optimizing bank lending decisions. *Expert Systems with Applications, 80*, 75–82. https://doi.org/10.1016/j.eswa.2017.03.021

Mirjalili, S. (2019). Genetic algorithm. Evolutionary algorithms and neural networks. In *Studies in computational intelligence* (Vol. 780, pp. 43–55). Springer. https://doi.org/10.1007/978-3-319-93025-1

Moayedi, H., Raftari, M., Sharifi, A., Jusoh, W. A. W., & Rashid, A. S. A. (2020). Optimization of ANFIS with GA and PSO estimating α ratio in driven piles. *Engineering with Computers, 36*(1), 227–238. https://doi.org/10.1007/s00366-018-00694-w

Molina, F. J., Alonso, C., & Andrade, C. (1993). Cover cracking as a function of rebar corrosion: Part 2-Numerical model. *Materials and Structures, 26*(9), 532–548. https://doi.org/10.1007/BF02472864

Muthulingam, S., & Rao, B. N. (2014). Non-uniform time-to-corrosion initiation in steel reinforced concrete under chloride environment. *Corrosion Science, 82*, 304–315. https://doi.org/10.1016/j.corsci.2014.01.023

Otieno, M., Ikotun, J., & Ballim, Y. (2019). Experimental investigations on the influence of cover depth and concrete quality on time to cover cracking due to carbonation-induced corrosion of steel in RC structures in an urban, inland environment. *Construction and Building Materials, 198*, 172–181. https://doi.org/10.1016/j.conbuildmat.2018.11.215

Patwal, R. S., Narang, N., & Garg, H. (2018). A novel TVAC-PSO based mutation strategies algorithm for generation scheduling of pumped storage hydrothermal system incorporating solar units. *Energy, 142*, 822–837. https://doi.org/10.1016/j.energy.2017.10.052

Paul, S. C., Panda, B., Huang, Y., Garg, A., & Peng, X. (2018). An empirical model design for evaluation and estimation of carbonation depth in concrete. *Measurement Journal of the International Measurement Confederation, 124*, 205–210. https://doi.org/10.1016/j.measurement.2018.04.033

Pellegrini, R., Serani, A., Liuzzi, G., Rinaldi, F., Lucidi, S., & Diez, M. (2020). Hybridization of multi-objective deterministic particle swarm with derivative-free local searches. *Mathematics, 8*, 546. https://doi.org/10.3390/math8040546

Pour-Ghaz, M., Isgor, O. B., & Ghods, P. (2009). The effect of temperature on the corrosion of steel in concrete. Part 1: Simulated polarization resistance tests and model development. *Corrosion Science, 51*(2), 415–425. https://doi.org/10.1016/j.corsci.2008.10.034

Pousinho, H. M. I., Mendes, V. M. F., & Catalão, J. P. S. (2012). Short-term electricity prices forecasting in a competitive market by a hybrid PSO-ANFIS approach. *International Journal of Electrical Power and Energy Systems, 39*, 29–35. https://doi.org/10.1016/j.ijepes.2012.01.001

Prügel-Bennetf, A. (2001). The mixing rate of different crossover operators. In W. M. S. Worthy & N. Martin (Eds.), *Foundations of genetic algorithms 6* (pp. 261–274). Morgan Kaufmann. https://doi.org/10.1016/B978-1558607347-7/50097-4

Qiao, G., Hong, Y., Ou, J., & Guan, X. (2015). Corrosion monitoring of the RC structures in time domain: Part II. Recognition algorithm based on fractional derivative theory. *Measurement: Journal of the International Measurement Confederation, 67*, 84–91. https://doi.org/10.1016/j.measurement.2014.12.048

Rahimzadeh, H., Sadeghi, M., & Ahmad, S. (2021). Chemometrics and Intelligent Laboratory Systems Modifying genetic algorithm by dynamic memory and solution reconstructing mechanism for selectivity control of chemical sensors. *Chemometrics and Intelligent Laboratory Systems, 214*(April), 104332. https://doi.org/10.1016/j.chemolab.2021.104332

Rodrigues, R., Gaboreau, S., Gance, J., Ignatiadis, I., & Betelu, S. (2020). Reinforced concrete structures: A review of corrosion mechanisms and advances in electrical methods for corrosion monitoring. *Construction and Building Materials, 269*, 121240. https://doi.org/10.1016/j.conbuildmat.2020.121240

Ryu, E., Kang, J., Lee, J., Shin, Y., & Kim, H. (2020). Automated detection of surface cracks and numerical correlation with thermal-structural behaviors of fire damaged concrete beams. *International Journal of Concrete Structures and Materials, 14*(1), 12. https://doi.org/10.1186/s40069-019-0387-3

Salami, B. A., Rahman, S. M., Oyehan, T. A., Maslehuddin, M., & Dulaijan, S. U. A. (2020). Ensemble machine learning model for corrosion initiation time estimation of embedded steel reinforced self-compacting concrete. *Measurement, 165*, 108141. https://doi.org/10.1016/j.measurement.2020.108141

Salleh, M. N. M., & Hussain, K. (2016). A review of training methods of ANFIS for applications in business and economics. *International Journal of u- and e- Service, Science and Technology, 9*(7), 165–172. https://doi.org/10.14257/ijunesst.2016.9.7.17

Sarkheyli, A., Zain, A. M., & Sharif, S. (2015). Robust optimization of ANFIS based on a new modified GA. *Neurocomputing, 166*, 357–366. https://doi.org/10.1016/j.neucom.2015.03.060

Serani, A, Diez, M., Campana, E. F., Fasano, G., Peri, D., & Iemma, U. (2015). Globally convergent hybridization of particle swarm optimization using line search-based derivative-free techniques. In X. S. Yang (Eds.) Recent advances in swarm intelligence and evolutionary computation. In *Studies in computational intelligence* (Vol. 585, pp. 25–47). Springer. https://doi.org/10.1007/978-3-319-13826-8

Serani, A., Leotardi, C., Iemma, U., Campana, E. F., Fasano, G., & Diez, M. (2016). Parameter selection in synchronous and asynchronous deterministic particle swarm optimization for ship hydrodynamics problems. *Applied Soft Computing Journal, 49*, 313–334. https://doi.org/10.1016/j.asoc.2016.08.028

Shafaei, A., & Khayati, G. R. (2020). A predictive model on size of silver nanoparticles prepared by green synthesis method using hybrid artificial neural network-particle swarm optimization algorithm. *Measurement, 151*, 107199. https://doi.org/10.1016/j.measurement.2019.107199

Somu, N., GR, M. R., & Ramamritham, K. (2020). A hybrid model for building energy consumption forecasting using long short term memory networks. *Applied Energy, 261*, 114131. https://doi.org/10.1016/j.apenergy.2019.114131

Song, Y., Wang, F., & Chen, X. (2019). An improved genetic algorithm for numerical function optimization. *Applied Intelligence, 49*(5), 1880–1902. https://doi.org/10.1007/s10489-018-1370-4

Song, Z., & Kusiak, A. (2010). Multiobjective optimization of temporal processes. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (cybernetics), 40*(3), 845–856. https://doi.org/10.1109/TSMCB.2009.2030667

Stewart, M. G., & Mullard, J. A. (2007). Spatial time-dependent reliability analysis of corrosion damage and the timing of first repair for RC structures. *Engineering Structures, 29*(7), 1457–1464. https://doi.org/10.1016/j.engstruct.2006.09.004

Sun, X., Kong, H., Wang, H., & Zhang, Z. (2018). Evaluation of corrosion characteristics and corrosion effects on the mechanical properties of reinforcing steel bars based on three-dimensional scanning. *Corrosion Science, 142*, 284–294. https://doi.org/10.1016/j.corsci.2018.07.030

Taffese, W. Z., & Sistonen, E. (2017). Machine learning for durability and service-life assessment of reinforced concrete structures: Recent advances and future directions. *Automation in Construction, 77*, 1–14. https://doi.org/10.1016/j.autcon.2017.01.016

Tang, X., Zhuang, L., Cai, J., & Li, C. (2010). Multi-fault classification based on support vector machine trained by chaos particle swarm optimization. *Knowledge-Based Systems, 23*(5), 486–490. https://doi.org/10.1016/j.knosys.2010.01.004

Topçu, I. B., Boğa, A. R., & Hocaoğlu, F. O. (2009). Modeling corrosion currents of reinforced concrete using ANN. *Automation in Construction, 18*(2), 145–152. https://doi.org/10.1016/j.autcon.2008.07.004

Umrao, R. K., Sharma, L. K., Singh, R., & Singh, T. N. (2018). Determination of strength and modulus of elasticity of heterogenous sedimentary rocks: An ANFIS predictive technique. *Measurement, 126*, 194–201. https://doi.org/10.1016/j.measurement.2018.05.064

Wang, X., Guan, N., Zhao, H., Wang, S., & Zhang, Y. (2020). A new image encryption scheme based on coupling map lattices with mixed multi-chaos. *Scientific Reports, 10*(1), 9784. https://doi.org/10.1038/s41598-020-66486-9

Wang, Y., Liu, C., Wang, Y., Li, Q., & Yan, B. (2019). Semi-empirical prediction model of chloride-induced corrosion rate in uncracked reinforced concrete exposed to a marine environment. *Electrochimica Acta, 331*, 135376. https://doi.org/10.1016/j.electacta.2019.135376

Yu, F., & Author, A. (2016). Improved Roulette wheel selection-based genetic algorithm for TSP. In *International conference on network and information systems for computers* (pp. 151–154). IEEE. https://doi.org/10.1109/ICNISC.2016.22

Zhang, J., & Lounis, Z. (2006). Sensitivity analysis of simplified diffusion-based corrosion initiation model of concrete structures exposed to chlorides. *Cement and Concrete Research, 36*(7), 1312–1323. https://doi.org/10.1016/j.cemconres.2006.01.015

Zhao, Y., Yu, J., & Jin, W. (2011). Damage analysis and cracking model of reinforced concrete structures with rebar corrosion. *Corrosion Science, 53*(10), 3388–3397. https://doi.org/10.1016/j.corsci.2011.06.018

Zhou, H., Chen, S., Zhou, Y., Lin, Z., Liang, X., Liu, J., & Xing, F. (2020). Field test of a reinforced concrete bridge under marine environmental corrosion. *Engineering Failure Analysis, 115*, 104669. https://doi.org/10.1016/j.engfailanal.2020.104669

Anoop, M. B., Rao, K. B., & Rao, T. V. S. R. A. (2002). Application of fuzzy sets for estimating service life of reinforced concrete structural members in corrosive environments. Eng Struct, 24(9), 1229–1242. https://doi.org/10.1016/S0141-0296(02)00060-3

Dey, A., Miyani, G., & Sil, A. (2019). Application of artificial neural network (ANN) for estimating reliable service life of reinforced concrete (RC) structure bookkeeping factors responsible for deterioration mechanism. Soft Comput, 24, 2109–2123. https://doi.org/10.1007/s00500-019-04042-y

Lv, Y., Wang, J., Wang, J., Xiong, C., Zou, L., Li, L., & Li, D. (2020). Steel corrosion prediction based on support vector machines. Chaos, Solitons and Fractals, 136, 109807. https://doi.org/10.1016/j.chaos.2020.109807

Parthiban, T., Ravi, R., Parthiban, G. T., Srinivasan, S., Ramakrishnan, K. R., & Raghavan, M. (2005). Neural network analysis for corrosion of steel in concrete. Corrosion Science, 47(7), 1625–1642. https://doi.org/10.1016/j.corsci.2004.08.011

Roxas, C. L. C., & Lejano, B. A. (2019). An artificial neural network model for the corrosion current denstiy of steel in mortar mixed with Seawater. Int J Geomate, 16(56), 79–84. https://doi.org/10.21660/2019.56.4585

Sadowski, L., & Nikoo, M. (2014). Corrosion current density prediction in reinforced concrete by imperialist competitive algorithm. Neural Comput & Applic, 25, 1627–1638. https://doi.org/10.1007/s00521-014-1645-6

Publisher's Note