Green Energy and Intelligent Transportation 2 (2023) 100040

Contents lists available at ScienceDirect

Green Energy and Intelligent Transportation

journal homepage: www.journals.elsevier.com/green-energy-and-intelligent-transportation

Full length article

Multiple learning neural network algorithm for parameter estimation of proton exchange membrane fuel cell models

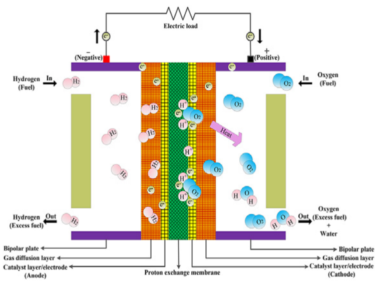
Yiying Zhang [a,b], Chao Huang [b], Hailong Huang [c,*], Jingda Wu [d]

[a] *School of Electrical and Information Engineering, Jiangsu University, China*
[b] *Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong*
[c] *Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong*
[d] *School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore*

HIGHLIGHTS

- Multiple learning neural network algorithm (MLNNA) is proposed.
- The performance of MLNNA is verified by the challenging CEC 2015 test suite.
- The performance of MLNNA is evaluated by two typical fuel cell models.
- The superiority of MLNNA on parameter extraction of fuel cell models is proven.

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Neural network algorithm
Parameter extraction
Proton exchange membrane fuel cell
Metaheuristics

ABSTRACT

Extracting the unknown parameters of proton exchange membrane fuel cell (PEMFC) models accurately is vital to design, control, and simulate the actual PEMFC. In order to extract the unknown parameters of PEMFC models precisely, this work presents an improved version of neural network algorithm (NNA), namely the multiple learning neural network algorithm (MLNNA). In MLNNA, six learning strategies are designed based on the created local elite archive and global elite archive to balance exploration and exploitation of MLNNA. To evaluate the performance of MLNNA, MLNNA is first employed to solve the well-known CEC 2015 test suite. Experimental results demonstrate that MLNNA outperforms NNA on most test functions. Then, MLNNA is used to extract the parameters of two PEMFC models including the BCS 500 W PEMFC model and the NedStack SP6 PEMFC model. Experimental results support the superiority of MLNNA in the parameter estimation of PEMFC models by comparing it with 10 powerful optimization algorithms.

* Corresponding author.
E-mail addresses: zhangyiying@ujs.edu.cn (Y. Zhang), hchao.huang@polyu.edu.hk (C. Huang), hailong.huang@polyu.edu.hk (H. Huang), jingda001@e.ntu.edu.sg (J. Wu).

https://doi.org/10.1016/j.geits.2022.100040
Received 26 July 2022; Received in revised form 13 September 2022; Accepted 4 October 2022
Available online 21 October 2022

Nomenclature

H_2	Hydrogen
H^+	Positive proton
e^-	Negative electron
O_2	Oxygen
H_2O	Water
E_{nernst}	Cell reversible voltage
V_{act}	Activation voltage loss
V_{ohm}	Ohmic voltage loss
V_{con}	Concentration voltage loss
n	The number of cells
V_{stack}	Output voltage
T	Operating cell temperature
P_{H_2}	Partial pressure of hydrogen
P_{O_2}	Partial pressure of oxygen
R_{ha}	Relative humidity of vapour at the anode
R_{hc}	Relative humidity of vapour at the cathode
P_{H_2O}	Saturation pressure of the water
P_{a}	Inlet pressure at the anode
P_{c}	Inlet pressure at the cathode
A	Membrane area
I	Operating current
C_{O_2}	Concentration of oxygen
ξ	Empirical coefficient
R_{m}	Membrane resistance
R_{c}	Connections resistance
l	Thickness of the membrane
p_{m}	Resistivity of the membrane
β	Parametric coefficient
J	Actual current density
J_{max}	Maximum current density
V_{mea}	Measured voltage
V_{cal}	Calculated voltage
t	The number of iterations
N	Population size
$\boldsymbol{w}_i^t$	Weight vector of individual *i*
$\boldsymbol{w}_g^t$	Optimal weight vector
$\boldsymbol{X}^t$	Population
$\boldsymbol{x}_i^t$	Position of individual *i*
$\boldsymbol{g}^t$	The current optimal solution
$\boldsymbol{v}_i^t$	Trail vector of individual *i*
$\boldsymbol{X}_{\mathrm{L}}^t$	Local elite archive
$\boldsymbol{X}_{\mathrm{G}}^t$	Global elite archive
ρ_{BP}	Selection probability
ρ_{TO}	Selection probability
T_{max}	The maximum number of iterations
PEMFC	Proton exchange membrane fuel cell
PEMFCs	Proton exchange membrane fuel cells
FC	Fuel cell
FCs	Fuel cells
NFL	No free lunch
NNA	Neural network algorithm
WOA	Whale optimization algorithm
SOA	Seagull optimization algorithm
TSO	Transient search algorithm
BSA	Backtracking search algorithm
TSA	Tunicate swarm algorithm
JAYA	Jaya algorithm
STOA	Sooty tern optimization algorithm
SCA	Sine cosine algorithm
MLNNA	Multiple learning neural network algorithm

1. Introduction

Under the background of the aggravation of urban air pollution, more and more researchers focus on generating efficient and green electricity through alternative energy resources. The fuel cell is regarded as one of the promising alternative energy resources [1–3]. Energy resources driven by fuel cells (FCs) are clean and efficient and produce electricity based on a series of electro–chemical reactions [4–6]. A FC system usually consists of three major components, which are cathode, anode, and electrolyte [7–9]. According to the electrolyte material, FCs can be divided into many types, such as alkaline FCs, proton exchange membrane FCs, phosphoric acid FCs, and solid oxide FCs [10]. Note that, among the various types of FCs, proton exchange membrane fuel cell (PEMFC) has received increasing attention due to the following three advantages: (1) PEMFC can generate electricity directly based on the reactions between hydrogen and oxygen, which doesn't produce harmful emissions [4]; (2) PEMFC usually operates in low-temperature environments, which is suitable for portable energy utilities [11]; (3) PEMFC has high energy efficiency and can produce power of high densities, which is appropriate for automotive utilities [12].

However, PEMFC also faces two challenges. On one hand, the catalyst in the electro–chemical reactions is expensive, which will increase its cost and harms its commercial competitiveness [13]. On the other hand, its output voltage is unregulated due to the operating losses, which include activation losses, ohmic losses, and concentration losses. This reflects the following three aspects [14–16]: (1) activation losses lead to the high decay of output voltage; (2) ohmic losses cause the decrease linearly of output voltage; (3) concentration losses can result in the decrease rapidly of the output voltage at higher loads. Given the two challenges, modelling PEMFC characteristics based on some nonlinear differential equations plays a vital role in the design, simulation, evaluation, analysis, and development of high-efficiency PEMFC systems [4,17].

At present, many modelling approaches for PEMFC have been proposed, such as mechanistic models [18], empirical models [19], and semi-empirical models [20]. Among the reported PEMFC models, mathematical modelling derived from semi-empirical equations developed by Mann et al. [21–23] is very popular and has been widely used for forecasting the polarization characteristics of PEMFC under various operating conditions. Although Mann's PEMFC model is very popular, several empirical parameters in the differential evolutions of this model are unknown. To achieve a precise PEMFC model, it is necessary to accurately estimate these unknown parameters.

Metaheuristic methods have three remarkable advantages compared with traditional numerical methods, which are population-based optimization technique, randomness, and insensitivity to the initial solutions. Benefiting from the three advantages, using metaheuristic methods to estimate the unknown parameters of PEMFC models has been a hot research topic in recent years. Hegazy et al. [8] adopted the bald eagle search algorithm to estimate the parameters of the BCS 500 W PEMFC model and NedStack PS6 PEMFC model. Xu et al. [24] designed a hybrid optimization method based on Jaya algorithm and Nelder-Mead simplex approach for three PEMFCs estimation cases including two with seven unknown parameters and one with nine unknown parameters. Sultan et al. [25] identified the parameters of four types of PEMFC models including 250 W stack PEMFC model, BCS 500 W PEMFC model, AVISTA SR-12 500 W PEMFC model, and Temasek 1000 W stack PEMFC model by the improved salp swarm algorithm. Rizk et al. [26] utilized an improved artificial ecosystem optimizer to identify the parameters of three types of PEMFCs, i.e. BCS 500 W PEMFC model, 250 W stack PEMFC model, and the NedStack PS6 PEMFC model. Duan et al. [27] employed a stain bowerbird optimizer to identify the parameters of four PEMFC models, i.e. Ballard V 5,000 W PEMFC model, SR-12 500 W PEMFC model, BCS 500 W PEMFC model, and Temasek 1000 W stack PEMFC model. Lu et al. [28] extracted the parameters of the NedStack PS6 PEMFC model and the Nexa PEMFC model by an improved crow search algorithm. The parameters of the BCS 500 W PEMFC model, 500 W SR-12 PEMFC model, and 250 W stack PEMFC model were estimated by atom search optimizer and harris hawks optimization in [29]. The parameters of the Ballard Mark V PEMFC model, Horizon H-12 PEMFC model, and the NedStack PS6 PEMFC model were computed by an adaptive sparrow search algorithm in [30]. The parameters of the 250 MW stack PEMFC model and the NedStack PS6 PEMFC model were evaluated by a modified monarch butterfly optimization in [31]. Diab et al. [32] estimated the parameters of the 250 W stack PEMFC model, SR-12 500 W PEMFC model, and 250 W stack PEMFC model, by marine predators algorithm and political optimizer. Seleem et al. [33] used an equilibrium optimizer to extract the parameters of the NedStack PS6 PEMFC model. Parameters extraction of Ballard Mark V PEMFC model, AVISTA SR-12 PEMFC model, and 250 W stack PEMFC model was addressed by whale optimization algorithm in [22]. Yang et al. [34] proposed an improved barnacles mating optimization algorithm to identify the parameters of the Horizon 500 W PEMFC model and NedStack PS6 PEMFC model. Although many achievements have been made in the field of parameter extraction of PEMFCs, more research is still necessary due to the following reasons:

1) Parameter precision. Parameter precision is closely associated with the quality of modelling PEMFCs. Improving the estimation accuracy of the unknown parameters is an extremely effective way of achieving highly accurate PEMFC models;
2) No-free-lunch (NFL) theorem [35–37]. In general, it is a common approach that parameter estimation of the PEMFC model is converted into an optimization problem to be solved. According to the basic idea of the NFL theorem, an optimization algorithm can't beat the other optimization algorithms on all optimization problems. Therefore, developing more efficient optimization algorithms to extract the parameters of PEMFC models with different characteristics has been a hot research direction.

Motivated by the mentioned two reasons, a new variant of neural network algorithm (NNA) [38], namely multiple learning neural network algorithm (MLNNA), is presented for parameter extraction of PEMFC models. NNA is inspired by artificial neural networks and has a very simple structure, which employs bias operator and transfer operator to balance its global search ability and local search ability. The bias operator plays a similar role to the bias current. The transfer operator is to pull one individual toward the direction of the current optimal solution. In view of the unique structure of the artificial neural network, NNA shows outstanding global search ability. In addition, NNA needs two parameters including population size and terminal condition. Further, the two parameters are required for every population-based optimization algorithm. However, NNA has a slow convergence rate and tends to trap into the local optimal solutions for complex nonlinear optimization problems. To enhance the global search ability of NNA, two elite archives are introduced. On one hand, a local elite archive used to save historical best solutions of each individual is built, which is employed to guide bias operator and transfer operator. On the other hand, a global elite archive used to save historical elite solutions of the population is created, whose task is to guide the transfer operator. In addition, to enhance the global search ability and keep the population diversity of MLNNA, six candidate learning strategies based on the introduced two elite archives are designed for bias operator and transfer operator in the proposed MLNNA. The main contributions of this work are stated below:

1) MLNNA is presented for parameter identification of PEMFC models. MLNNA only requires the essential parameters, i.e. population size and terminal condition, for optimization. Thus, it is very easy for MLNNA to be applied to different types of PEMFC models;
2) Local elite archive and global elite archive are introduced to guide the search directions of bias operator and transfer operator, respectively;
3) The optimization performance of MLNNA is investigated by the well-known CEC 2015 test suite consisting of 15 challenging benchmark test functions;
4) MLNNA is employed for parameter estimation of two different types of PEMFC models, i.e. the BCS 500 W PEMFC model and the NedStack PS6 PEMFC model.

The rest of this work is organized as follows. Section 2 introduces the problem formulation of the PEMFC model. Section 3 presents the proposed MLNNA. Section 4 checks the performance of MLNNA for numerical problems. The performance of MLNNA for parameter extraction of PEMFC models is evaluated in Section 5. Section 6 discusses how effective the improved strategies are in MLNNA. Lastly, Section 7 presents the conclusions and further work.

2. Problem statement

This section is to describe the PEMFC model, which includes three subsections. Section 2.1 presents the chemical reaction of hydrogen and oxygen. Section 2.2 derives the mathematical formulation of the PEMFC model. The objective function of the PEMFC model is formulated in Section 2.3.

2.1. Basic knowledge

As mentioned above, the basic principle of the PEMFC is that the chemical energy is converted into the electrical power by the chemical reaction between hydrogen and oxygen. The expression of the chemical reaction can be written by [11,39].

$$H_2 \rightarrow 2H^+ + 2e^- \tag{1}$$

$$2H^+ + 2e^- + \frac{1}{2}O_2 \rightarrow H_2O + \text{heat} + \text{electricity} \tag{2}$$

where H_2 is hydrogen, H^+ is a positive proton, e^- is a negative electron, O_2 is oxygen, and H_2O is water. From Eq. (1) and Eq. (2), hydrogen and oxygen are the fuel; water and electricity are the products. In addition, this reaction also generates some heat.

Fig. 1 demonstrates the schematic of the proton exchange membrane fuel cell. According to Fig. 1, the hydrogen is first divided into positive protons and negative electrons by the platinum-based electrodes. The positive protons and the negative electrons move to the cathode via the membrane (the membrane plays the role of the electrolyte) and the external circuit, respectively. The positive protons and the negative electrons are mixed with the oxygen at the cathode, which produces the water and releases heat. Note that, the moving process of the negative electrons will generate the electricity of the cell.

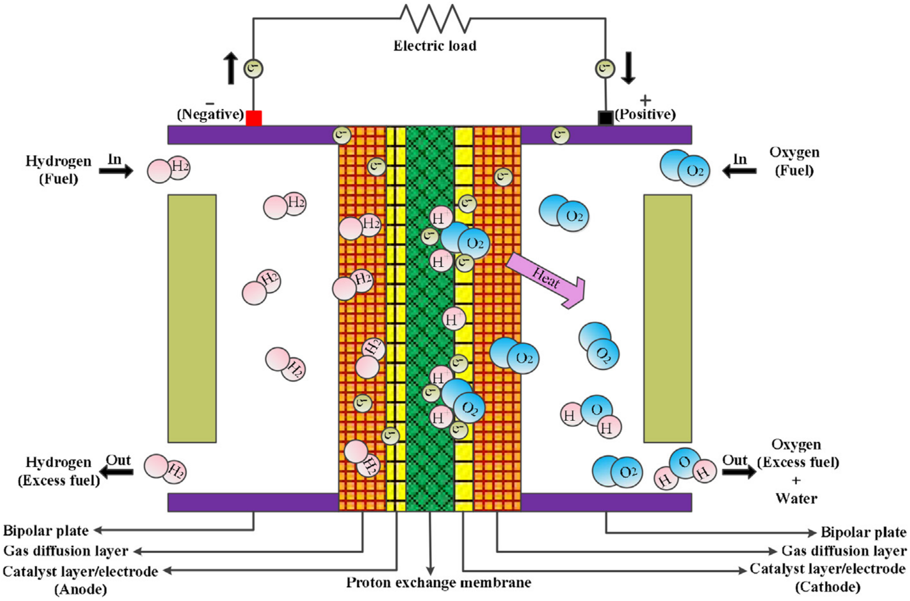


Fig. 1. The schematic of the proton exchange membrane fuel cell.

2.2. *Mathematical formulation*

The applied PEMFC model in this work is from [21], which is a combination of modelling based on mechanistic modelling and empirical modelling. For this model, the output voltage V_{cell} of a sole PEMFC is written by [4,40]

$$V_{\mathrm{cell}} = E_{\mathrm{nernst}} - V_{\mathrm{act}} - V_{\mathrm{ohm}} - V_{\mathrm{con}} \tag{3}$$

where E_{nernst} is the cell reversible voltage, V_{act} is the activation voltage loss, V_{ohm} is the ohmic voltage loss, and V_{con} is the concentration voltage loss. When n cells are connected in series to form a stack, the output voltage V_{stack} of this stack can be represented by

$$V_{\mathrm{stack}} = n \cdot V_{\mathrm{cell}} \tag{4}$$

2.2.1. *Cell reversible voltage E_{nernst}*

E_{nernst} can be computed by

$$E_{\mathrm{nernst}} = 1.229 - 0.85 \times 10^{-3}(T - 298.15) + 4.308, 5 \times 10^{-5} \times T(\ln(P_{\mathrm{H_2}}) + 0.5 \times \ln(P_{\mathrm{O_2}})) \tag{5}$$

where T is the operating cell temperature (K), $P_{\mathrm{H_2}}$ is the partial pressure of hydrogen (atm), and $P_{\mathrm{O_2}}$ is the partial pressure of oxygen (atm). $P_{\mathrm{H_2}}$ and $P_{\mathrm{O_2}}$ can be denoted by

$$P_{\mathrm{H_2}} = \frac{R_{\mathrm{ha}} \cdot P_{\mathrm{H_2O}}}{2}\left[\frac{1}{\frac{R_{\mathrm{ha}} \cdot P_{\mathrm{H_2O}}}{P_{\mathrm{a}}} \cdot \exp\left(\frac{1.635T}{A \cdot T^{1.334}}\right)} - 1\right] \tag{6}$$

$$P_{\mathrm{H_2}} = R_{\mathrm{hc}} \cdot P_{\mathrm{H_2O}}\left[\frac{1}{\frac{R_{\mathrm{hc}} \cdot P_{\mathrm{H_2O}}}{P_{\mathrm{c}}} \cdot \exp\left(\frac{4.192T}{A \cdot T^{1.334}}\right)} - 1\right] \tag{7}$$

where R_{ha} is the relative humidity of vapour at the anode, R_{hc} is the relative humidity of vapour at the cathode, $P_{\mathrm{H_2O}}$ is the saturation pressure of the water (atm), P_{a} is the inlet pressure at the anode (atm), P_{c} is the inlet pressure at the cathode (atm), and A is the membrane area (cm^2). $P_{\mathrm{H_2O}}$ can be produced by

$$P_{\mathrm{H_2O}} = 2.95 \times 10^{-2}(T - 273.15) - 9.18 \times 10^{-5}(T - 273.15)^2 + 1.44 \times 10^{-7}(T - 273.15)^3 - 2.18 \tag{8}$$

2.2.2. *Voltage activation voltage loss V_{act}*

V_{act} can be obtained by

$$V_{\mathrm{act}} = -[\xi_1 + \xi_2 T + \xi_3 T \ln(\mathrm{C_{O_2}}) + \xi_4 T \ln(I)] \tag{9}$$

where I is the operating current (A), $C_{\mathrm{O_2}}$ is the concentration of oxygen mol/cm^3, $\xi_i (i \in \{1, 2, 3, 4\})$ are empirical coefficients. $C_{\mathrm{O_2}}$ can be computed by

$$C_{\mathrm{O_2}} = \frac{P_{\mathrm{O_2}}}{5.08 \times 10^6 \times \exp\left(-\frac{498}{T}\right)} \tag{10}$$

2.2.3. *Ohmic voltage loss V_{ohm}*

V_{ohm} can be generated by

$$V_{\mathrm{ohm}} = I(R_{\mathrm{m}} + R_{\mathrm{c}}) \tag{11}$$

$$R_{\mathrm{m}} = \frac{p_{\mathrm{m}} \cdot l}{A} \tag{12}$$

where R_{m} is the membrane resistance (Ω), R_{c} is the resistance of the connection (Ω), l is the thickness of the membrane (cm), and p_{m} is the resistivity of the membrane (Ω·cm). p_{m} can be computed by

$$p_{\mathrm{m}} = \frac{181.6\left[1 + 0.03\left(\frac{I}{A}\right) + 0.062\left(\frac{T}{303}\right)^2\left(\frac{I}{A}\right)^{2.5}\right]}{\left[\lambda - 0.634 - 3\left(\frac{I}{A}\right)\right]\exp\left(4.18\frac{T-303}{T}\right)} \tag{13}$$

where λ is an adjustable parameter.

2.2.4. *Concentration voltage loss V_{con}*

V_{con} can be achieved by

$$V_{\mathrm{con}} = -\beta \ln\left(\frac{J_{\max} - J}{J_{\max}}\right) \tag{14}$$

$$J = \frac{I}{A} \tag{15}$$

where β is a parametric coefficient, J is the actual current density A/cm^2, and $J_{\max}$ is the maximum value of J A/cm^2.

2.3. Objective function

From Eqs. (3–15), to evaluate the performance of the applied PEMFC model, seven unknown parameters need to be estimated, which are ξ_1, ξ_2, ξ_3, ξ_4, λ, R_c and β. To estimate these parameters, it is a common method that parameter estimation of the PEMFC model is converted into an optimization problem. Further, the optimization problem is to minimize the total of the squared errors (TSE) between the measured voltage points and the calculated stack voltage points. Therefore, mathematically, this objective function can be expressed by

$$\begin{aligned} &\text{Minimize}(TSE) = \text{Minimize}\, f(\xi_1, \xi_2, \xi_3, \xi_4, \lambda, R_c, \beta) = \\ &\text{Minimize}\left(\sum_{i=1}^{N} [V_{\text{cal},i} - V_{\text{mea},i}]^2\right) \\ &s.t.\ \lambda_{\min} \le \lambda \le \lambda_{\max}, R_{\text{c,min}} \le R_{\text{c}} \le R_{\text{c,max}}, \beta_{\min} \le \beta \le \beta_{\max}, \xi_{j,\min} \le \xi_j \le \xi_{j,\max}, \\ &j = 1, 2, 3, 4 \end{aligned} \tag{16}$$

where N is the number of measured voltage points, $V_{\text{mea},i}$ is the measured voltage at the measured voltage point i, $V_{\text{cal},i}$ is the calculated voltage at the measured voltage point i, $\xi_{j,\min}$ is the lower boundary of ξ_j, $\xi_{j,\max}$ is the upper boundary of ξ_j, $\lambda_{\min}$ is the lower boundary of λ, $\lambda_{\max}$ is the upper boundary of λ, $R_{\text{c,min}}$ is the lower boundary of R_{c}, $R_{\text{c,max}}$ is the upper boundary of R_{c}, $\beta_{\min}$ is the lower boundary of β, and $\beta_{\max}$ is the upper boundary of β. As can be seen from Eq. (16), a smaller TSE means a more accurate PEMFC model.

3. The proposed multiple learning neural network algorithm

In this section, the proposed MLNNA is discussed. Section 3.1 presents the motivation of MLNNA. The framework of MLNNA is presented in Section 3.2. Lastly, Section 3.3 describes the implementation of MLNNA for optimization.

3.1. Motivation

Fig. 2 presents the framework of NNA. As can be seen from Fig. 2, the inspiration for NNA is from the artificial neural network and the implementation of NNA can be described as follows. In one loop, the weight matrix is first updated. Then the population is updated based on the weight matrix. Next, if the bias condition is met, the bias population and bias weight matrix will be performed; otherwise, the transfer population is executed. The obtained population and weight matrix will go to the next iteration of the loop.

In NNA, each individual $\boldsymbol{x}_i^t = [x_{i,1}^t, x_{i,2}^t, \ldots, x_{i,D}^t]$ has a weight vector $\boldsymbol{w}_i^t = [w_{i,1}^t, w_{i,2}^t, \ldots, w_{i,N}^t]$, where D is the number of variables, t is the number of iterations, and N is population size. In addition, $\boldsymbol{w}_i^t$ needs to meet the following formula:

$$\sum_{i=1}^{N} w_{i,j}^t = 1, i = 1, 2, \ldots, N, j = 1, 2, \ldots, N \tag{17}$$

$$\sum_{i=1}^{N} w_{i,j}^t = 1, i = 1, 2, \ldots, N, j = 1, 2, \ldots, N0 < w_{i,j}^t < 1, i = 1, 2, \ldots, N, j = 1, 2, \ldots, N \tag{18}$$

The weight matrix $\boldsymbol{W}^t = \{\boldsymbol{w}_1^t, \boldsymbol{w}_2^t, \ldots, \boldsymbol{w}_N^t\}$ is updated by

$$w_{i,j}^t = \left| w_{i,j}^t + 2\alpha_1 \cdot \left(w_{g,j}^t - w_{i,j}^t \right) \right| \tag{19}$$

where α_1 is a random number between 0 and 1 with uniform distribution and $\boldsymbol{w}_{g,j}^t$ is the jth variable of the optimal weight vector $\boldsymbol{w}_g^t = [\boldsymbol{w}_{g,1}^t, \boldsymbol{w}_{g,2}^t, \ldots, \boldsymbol{w}_{g,D}^t]$. Note that the indexes of optimal weight vector and optimal individual are the same. The way of weight matrix $\boldsymbol{W}^t = \{\boldsymbol{w}_1^t, \boldsymbol{w}_2^t, \ldots, \boldsymbol{w}_N^t\}$ used to update the population $\boldsymbol{X}^t = \{\boldsymbol{x}_1^t, \boldsymbol{x}_2^t, \ldots, \boldsymbol{x}_N^t\}$ can be expressed as

$$\boldsymbol{v}_i^t = \boldsymbol{x}_i^t + \boldsymbol{x}_i^t \cdot \sum_{i=1}^{N} w_{i,j}^t, i = 1, 2, \ldots, N, j = 1, 2, \ldots, N \tag{20}$$

where $\boldsymbol{v}_i^t$ is the trail vector of the ith individual.

Bias operator and transfer operator are the core components of NNA, which are employed to perform global exploration and local exploitation, respectively. The modification factor κ^t is employed to control the execution time of the two operators. According to the authors of NNA, κ^t is computed by

$$\kappa^{t+1} = \kappa^t \times 0.99 \tag{21}$$

The transfer operator can be written as

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{v}_i^t + 2\alpha_2 \left(\boldsymbol{g}^t - \boldsymbol{v}_i^t \right) \tag{22}$$

where $\boldsymbol{g}^t$ is the current optimal solution and α_2 is a random number between 0 and 1 following a uniform distribution. From Eq. (6), the transfer operator means that $\boldsymbol{v}_i^t$ moves towards the direction of $\boldsymbol{g}^t$. The bias operator is more complex than the transfer operator, which includes bias

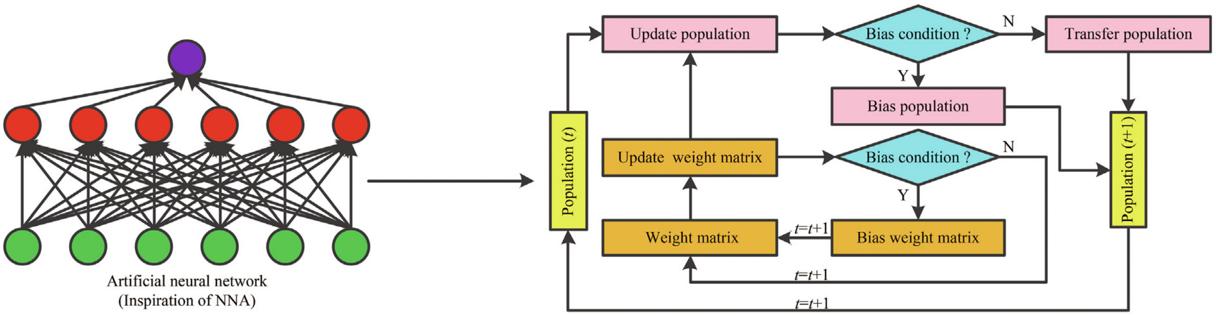


Fig. 2. The framework of NNA.

population and bias weight matrix. If a biased population is performed for one individual, it means $N_{\mathrm{P}}(1 \leq N_{\mathrm{P}} \leq D)$ elements in this individual are replaced with the randomly generated elements. Further, the bias population can be represented as

$$N_{\mathrm{P}} = \lceil D \cdot \kappa^t \rceil \tag{23}$$

$$v^t_{i,Q(s)} = l_{Q(s)} + \alpha_3 \left(u_{Q(s)} - l_{Q(s)}\right), s = 1, 2, \ldots, N_{\mathrm{P}} \tag{24}$$

where α_3 is a random number between 0 and 1 with uniform distribution, Q is a set consisting of N_{P} integers between 1 and D, $l_{Q(s)}$ is the lower limit of the $Q(s)^{\mathrm{th}}$ element, $u_{Q(s)}$ is the upper limit of the $Q(s)^{\mathrm{th}}$ element. If a bias weight matrix is performed for one individual, which indicates $N_{\mathrm{W}}(1 \leq N_{\mathrm{W}} \leq N)$ elements in the weight vector of this individual are replaced with the randomly generated elements. Further, the bias weight matrix can be defined as

$$N_{\mathrm{W}} = \lceil N \cdot \kappa^t \rceil \tag{25}$$

$$w^t_{i,R(r)} = c, r = 1, 2, \ldots, N_{\mathrm{W}} \tag{26}$$

where R is a set consisting of N_{W} integers between 1 and N, and c is a random number between 0 and 1 following a uniform distribution. In addition, the population $\boldsymbol{X}^t$ in NNA is initialized by

$$x^t_{i,j} = l_j + \alpha_4 \cdot \left(u_j - l_j\right) \tag{27}$$

where α_4 is a random number between 0 and 1 following a uniform distribution.

For one metaheuristic method, how to balance its global exploration and local exploitation is closely related to the performance of this method in solving global optimization problems. More specifically, if this method pays more attention to global exploration, this method will suffer from a slow convergence rate; if this method spends more time on local exploitation, this method may be trapped into a local optimum. In NNA, a modification factor is employed to balance its global exploration and local exploitation. According to Eq. (21), Fig. 3 shows the change of modification factor with the increasing number of iterations. From Fig. 3, the modification factor becomes small gradually. When the number of iterations is equal to 600, the value of the modification factor is very close to 0. The smaller the value of the modification factor is, the larger the chance of performing transfer operator (global exploration) is. Given the disadvantage of balancing global exploration and local exploitation, NNA may easily get trapped in a local optimum for complex optimization problems.

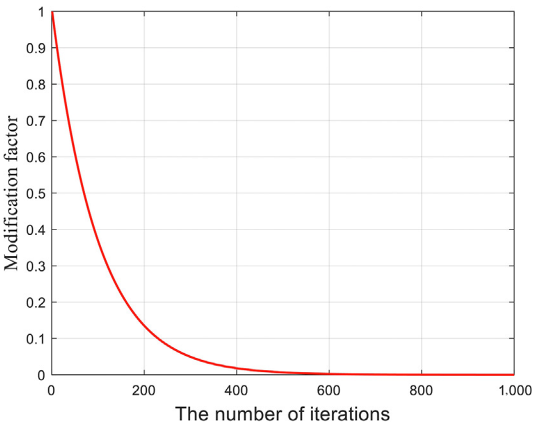


Fig. 3. The change of modification factor with the increasing of the number of iterations.

Multiple learning strategies are very helpful for keeping the population diversity, which is one of the often used methods to improve the convergence performance of metaheuristic methods [41–43]. Motivated by this, to enhance the global search ability of NNA, the proposed MLNNA designs three candidate learning strategies to perform bias population and transfer operator, respectively.

3.2. The framework of the proposed MLNNA

As mentioned previously, this work focuses on improving the global search ability of NNA in solving complex nonlinear optimization problems by the designed multiple learning strategies for biased population and bias transfer operators. Further, the designed multiple learning strategies are based on two introduced archives, i.e. local elite archive and global elite archive.

- Local elite archive. This archive is to save the historical best solution of each individual. Local elite archive $\boldsymbol{X}^t_{\mathrm{L}}$ can be denoted as $\boldsymbol{X}^t_{\mathrm{L}} = \{\boldsymbol{x}^t_{\mathrm{L},1}, \boldsymbol{x}^t_{\mathrm{L},2}, \ldots, \boldsymbol{x}^t_{\mathrm{L},N}\}$, where $\boldsymbol{x}^t_{\mathrm{L},i} = [x^t_{\mathrm{L},i,1}, x^t_{\mathrm{L},i,2}, \ldots, x^t_{\mathrm{L},i,D}]$ means the historical best solution of individual i, $i = 1, 2, \ldots, N$;
- Global elite archive. This archive is to save the historical outstanding solutions of the whole population. Global elite archive $\boldsymbol{X}^t_{\mathrm{G}}$ has the same length with $\boldsymbol{X}^t$, which can be denoted as $\boldsymbol{X}^t_{\mathrm{G}} = \{\boldsymbol{x}^t_{\mathrm{G},1}, \boldsymbol{x}^t_{\mathrm{G},2}, \ldots, \boldsymbol{x}^t_{\mathrm{G},N}\}$.

Next, the designed learning strategies for bias population and transfer operator based on the two archives are introduced in detail.

3.2.1. The designed learning strategies for bias population

In NNA, bias population is performed by Eq. (24). Eq. (24) indicates that some elements of trail vector $\boldsymbol{v}^t_i$ are replaced with some random numbers meeting the boundary constraints. This mechanism in Eq. (24) doesn't consider the obtained helpful information. To overcome this drawback, a new mechanism used to perform bias population is introduced, which can be expressed as

$$v^t_{i,Q(s)} = \begin{cases} x^t_{\mathrm{L},i,Q(s)}, \text{if } \ 0 \leq \rho_{\mathrm{BP}} < 1/3 \\ O^t_{Q(s)}, \text{if } \ 1/3 \leq \rho_{\mathrm{BP}} < 2/3 \\ l_{Q(s)} + \alpha_3\left(u_{Q(s)} - l_{Q(s)}\right), \text{if } \ 2/3 \leq \rho_{\mathrm{BP}} < 1 \end{cases} \tag{28}$$

where ρ_{BP} is a selection probability and $\boldsymbol{O}^t = [O^t_1, O^t_2, \ldots, O^t_D]$ is an opposite vector produced by opposition-based learning. In addition, ρ_{BP} is a random number between 0 and 1 with a uniform distribution. Further, $\boldsymbol{O}^t$ can be computed by

$$\boldsymbol{O}^t = \boldsymbol{V}^t_{\max} + \boldsymbol{V}^t_{\min} - \boldsymbol{v}^t_i \tag{29}$$

where $\boldsymbol{V}^t_{\max}$ and $\boldsymbol{V}^t_{\min}$ are the upper boundary and the lower boundary of the trail population $\boldsymbol{V}^t = \{\boldsymbol{v}^t_1, \boldsymbol{v}^t_2, \ldots, \boldsymbol{v}^t_N\}$, respectively. According to Eqs. (28) and (29), the built mechanism for bias population has the following features:

- This mechanism considers the historical optimal solution of each individual as shown in the first term on the right-hand side of Eq. (28) (i.e. $0 \leq \rho_{\mathrm{BP}} < 1/3$ is met). The historical optimal solution of one individual contains some beneficial information for this individual to find the better solutions. Given this, many reported metaheuristic methods, such as particle swarm optimization and its many improved versions, employ historical optimal solutions of individuals to guide the search directions of individuals. Thus, using historical optimal solutions in Eq. (28) also can help the corresponding individuals to move towards the direction that there are more chances for MLNNA to achieve better solutions;

- This mechanism takes into account the opposite solutions of individuals as shown in the second term on the right-hand side of Eq. (28) (i.e. $1/3 \le \rho_{\mathrm{BP}} < 2/3$ is met). Opposition-based learning has been widely used to improve the performance of metaheuristic methods, such as self-adaptive sine cosine algorithm with opposition-based learning [44], opposition-based learning Harris hawks optimization [45], crow search algorithm based on opposition-based learning [46], and selective opposition based grey wolf optimization [47]. The effectiveness of opposition-based learning is based on the theory that one solution and its opposite solution have the same chance to find better solutions [48]. That is, one solution may be inferior to its opposite solution in terms of fitness value. Motivated by the theory of opposition-based learning, as can be seen from Eq. (29), the opposite vector $\boldsymbol{O}^t$ is introduced to the designed mechanism, which can increase the chance of MLNNA obtaining better solutions;
- This mechanism also keeps the feature of the original NNA as presented in the third term on the right-hand side of Eq. (28) (i.e. $2/3 \le \rho_{\mathrm{BP}} < 1$ is met);
- The three learning strategies in Eq. (28) are employed to perform a bias population from three different angles, which have the same importance to improve the global search ability of NNA. Therefore, the three learning strategies share the same selected probability.

3.2.2. *The designed learning strategies for transfer operator*

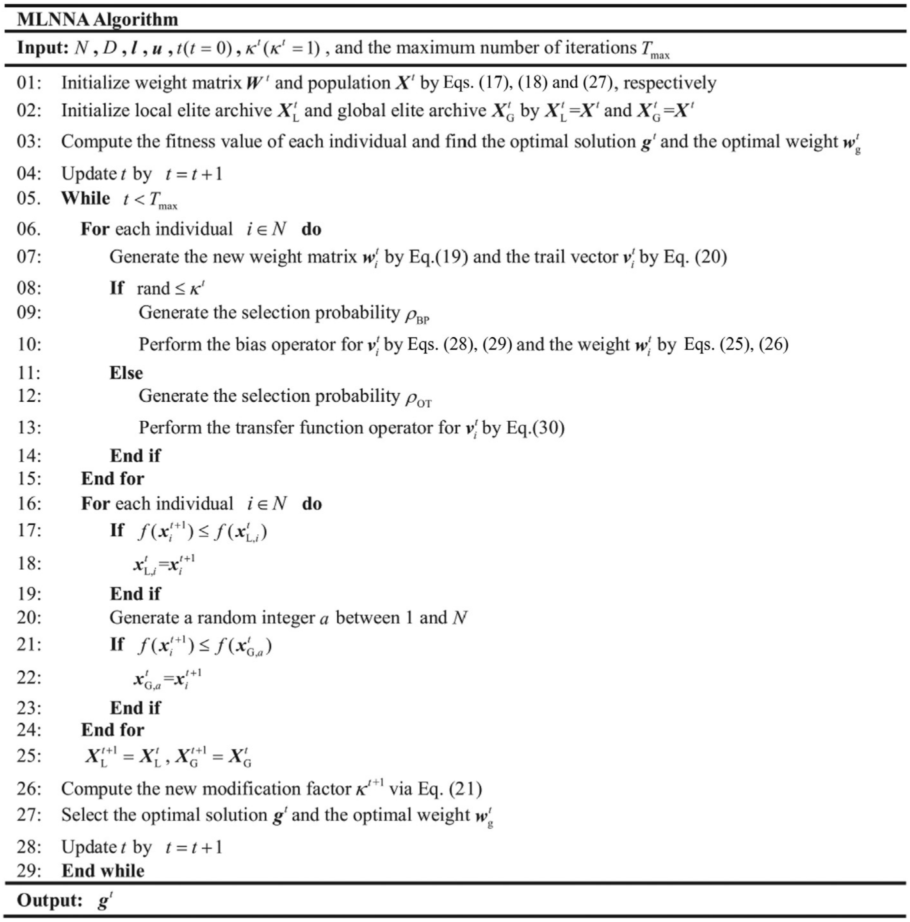
As discussed in Section 3.1, with the number of iterations increased, more time of NNA will be consumed by the transfer operator. In NNA, as can be seen from Eq. (22), the transfer operator is guided by the current optimal solution $\boldsymbol{g}^t$. Thus, if $\boldsymbol{g}^t$ is a locally optimal solution rather than a global optimal solution, NNA is more likely to be trapped in the local optimum. To increase the chance of NNA escaping from the local optimum, the transfer operator in MLNNA is driven by the current optimal solution $\boldsymbol{g}^t$, local elite archive $\boldsymbol{X}_{\mathrm{L}}^t$ and global elite archive $\boldsymbol{X}_{\mathrm{G}}^t$, which can be expressed as

$$\boldsymbol{x}_i^{t+1} = \begin{cases} \boldsymbol{v}_i^t + 2\alpha_4\left(\boldsymbol{g}^t - \boldsymbol{v}_i^t\right), \text{if } \ 0 \le \rho_{\mathrm{TO}} < 1/3 \\ \boldsymbol{v}_i^t + 2\alpha_5\left(\boldsymbol{x}_{\mathrm{L},i}^t - \boldsymbol{v}_i^t\right), \text{if } \ 1/3 \le \rho_{\mathrm{TO}} < 2/3 \\ \boldsymbol{v}_i^t + 2\alpha_6\left(\boldsymbol{x}_{\mathrm{G},i}^t - \boldsymbol{v}_i^t\right), \text{if } \ 2/3 \le \rho_{\mathrm{TO}} < 1 \end{cases} \tag{30}$$

where α_4, α_5 and α_6 are three random numbers between 0 and 1 with uniform distribution, and ρ_{TO} is the selection probability. In addition, ρ_{TO} is a random number between 0 and 1 following uniform distribution. According to Eq. (30), the first term of the right-hand side (i.e. $0 \le \rho_{\mathrm{TO}} < 1/3$ is met) is the same as Eq. (22) in NNA, which is guided by the current optimal solution $\boldsymbol{g}^t$; the second term of the right-hand side (i.e. $1/3 \le \rho_{\mathrm{TO}} < 2/3$ is met) is guided by the local elite archive $\boldsymbol{X}_{\mathrm{L}}^t$, which is local learning model; the third term of right-hand side (i.e. $2/3 \le \rho_{\mathrm{TO}} < 1$ is met) is guided by the global elite archive $\boldsymbol{X}_{\mathrm{G}}^t$, which is a global learning model. The three learning strategies have the same importance for MLNNA to find better solutions and are assigned the same selected probability.

3.3. *Implementation of MLNNA*

Fig. 4 shows the pseudocode of MLNNA. By observing Fig. 4, MLNNA has a simple structure and its core idea is the designed multiple learning strategies for the bias operator and the transfer operator. The features of MLNNA can be summarized as follows:



```
MLNNA Algorithm
Input: $N$, $D$, $\boldsymbol{l}$, $\boldsymbol{u}$, $t(t=0)$, $\kappa^t(\kappa^t=1)$, and the maximum number of iterations $T_{\max}$
Initialize weight matrix $\boldsymbol{W}^t$ and population $\boldsymbol{X}^t$ by Eqs. (17), (18) and (27), respectively
Initialize local elite archive $\boldsymbol{X}_{\mathrm{L}}^t$ and global elite archive $\boldsymbol{X}_{\mathrm{G}}^t$ by $\boldsymbol{X}_{\mathrm{L}}^t=\boldsymbol{X}^t$ and $\boldsymbol{X}_{\mathrm{G}}^t=\boldsymbol{X}^t$
Compute the fitness value of each individual and find the optimal solution $\boldsymbol{g}^t$ and the optimal weight $\boldsymbol{w}_{\mathrm{g}}^t$
Update $t$ by $t=t+1$
While $t<T_{\max}$
    For each individual $i \in N$ do
        Generate the new weight matrix $\boldsymbol{w}_i^t$ by Eq.(19) and the trail vector $\boldsymbol{v}_i^t$ by Eq. (20)
        If rand $\le \kappa^t$
            Generate the selection probability $\rho_{\mathrm{BP}}$
            Perform the bias operator for $\boldsymbol{v}_i^t$ by Eqs. (28), (29) and the weight $\boldsymbol{w}_i^t$ by Eqs. (25), (26)
        Else
            Generate the selection probability $\rho_{\mathrm{OT}}$
            Perform the transfer function operator for $\boldsymbol{v}_i^t$ by Eq.(30)
        End if
    End for
    For each individual $i \in N$ do
        If $f(\boldsymbol{x}_i^{t+1}) \le f(\boldsymbol{x}_{\mathrm{L},i}^t)$
            $\boldsymbol{x}_{\mathrm{L},i}^t=\boldsymbol{x}_i^{t+1}$
        End if
        Generate a random integer $a$ between 1 and $N$
        If $f(\boldsymbol{x}_i^{t+1}) \le f(\boldsymbol{x}_{\mathrm{G},a}^t)$
            $\boldsymbol{x}_{\mathrm{G},a}^t=\boldsymbol{x}_i^{t+1}$
        End if
    End for
    $\boldsymbol{X}_{\mathrm{L}}^{t+1}=\boldsymbol{X}_{\mathrm{L}}^t$, $\boldsymbol{X}_{\mathrm{G}}^{t+1}=\boldsymbol{X}_{\mathrm{G}}^t$
    Compute the new modification factor $\kappa^{t+1}$ via Eq. (21)
    Select the optimal solution $\boldsymbol{g}^t$ and the optimal weight $\boldsymbol{w}_{\mathrm{g}}^t$
    Update $t$ by $t=t+1$
End while
Output: $\boldsymbol{g}^t$
```

Fig. 4. The pseudocode of the proposed MLNNA.

Table 1
Experimental results of NNA and MLNNA on CEC 2015 test suite.

No.	NNA				MLNNA			
	MAX	MEAN	MIN	STD	MAX	MEAN	MIN	STD
F_1	2.583E+07	1.049E+07	3.993E+06	5.167E+06	3.614E+06	1.711E+06	6.374E+05	8.257E+05
F_2	4.640E+04	8.668E+03	2.105E+02	9.466E+03	4.639E+04	1.647E+04	2.042E+02	1.674E+04
F_3	3.204E+02	3.201E+02	3.200E+02	8.657E-02	3.203E+02	3.201E+02	3.200E+02	9.398E-02
F_4	9.039E+02	7.434E+02	6.411E+02	7.319E+01	8.358E+02	6.909E+02	5.752E+02	5.891E+01
F_5	9.352E+03	7.216E+03	5.191E+03	8.846E+02	8.323E+03	6.757E+03	5.458E+03	7.268E+02
F_6	2.566E+06	1.118E+06	2.709E+05	5.790E+05	9.746E+05	4.304E+05	3.968E+04	2.339E+05
F_7	8.230E+02	7.712E+02	7.191E+02	2.226E+01	8.333E+02	7.731E+02	7.238E+02	2.432E+01
F_8	2.192E+06	1.053E+06	2.882E+05	5.477E+05	6.954E+05	2.781E+05	2.664E+04	1.670E+05
F_9	1.012E+03	1.009E+03	1.007E+03	1.139E+00	1.009E+03	1.007E+03	1.006E+03	6.406E-01
F_{10}	7.670E+05	2.028E+05	1.800E+04	1.826E+05	5.332E+04	1.836E+04	7.985E+03	1.281E+04
F_{11}	3.017E+03	2.691E+03	2.452E+03	1.367E+02	2.871E+03	2.614E+03	2.296E+03	1.281E+02
F_{12}	1.400E+03	1.374E+03	1.312E+03	4.068E+01	1.400E+03	1.391E+03	1.310E+03	2.729E+01
F_{13}	1.300E+03	1.300E+03	1.300E+03	2.221E-02	1.300E+03	1.300E+03	1.300E+03	8.275E-03
F_{14}	8.266E+04	7.178E+04	5.093E+04	8.054E+03	8.083E+04	6.910E+04	6.042E+04	8.400E+03
F_{15}	1.617E+03	1.613E+03	1.606E+03	2.178E+00	1.616E+03	1.609E+03	1.602E+03	4.249E+00

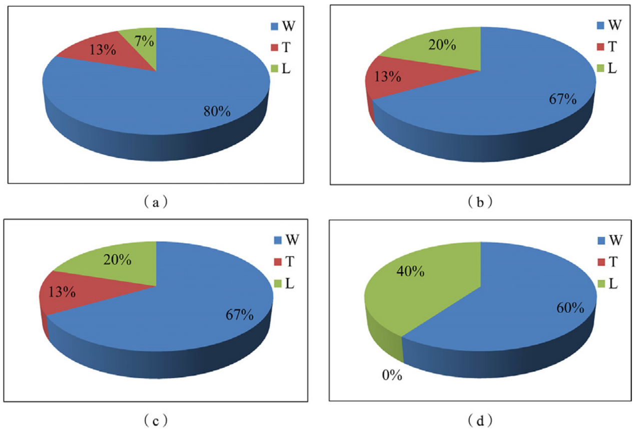


Fig. 5. The statistical results of NNA and MLNNA on CEC 2015 test suite. “W” means the percentage that MLNNA outperforms NNA; “T” indicates the percentage that MLNNA has the same performance with NNA; “L” denotes the percentage that MLNNA is inferior to NNA. (a) MAX. (b) MEAN. (c) MIN. (d) STD.

- When solving optimization problems, MLNNA only needs the basic population size and terminal condition. Thus, it is very easy for MLNNA to be applied for solving different types of global optimization problems.
- As can be seen from Eqs. (28) and (30), multiple learning strategies are designed for performing the optimization tasks in MLNNA, which are very helpful for keeping the population diversity and can increase the chance of MLNNA to escape from the local optimum to some extent.
- Like neighbourhood topology [49,50] used in metaheuristic methods, the built global elite archive in MLNNA also can enhance the global search ability of NNA by increasing the chance of communication among individuals.
- MLNNA doesn't introduce complex operation rules. Thus, MLNNA is very easy to be coded and implemented.

Table 3
Search boundaries of the applied PEMFC models.

Search boundary	The unknown parameters						
	ξ_1	ξ_2	ξ_3	ξ_4	λ	R_c	β
Lower boundary	−1.199,69	0.001	3.60e-5	−2.60e-4	10	1e-4	0.013,6
Upper boundary	−0.853,20	0.005	9.80e-5	−9.54e-5	24	8e-4	0.500,0

4. Performance evaluation of MLNNA

This section verifies the validity of the improved strategies by comparing the performance between NNA and MLNNA on the well-known CEC 2015 test suite [51]. CEC 2015 test suite consists of 15 test functions, i.e. two unimodal functions (i.e. F_1 and F_2), three simple multimodal functions (i.e. F_3, F_4, and F_5), three hybrid functions (i.e. F_6, F_7, and F_8), and seven composition functions (i.e. F_9, F_{10}, F_{11}, F_{12}, F_{13}, F_{14}, and F_{15}). Note that, these test functions are minimization problems and most of these test functions are complex multimodal functions. Thus, the selected CEC 2015 test suite is suitable for investigating the global search ability of MLNNA for complex optimization problems. As done in [38], the population size and the maximum number of iterations are set to 50 and 5,000 for NNA and MLNNA, respectively. Besides, each algorithm is executed 30 independent runs for each test function with 50-dimensional and experimental results are recorded based on the following indicators: the mean solution (MEAN), the worst solution (MAX), the best solution (MIN) and standard variance (STD).

Table 1 shows the experimental results obtained by NNA and MLNNA on CEC 2015 test suite. Fig. 5 presents the statistical results of NNA and MLNNA based on Table 1. According to Table 1 and Fig. 5, MLNNA outperforms NNA on 80 percent of test functions (i.e. F_1, F_2, F_3, F_4, F_5, F_6, F_8, F_9, F_{10}, F_{11}, F_{14}, and F_{15}) in terms of MAX; MLNNA can beat NNA on 67 percent of test functions (i.e. F_1, F_4, F_5, F_6, F_8, F_9, F_{10}, F_{11}, F_{14}, and F_{15}) in terms of MEAN, MLNNA is superior to NNA on 67 percent of test functions (i.e. F_1, F_2, F_4, F_6, F_8, F_9, F_{10}, F_{11}, F_{12}, and F_{15}) in terms of MIN; MLNNA shows advantages over NNA on 60 percent of test functions (i.e. F_1, F_4, F_5, F_6, F_9, F_{10}, F_{11}, F_{12}, and F_{13}) in terms of STD. In addition,

Table 2
Datasheets of the applied PEMFC models.

Model	n	A (cm^2)	l (μm)	P_{H_2} (bar)	P_{O_2} (bar)	T (K)	R_{ha}	R_{hc}	J_{max} ($A \cdot cm^{-2}$)
NedStack PS6	65	240	178	1.0	1.0	343	100%	100%	1.400
BCW 500 W	32	64	178	1.0	0.209,5	333	100%	100%	0.469

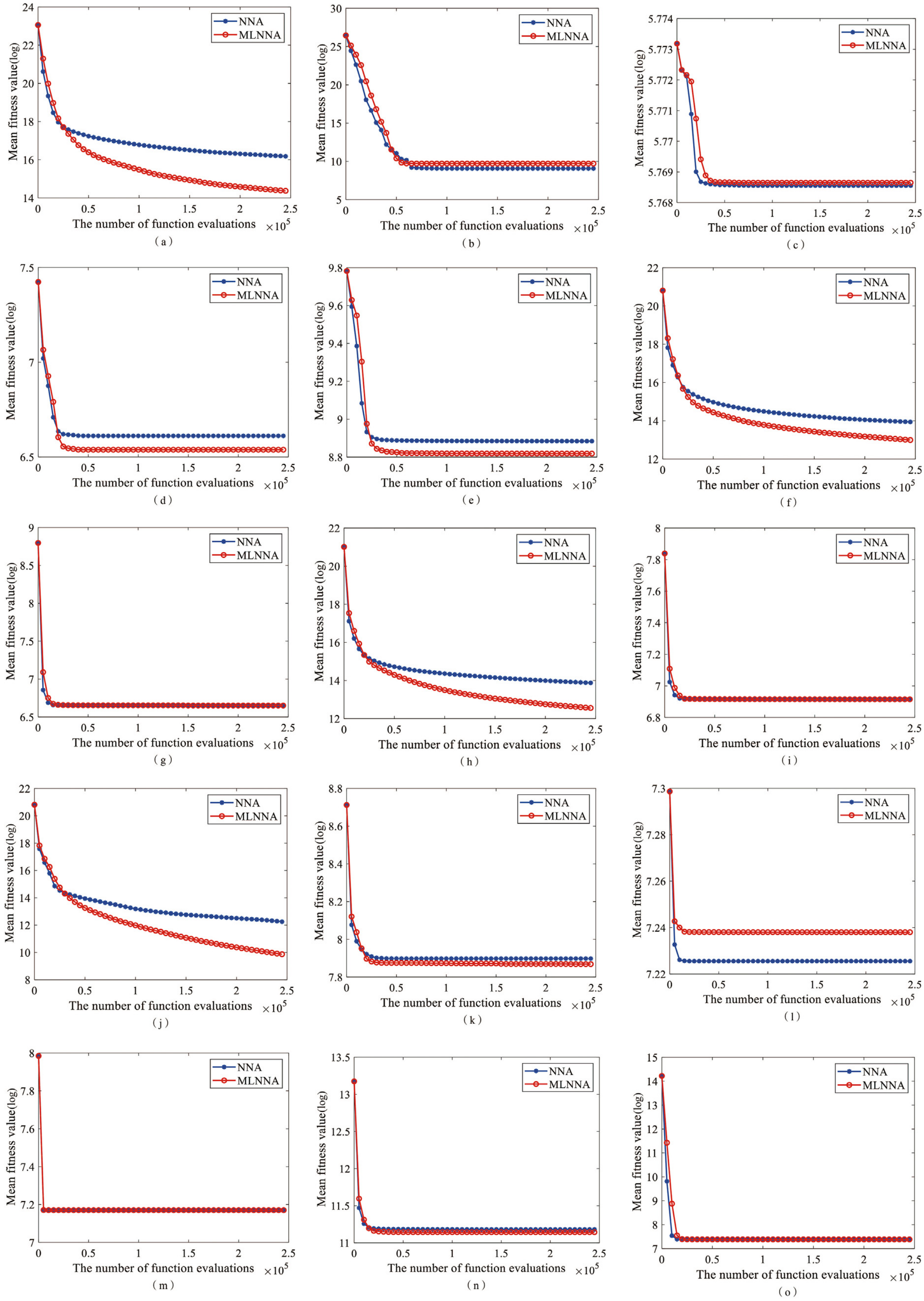


Fig. 6. Convergence curves obtained by NNA and MLNNA.(a) F1.(b) F2.(c) F3.(d) F4.(e) F5.(f) F6.(g) F7.(h) F8.(i) F9.(j) F10.(k) F11.(l) F12.(m) F13.(n) F14.(o) F15.

MLNNA is only inferior to NNA on one test function (i.e. F_1), three test functions (i.e. F_2, F_7, and F_{12}), three test functions (i.e. F_5, F_7, and F_{14}), and six test functions (i.e. F_2, F_3, F_7, F_8, F_{14}, and F_{15}) in terms of MAX, MEAN, MIN, and STD, respectively. MLNNA outperforms NNA on CEC 2015 test suite in terms of solution quality.

To compare the convergence performance of NNA and MLNNA, Fig. 6 shows the convergence curves obtained by MLNNA and NNA on CEC 2015 test suite. From Fig. 6, MLNNA has obvious convergence advantages over NNA on six test functions, i.e. F_1, F_4, F_5, F_6, F_8, and F_{10}. For F_{11} and F_{14}, although NNA shows competitiveness, MLNNA still can find better solutions with a faster speed. In addition, MLNNA and NNA have similar convergence performances on F_7, F_{13}, and F_{15}. Besides, NNA only outperforms MLNNA on the rest two test functions, i.e. F_2, and F_{12}, in terms of convergence performance. Clearly, MLNNA is superior to NNA in terms of convergence performance on CEC 2015 test suite.

5. Application in the parameter estimation of PEMFC models

Section 4 has demonstrated the excellent global search ability of MLNNA in solving complex numerical optimization problems. This lays a very good foundation for the application of MLNNA in the parameter estimation of PEMFC models. The considered PEMFC models in this section are the classical NedStack PS6 PEMFC model and BCS 500 W PEMFC model, whose datasheets and search boundaries have been shown in Tables 2 and 3, respectively. The data information in Tables 2 and 3 can be found in [52].

To verify the competitiveness of MLNNA, 10 powerful metaheuristic methods are compared with MLNNA, which are NNA, whale optimization algorithm (WOA) [53], seagull optimization algorithm (SOA) [54], transient search algorithm (TSO) [55], backtracking search algorithm (BSA) [56], tunicate swarm algorithm (TSA) [57], Jaya algorithm (JAYA) [58], sooty tern optimization algorithm (STOA) [59], sine cosine algorithm (SCA) [60], and chaotic neural network algorithm with competitive learning (CLNNA) [61]. For a fair comparison, population size and the maximum number of function evaluations for all the applied

Table 6
The estimated data based on the extracted optimal parameters by MLNNA on BCW 500 W PEMFC model.

No.	Measured data		Estimated data	Estimated error
	$I_{\text{mea}}(\text{A})$	$V_{\text{mea}}(\text{V})$	$V_{\text{cal}}(\text{V})$	$\lvert V_{\text{cal}} - V_{\text{mea}}\rvert^2$
1	0.6	29	28.997,219,3	0.000,007,7
2	2.1	26.31	26.305,937,5	0.000,016,5
3	3.58	25.09	25.093,557,0	0.000,012,7
4	5.08	24.25	24.254,622,5	0.000,021,4
5	7.17	23.37	23.375,418,3	0.000,029,4
6	9.55	22.57	22.584,616,8	0.000,213,7
7	11.35	22.06	22.071,328,7	0.000,128,3
8	12.54	21.75	21.758,464,4	0.000,071,6
9	13.73	21.45	21.461,263,1	0.000,126,9
10	15.73	21.09	20.987,741,3	0.010,456,8
11	17.02	20.68	20.694,508,7	0.000,210,5
12	19.11	20.22	20.230,984,5	0.000,120,7
13	21.2	19.76	19.770,941,1	0.000,119,7
14	23	19.36	19.366,022,1	0.000,036,3
15	25.08	18.86	18.866,463,5	0.000,041,8
16	27.17	18.27	18.274,718,5	0.000,022,3
17	28.06	17.95	17.953,309,7	0.000,011,0
18	29.26	17.3	17.292,879,3	0.000,050,7
TSE				0.011,697,8

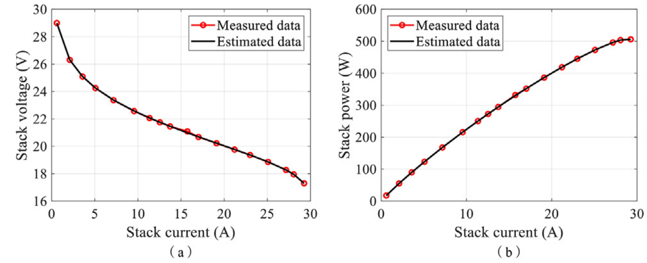


Fig. 7. Characteristic curves based on the estimated optimal parameters by MLNNA on BCW 500 W PEMFC model. (a) *V-I* curve. (b) *P-I* curve.

Table 4
Statistical results obtained by MLNNA and 10 compared algorithms on BCW 500 W PEMFC model.

Algorithm	Indicator				
	MAX	MEAN	MDN	MIN	STD
WOA	2.798,572,0	0.500,550,8	0.250,207,1	0.020,157,0	0.652,639,3
SOA	0.058,284,2	0.026,365,5	0.026,165,1	0.014,142,2	0.009,763,4
TSO	5.368,867,0	3.510,389,4	4.437,429,2	0.038,903,6	1.546,316,4
BSA	0.014,414,6	0.012,071,0	0.011,947,8	0.011,713,0	0.000,503,0
NNA	0.013,286,1	0.012,059,2	0.011,955,1	0.011,719,1	0.000,372,0
TSA	0.028,108,4	0.018,040,5	0.016,904,1	0.012,613,0	0.004,267,5
SCA	0.612,602,4	0.147,508,5	0.109,207,7	0.023,830,7	0.117,212,0
JAYA	0.272,693,8	0.079,100,9	0.060,150,8	0.030,005,8	0.054,918,7
STOA	0.251,756,2	0.036,466,4	0.024,290,9	0.012,835,7	0.044,512,3
CLNNA	0.021,017,6	0.013,205,8	0.012,672,4	0.011,785,9	0.001,830,2
MLNNA	**0.013,150,1**	**0.011,803,3**	**0.011,706,5**	**0.011,697,8**	**0.000,266,9**

Table 5
The estimated optimal parameters by MLNNA and 10 compared algorithms on BCW 500 W PEMFC model.

Algorithm	The estimated optimal parameters							TSE
	ξ_1	ξ_2	ξ_3	ξ_4	λ	R_c	β	
WOA	−0.870,005,3	0.002,268,4	0.000,038,4	−0.000,190,1	19.903,153,8	0.313,377,9	0.014,226,1	0.020,157,0
SOA	−1.166,717,9	0.004,021,3	0.000,093,9	−0.000,192,7	20.587,955,7	0.139,593,5	0.015,697,9	0.014,142,2
TSO	−0.855,268,5	0.002,697,6	0.000,069,4	−0.000,183,8	19.650,330,9	0.492,583,6	0.013,605,6	0.038,903,6
BSA	−0.961,315,8	0.003,015,6	0.000,068,9	−0.000,193,0	20.892,120,5	0.100,000,0	0.016,173,5	0.011,713,0
NNA	−0.853,729,3	0.002,381,2	0.000,048,9	−0.000,192,8	20.700,761,0	0.100,001,9	0.016,028,4	0.011,719,1
TSA	−1.199,700,0	0.003,303,1	0.000,041,4	−0.000,191,3	24.000,000,0	0.442,336,2	0.015,940,2	0.012,613,0
SCA	−1.157,253,9	0.003,649,0	0.000,071,5	−0.000,197,2	24.000,000,0	0.186,353,4	0.017,192,3	0.023,830,7
JAYA	−0.853,200,0	0.002,178,9	0.000,036,0	−0.000,190,1	22.155,864,3	0.526,215,7	0.013,600,0	0.030,005,8
STOA	−1.012,049,2	0.002,734,9	0.000,041,0	−0.000,193,3	22.845,001,9	0.268,871,6	0.016,113,7	0.012,835,7
CLNNA	−0.935,050,8	0.003,074,9	0.000,077,8	−0.000,193,3	21.597,296,0	0.139,526,4	0.016,306,3	0.011,785,9
MLNNA	−0.874,717,4	0.002,301,6	0.000,039,7	−0.000,193,0	20.876,926,2	0.100,000,0	0.016,126,0	**0.011,697,8**

Table 7
The ranking results from Friedman test for the applied algorithms on BCW 500 W PEMFC model.

Algorithm	WOA	SOA	TSO	BSA	NNA	TSA	SCA	JAYA	STOA	CLNNA	MLNNA
Ranking	8.77 (9)	6.53 (6)	10.97 (11)	2.40 (2)	2.53 (3)	5.33 (5)	9.07 (10)	8.47 (8)	6.80 (7)	3.73 (4)	1.40 (1)

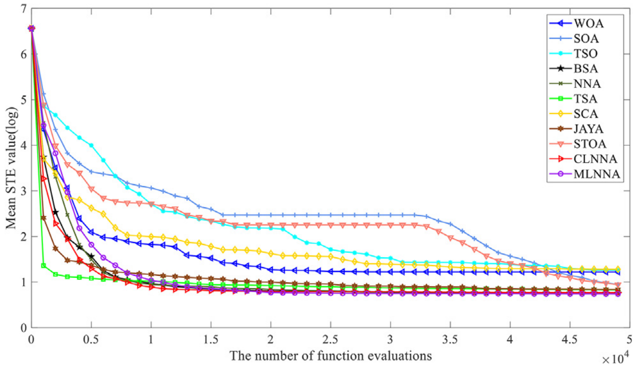


Fig. 8. Convergence of MLNNA and the compared algorithms on the BCW 500 W PEMFC model.

algorithms are set to 50 and 50,000, respectively. The other control parameters of the compared algorithms are extracted directly from the corresponding references. Each algorithm is executed 30 independent runs for each case and the statistical results are recorded. To compare the performance among the applied algorithms, the obtained experimental results are discussed by the following methods:

(1) Value-based method. This method is to evaluate the obtained solution quality in terms of five performance metrics, i.e. the mean solution (MEAN), the worst solution (MAX), the median solution (MDN), the best solution (MIN) and standard variance (STD). The best results are highlighted in bold as shown in Tables 4, 5, 8 and 9;
(2) Rank-based method. This method is to sort the applied algorithms according to the obtained experimental results. In this section, the well-known Friedman test [62–65] is employed to get the average rankings of the applied algorithms;
(3) Convergence-based method. This method is to compare the convergence performance among the applied algorithms by observing the obtained convergence curves by the applied algorithms on each PEMFC model.

5.1. Experimental results on BCW 500 W PEMFC model

Table 4 shows the results of MLNNA and the compared algorithms on the BCW 500 W PEMFC model. From Table 4, MLNNA can find the best MAX, MEAN, MED, MIN, and STD, which means that MLNNA is superior to the compared algorithms in of the solution quality and the stability. By observing Table 4, the obtained best solution is 0.011,697,8. The best solutions of BSA, NNA, and CLNNA are very close to that of MLNNA, which are 0.011,713,0, 0.011,719,1, and 0.011,785,9, respectively. In addition, the best solutions of TSO and JAYA are more than 0.03, which is obviously more than those of the other algorithms. That is, TSO and JAYA aren not suitable for extracting the unknown parameters of the BCW 500 W PEMFC model. Table 5 shows the estimated optimal parameters by MLNNA and the compared algorithms on BCW 500 W PEMFC model. The estimated data based on the extracted optimal parameters by MLNNA on BCW 500 W PEMFC model has been listed in Table 6. According to Table 6, Fig. 7 shows the characteristic curves based on the estimated optimal parameters by MLNNA on BCW 500 W PEMFC model. As seen in Fig. 7, the estimated data obtained by MLNNA is very similar to the measured data.

In addition, Table 7 lists the ranking results from the Friedman test for the applied algorithms on BCW 500 W PEMFC model. By observing Table 7, all algorithms can be sorted from the best to the worst in the following order: MLNNA, BSA, NNA, CLNNA, TSA, SOA, STOA, JAYA, WOA, SCA, and TSO. MLNNA is the best of all algorithms in terms of the performance of estimating the unknown parameters of the BCW 500 W PEMFC model. Besides, Fig. 8 displays the average convergence curves of MLNNA and the compared algorithms on the BCW 500 W PEMFC model. As can be seen from Fig. 8, MLNNA has obvious convergence advantages over TSA, SOA, STOA, JAYA, WOA, SCA, and TSO. Although NNA, CLNNA, and BSA show strong competitiveness, they are slightly inferior to MLNNA in terms of convergence performance. Note that, NNA, CLNNA, and BSA can obtain a faster convergence speed than MLNNA before 15,000 function evaluations. After this, NNA, CLNNA, and BSA gradually converge to some fixed values while MLNNA still can continue to find better solutions. This demonstrates the excellent ability of MLNNA to escape from the local optima.

5.2. Experimental results on the NedStack PS6 PEMFC model

The experimental results of MLNNA and the compared algorithms on the NedStack PS6 PEMFC model have been displayed in Table 8. As seen in Table 8, MLNNA can get the best MEAN and MDN. MLNNA and NNA can share the best MIN, which is 2.079,172,7. In terms of MAX and STD, MLNNA is only slightly inferior to BSA and can beat the other nine algorithms. In addition, the best solutions achieved by BSA, JAYA, and CLNNA are 2.079,172,7, 2.079,168,9, and 2.080,330,5, which are very close to that of MLNNA. For TSO, TSA, SCA, and STOA, their best solutions are significantly more than those of the rest seven algorithms, which shows that the four algorithms aren't suitable for parameter

Table 8
Statistical results of MLNNA and 10 compared algorithms on the NedStack PS6 PEMFC model.

Algorithm	Indicator				
	MAX	MEAN	MDN	MIN	STD
WOA	4.815,119,6	3.391,756,2	2.104,337,0	2.104,337,0	1.028,058,1
SOA	3.487,852,1	2.424,022,0	2.257,518,9	2.105,816,9	0.358,277,5
TSO	7.314,491,3	3.499,808,6	2.991,472,6	2.127,722,9	1.376,477,8
BSA	**2.198,173,5**	2.132,802,4	2.128,365,1	2.079,172,7	**0.039,686,7**
NNA	2.409,173,4	2.130,814,2	2.087,611,3	**2.079,165,7**	0.077,927,9
TSA	2.670,029,8	2.287,907,3	2.243,680,0	2.148,199,3	0.127,647,2
SCA	6.416,707,7	3.598,925,7	3.249,874,7	2.299,791,6	1.109,547,7
JAYA	4.960,889,1	2.309,973,0	2.163,201,1	2.079,168,9	0.528,338,8
STOA	3.375,255,4	2.414,671,5	2.225,163,2	2.130,789,4	0.356,851,5
CLNNA	2.285,108,7	2.153,205,3	2.140,810,8	2.080,330,5	0.052,937,7
MLNNA	2.199,531,0	**2.096,328,3**	**2.079,165,7**	**2.079,165,7**	0.039,777,4

Table 9
The estimated optimal parameters by MLNNA and 10 compared algorithms on NedStack PS6 PEMFC model.

Algorithm	The estimated optimal parameters							TSE
	ξ_1	ξ_2	ξ_3	ξ_4	λ	R_c	β	
WOA	−0.853,200,0	0.003,267,3	0.000,098,0	−0.000,095,4	13.226,355,2	0.100,252,9	0.017,246,5	2.104,337,0
SOA	−0.868,326,3	0.002,444,6	0.000,036,0	−0.000,095,4	13.183,541,4	0.100,000,0	0.016,283,3	2.105,816,9
TSO	−0.853,200,0	0.002,535,3	0.000,045,7	−0.000,095,4	14.258,862,5	0.100,928,4	0.034,383,2	2.127,722,9
BSA	−0.997,674,0	0.002,884,5	0.000,040,6	−0.000,095,4	13.094,760,2	0.100,000,0	0.013,600,0	2.079,172,7
NNA	−1.199,699,2	0.004,246,1	0.000,095,9	−0.000,095,4	13.094,707,9	0.100,000,0	0.013,600,0	**2.079,165,7**
TSA	−1.199,700,0	0.003,813,8	0.000,064,8	−0.000,095,4	14.376,959,8	0.100,104,6	0.037,295,6	2.148,199,3
SCA	−1.199,700,0	0.003,480,5	0.000,041,4	−0.000,095,4	15.557,642,0	0.100,000,0	0.051,493,4	2.299,791,6
JAYA	−1.199,700,0	0.004,276,0	0.000,098,0	−0.000,095,4	13.095,492,7	0.100,000,0	0.013,600,0	2.079,168,9
STOA	−0.877,283,3	0.002,612,3	0.000,046,1	−0.000,095,4	13.936,326,9	0.102,480,8	0.028,849,5	2.130,789,4
CLNNA	−0.922,863,7	0.002,606,0	0.000,036,3	−0.000,095,4	13.105,970,5	0.100,039,3	0.013,933,1	2.080,330,5
MLNNA	−1.097,728,8	0.003,143,9	0.000,038,3	−0.000,095,4	13.094,707,9	0.100,000,0	0.013,600,0	**2.079,165,7**

Table 10
The estimated data based on the extracted optimal parameters by MLNNA on the NedStack PS6 PEMFC model.

No.	Measured data		Estimated data	Estimated error
	I_{mea}(A)	V_{mea}(V)	V_{cal}(V)	$\lvert V_{\text{cal}} - V_{\text{mea}}\rvert^2$
1	2.25	61.64	62.353,858,8	0.509,594,4
2	6.75	59.57	59.779,753,5	0.043,996,5
3	9	58.94	59.048,342,0	0.011,738,0
4	15.75	57.54	57.496,144,3	0.001,923,3
5	20.25	56.8	56.717,480,5	0.006,809,5
6	24.75	56.13	56.044,192,1	0.007,363,0
7	31.5	55.23	55.157,027,4	0.005,325,0
8	36	54.66	54.620,427,6	0.001,566,0
9	45	53.61	53.632,898,6	0.000,524,3
10	51.75	52.86	52.943,890,3	0.007,037,6
11	67.5	51.91	51.439,581,9	0.221,293,2
12	72	51.22	51.027,142,0	0.037,194,2
13	90	49.66	49.418,847,6	0.058,154,5
14	99	49	48.628,063,9	0.138,336,4
15	105.8	48.15	48.032,343,2	0.013,843,1
16	110.3	47.52	47.638,017,6	0.013,928,2
17	117	47.1	47.049,696,4	0.002,530,5
18	126	46.48	46.255,102,9	0.050,578,7
19	135	45.66	45.452,990,4	0.042,853,0
20	141.8	44.85	44.840,386,7	0.000,092,4
21	150.8	44.24	44.018,951,8	0.048,862,3
22	162	42.45	42.976,582,2	0.277,288,8
23	171	41.66	42.120,058,8	0.211,654,1
24	182.3	40.68	41.017,120,6	0.113,650,3
25	189	40.09	40.347,023,2	0.066,060,9
26	195.8	39.51	39.653,440,7	0.020,575,2
27	204.8	38.73	38.712,836,6	0.000,294,6
28	211.5	38.15	37.994,529,9	0.024,170,9
29	220.5	37.38	37.003,268,4	0.141,926,7
TSE				2.079,165,7

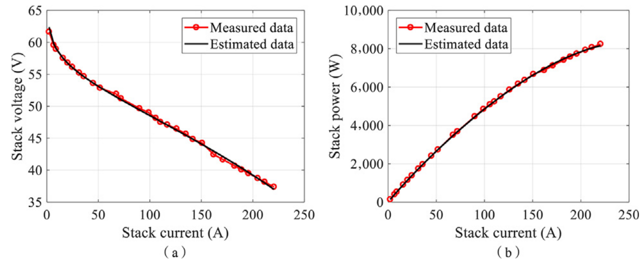


Fig. 9. Characteristic curves based on the estimated optimal parameters by MLNNA on NedStack PS6 PEMFC model. (a) *V-I* curve. (b) *P-I* curve.

estimation of the NedStack PS6 PEMFC model. Table 9 presents the extracted optimal parameters by MLNNA and the compared algorithms on the NedStack PS6 PEMFC model. The estimated data based on the extracted optimal parameters by MLNNA on the NedStack PS6 PEMFC model has been shown in Table 10. According to Table 10, Fig. 9 depicts the characteristic curves based on the estimated optimal parameters by MLNNA on the NedStack PS6 PEMFC model. As can be seen from Fig. 9, the measured data includes 29 voltage points. The measured data and the estimated data have large differences at voltage point 11, voltage point 14, voltage point 22, and voltage point 23. However, the estimated data is very similar to the measured data at the rest 25 voltage points.

In addition, the ranking results from the Friedman test for the applied algorithms on the NedStack PS6 PEMFC model have been presented in Table 11. According to Table 11, all algorithms can be sorted from the best to the worst in the following order: MLNNA, NNA, BSA, CLNNA, JAYA, TSA, STOA (SOA), WOA, TSO, and SCA. MLNNA is the best of all algorithms in terms of the performance of estimating the unknown parameters of the NedStack PS6 PEMFC model. Besides, Fig. 10 shows the average convergence curves of MLNNA and the compared algorithms on the NedStack PS6 PEMFC model. From Fig. 10, MLNNA can get a better solution with a faster speed than TSA, SOA, STOA, JAYA, WOA, SCA, and TSO. Note that, NNA, CLNNA, and BSA show excellent global search ability. However, MLNNA is slightly superior to NNA, CLNNA, and BSA in terms of convergence performance. By observing Fig. 10, like Fig. 8, MLNNA still shows a stronger ability to escape from the local optimal solutions compared with NNA, BSA, and CLNNA.

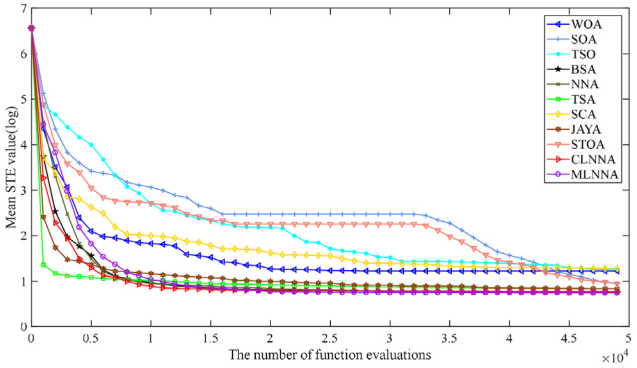


Fig. 10. Convergence curves of MLNNA and the compared algorithms on the NedStack PS6 PEMFC model.

Table 11
The ranking results from Friedman test for the applied algorithms on the NedStack PS6 PEMFC model.

Algorithm	WOA	SOA	TSO	BSA	NNA	TSA	SCA	JAYA	STOA	CLNNA	MLNNA
Ranking	9.10 (9)	6.83 (7)	9.47 (10)	3.57 (3)	3.07 (2)	6.53 (6)	9.60 (11)	5.13 (5)	6.83 (7)	4.20 (4)	1.67 (1)

6. Discussion on the effectiveness of the improved strategies

This work is aimed at extracting the unknown parameters of PEMFC models by the designed MLNNA. MLNNA is an improved version of NNA. This section is to discuss the validity of the improved strategies for introducing MLNNA based on the experimental results in Section 4 and Section 5.

The core idea of MLNNA is to enhance the global search ability of NNA in solving complex nonlinear optimization problems by designing multiple learning strategies, which are based on the introduced two elite archives, i.e. local elite archive and global elite archive. The two elite archives are used to design the learning strategies of the transfer operator and bias operator. Further, in MLNNA, there are three candidate bias operators and three candidate transfer operators. The three bias operators are driven by the individuals in the local elite archive, the individuals in the opposite population, and the random individuals in the search space. The three transfer operators are driven by the current optimal individual, the individuals in the local elite archive, and the individuals in the global elite archive. MLNNA can make full use of the obtained population information, which is very helpful for improving the population diversity and increasing the chance of escaping from the local optimal solutions.

The global search ability of MLNNA on complex numerical optimization problems is checked by comparing the performance between NNA and MLNNA on the well-known CEC 2015 test suite in Section 4. CEC 2015 test suite includes two unimodal test functions and 13 multimodal test functions. Note that, multimodal test functions with multiple local optimal solutions are more difficult to be solved compared with unimodal test functions. According to the experimental results presented in Table 1, MLNNA outperforms NNA on 80 percent of test functions, 67 percent of test functions, 67 percent of test functions, and 60 percent of test functions in terms of MAX, MEAN, MIN and STD, respectively. In addition, as can be seen from Fig. 6, MLNNA has obvious convergence advantages over NNA on more than half of the test functions. That is, MLNNA has better global search ability than NNA.

Section 5 discusses and analyzes the application of MLNNA in parameter estimation of PEMFC models. The BCW 500 W PEMFC model and the NedStack SP6 PEMFC model are considered in this work. Each model has seven unknown parameters to be estimated. The experimental results can be described as follows: (1) for BCW 500 W PEMFC model, from Table 4, MLNNA can get better solutions than the compared algorithms on the defined five value-based indicators, i.e. MAX, MEAN, MDN, MIN, and STD. Fig. 7 presents the characteristic curves based on the estimated optimal parameters by MLNNA, which demonstrate the estimated parameters have high accuracy. In addition, according to the results of the Friedman test shown in Table 7, MLNNA is the best of all the applied algorithms. Besides, as can be seen from Fig. 8, MLNNA displays better convergence ability than 10 compared algorithms; (2) for the NedStack PS6 PEMFC model, by observing Table 8, BSA can obtain the first place on MEAN, MDN, and MIN, which is only inferior to BSA on MAX and STD. According to the characteristic curves based on the estimated optimal parameters by MLNNA presented in Fig. 9, the extracted parameters have high precision. In addition, looking at Table 11, MLNNA is superior to the compared algorithms based on the Friedman test results. Besides, MLNNA can achieve a better convergence performance than 10 compared algorithms as shown in Fig. 10.

Based on the above discussion, the designed multiple learning strategies driven by the created local archive and global archive are very effective for improving the global search ability of NNA in solving complex nonlinear optimization problems.

7. Conclusion and further work

This work presents a new variant of neural network algorithm (NNA), named multiple learning neural network algorithm (MLNNA), for extracting the unknown parameters of proton exchange membrane fuel cell (PEMFC) models. In view of the high non-linearity of the objective function, it is a very challenging task to estimate the unknown parameters of PEMFC models. The core idea of MLNNA is to accelerate the convergence rate and enhance the global search ability of NNA by the designed multiple learning strategies. Further, the designed multiple learning strategies are based on the built local elite archive and global elite archive, which are introduced to bias operator and transfer operator. Note that, MLNNA only needs the essential population size and terminal condition for solving global optimization problems. The effectiveness of the improved strategies in MLNNA is evaluated by the well-known CEC 2015 test suite and two typical PEMFC models, i.e. BCS 500 W PEMFC model and the NedStack PS6 PEMFC model. Experimental results prove the excellent global search ability of MLNNA and the great potential of MLNNA to be applied to parameter extraction of PEMFC models.

Further work will focus on the following two aspects. On the one hand, although the excellent global search ability of MLNNA has been proven in the parameter extraction of proton exchange membrane fuel cell models, the performance of MLNNA on other practical engineering problems needs to be investigated according to No-Free-Lunch theorem. Thus, we plan to use MLNNA to solve more practical problems, such as energy efficient cluster head selection in wireless sensor networks, wind power prediction, and space nuclear reactor fuel design. On the other hand, the created local archive and global archive are the basis of MLNNA. However, when solving a large scale optimization problem, there is a large amount of data that needs to be moved in and out of the two archives frequently. This will cause a huge time overhead and reduces the computational efficiency. Therefore, how to improve the computational efficiency of MLNNA in solving large scale optimization problems is another research topic.

CrediT authorship contribution statement

Author A: Conceptualization, Methodology, Original draft; Author B: Supervision, Review & editing; Author C: Supervision, Conceptualization, Review & editing; Author D: Formal analysis, Data curation.

Data availability

The data and materials used to support the findings of this study are available from the corresponding author upon reasonable request.

Funding

This research was supported by the Postdoc Matching Fund Scheme (P0040875), the Hong Kong Polytechnic University.

Declaration of competing interest

The authors declare that there is no conflict of interest.

Acknowledgements

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] Ahmed SF, Mofijur M, Islam N, Parisa TA, Rafa N, Bokhari A, et al. Insights into the development of microbial fuel cells for generating biohydrogen, bioelectricity, and treating wastewater. Energy 2022;254:124163. https://doi.org/10.1016/j.energy.2022.124163.

[2] Al-Othman A, Tawalbeh M, Martis R, Dhou S, Orhan M, Qasim M, et al. Artificial intelligence and numerical models in hybrid renewable energy systems with fuel cells: advances and prospects. Energy Convers Manag 2022;253:115154. https://doi.org/10.1016/j.enconman.2021.115154.

[3] Yu Z, Feng C, Lai Y, Xu G, Wang D. Performance assessment and optimization of two novel cogeneration systems integrating proton exchange membrane fuel cell with

organic flash cycle for low temperature geothermal heat recovery. Energy 2022;243:122725. https://doi.org/10.1016/j.energy.2021.122725.

[4] Priya K, Rajasekar N. Application of flower pollination algorithm for enhanced proton exchange membrane fuel cell modelling. Int J Hydrogen Energy 2019;44:18438–49. https://doi.org/10.1016/j.ijhydene.2019.05.022.

[5] Santos DFM, Ferreira RB, Falcão DS, Pinto AMFR. Evaluation of a fuel cell system designed for unmanned aerial vehicles. Energy 2022;253:124099. https://doi.org/10.1016/j.energy.2022.124099.

[6] Wilberforce T, Abdelkareem MA, Elsaid K, Olabi AG, Sayed ET. Role of carbon-based nanomaterials in improving the performance of microbial fuel cells. Energy 2022;240:122478. https://doi.org/10.1016/j.energy.2021.122478.

[7] Baroutaji A, Arjunan A, Ramadan M, Robinson J, Alaswad A, Abdelkareem MA, et al. Advancements and prospects of thermal management and waste heat recovery of PEMFC. International Journal of Thermofluids 2021;9:100064. https://doi.org/10.1016/j.ijft.2021.100064.

[8] Rezk H, Olabi AG, Ferahtia S, Sayed ET. Accurate parameter estimation methodology applied to model proton exchange membrane fuel cell. Energy 2022:124454. https://doi.org/10.1016/j.energy.2022.124454.

[9] Kwon O, Kim J, Choi H, Cha H, Shin M, Jeong Y, et al. CNT sheet as a cathodic functional interlayer in polymer electrolyte membrane fuel cells. Energy 2022;245:123237. https://doi.org/10.1016/j.energy.2022.123237.

[10] El-Hay EA, El-Hameed MA, El-Fergany AA. Optimized Parameters of SOFC for steady state and transient simulations using interior search algorithm. Energy 2019;166:451–61. https://doi.org/10.1016/j.energy.2018.10.038.

[11] Nejad HC, Farshad M, Gholamalizadeh E, Askarian B, Akbarimajd A. A novel intelligent-based method to control the output voltage of Proton Exchange Membrane Fuel Cell. Energy Convers Manag 2019;185:455–64. https://doi.org/10.1016/j.enconman.2019.01.086.

[12] Zhao Y, Mao Y, Zhang W, Tang Y, Wang P. Reviews on the effects of contaminations and research methodologies for PEMFC. Int J Hydrogen Energy 2020;45:23174–200. https://doi.org/10.1016/j.ijhydene.2020.06.145.

[13] Ohenoja M, Leiviskä K. Observations on the parameter estimation problem of polymer electrolyte membrane fuel cell polarization curves. Fuel Cell 2020;20:516–26. https://doi.org/10.1002/fuce.201900155.

[14] Ashraf H, Abdellatif SO, Elkholy MM, El-Fergany AA. Honey badger optimizer for extracting the ungiven parameters of PEMFC model: steady-state assessment. Energy Convers Manag 2022;258:115521. https://doi.org/10.1016/j.enconman.2022.115521.

[15] El-Hay EA, El-Hameed MA, El-Fergany AA. Improved performance of PEM fuel cells stack feeding switched reluctance motor using multi-objective dragonfly optimizer. Neural Comput Appl 2019;31:6909–24. https://doi.org/10.1007/s00521-018-3524-z.

[16] Shaheen MAM, Hasanien HM, El Moursi MS, El-Fergany AA. Precise modeling of PEM fuel cell using improved chaotic MayFly optimization algorithm. Int J Energy Res 2021;45:18754–69. https://doi.org/10.1002/er.6987.

[17] Mohammadi A, Cirrincione G, Djerdir A, Khaburi D. A novel approach for modeling the internal behavior of a PEMFC by using electrical circuits. Int J Hydrogen Energy 2018;43:11539–49. https://doi.org/10.1016/j.ijhydene.2017.08.151.

[18] Giner-Sanz JJ, Ortega EM, Pérez-Herranz V. Mechanistic equivalent circuit modelling of a commercial polymer electrolyte membrane fuel cell. J Power Sources 2018;379:328–37. https://doi.org/10.1016/j.jpowsour.2018.01.066.

[19] Busquet S, Hubert CE, Labbé J, Mayer D, Metkemeijer R. A new approach to empirical electrical modelling of a fuel cell, an electrolyser or a regenerative fuel cell. J Power Sources 2004;134:41–8. https://doi.org/10.1016/j.jpowsour.2004.02.018.

[20] Han J, Han J, Ji H, Yu S. Model-based" design of thermal management system of a fuel cell "air-independent" propulsion system for underwater shipboard. Int J Hydrogen Energy 2020;45:32449–63. https://doi.org/10.1016/j.ijhydene.2020.08.233.

[21] Mann RF, Amphlett JC, Hooper MAI, Jensen HM, Peppley BA, Roberge PR. Development and application of a generalised steady-state electrochemical model for a PEM fuel cell. J Power Sources 2000;86:173–80. https://doi.org/10.1016/S0378-7753(99)00484-X.

[22] El-Fergany AA, Hasanien HM, Agwa AM. Semi-empirical PEM fuel cells model using whale optimization algorithm. Energy Convers Manag 2019;201:112197. https://doi.org/10.1016/j.enconman.2019.112197.

[23] Alizadeh M, Torabi F. Precise PEM fuel cell parameter extraction based on a self-consistent model and SCCSA optimization algorithm. Energy Convers Manag 2021;229:113777. https://doi.org/10.1016/j.enconman.2020.113777.

[24] Xu S, Wang Y, Wang Z. Parameter estimation of proton exchange membrane fuel cells using eagle strategy based on JAYA algorithm and Nelder-Mead simplex method. Energy 2019;173:457–67. https://doi.org/10.1016/j.energy.2019.02.106.

[25] Sultan HM, Menesy AS, Kamel S, Selim A, Jurado F. Parameter identification of proton exchange membrane fuel cells using an improved salp swarm algorithm. Energy Convers Manag 2020;224:113341. https://doi.org/10.1016/j.enconman.2020.113341.

[26] Rizk-Allah RM, El-Fergany AA. Artificial ecosystem optimizer for parameters identification of proton exchange membrane fuel cells model. Int J Hydrogen Energy 2021;46:37612–27. https://doi.org/10.1016/j.ijhydene.2020.06.256.

[27] Duan B, Cao Q, Afshar N. Optimal parameter identification for the proton exchange membrane fuel cell using Satin Bowerbird optimizer. Int J Energy Res 2019;43:8623–32. https://doi.org/10.1002/er.4859.

[28] Lu X, Kanghong D, Guo L, Wang P, Yildizbasi A. Optimal estimation of the proton exchange membrane fuel cell model parameters based on extended version of crow search algorithm. J Clean Prod 2020;272:122640. https://doi.org/10.1016/j.jclepro.2020.122640.

[29] Mossa MA, Kamel OM, Sultan HM, Diab AAZ. Parameter estimation of PEMFC model based on Harris Hawks' optimization and atom search optimization algorithms. Neural Comput Appl 2021;33:5555–70. https://doi.org/10.1007/s00521-020-05333-4.

[30] Zhu Y, Yousefi N. Optimal parameter identification of PEMFC stacks using adaptive sparrow search algorithm. Int J Hydrogen Energy 2021;46:9541–52. https://doi.org/10.1016/j.ijhydene.2020.12.107.

[31] Yuan Z, Wang W, Wang H. Optimal parameter estimation for PEMFC using modified monarch butterfly optimization. Int J Energy Res 2020;44:8427–41. https://doi.org/10.1002/er.5527.

[32] Diab AAZ, Tolba MA, El-Magd AGA, Zaky MM, El-Rifaie AM. Fuel cell parameters estimation via marine predators and political optimizers. IEEE Access 2020;8:166998–7018. https://doi.org/10.1109/ACCESS.2020.3021754.

[33] Seleem SI, Hasanien HM, El-Fergany AA. Equilibrium optimizer for parameter extraction of a fuel cell dynamic model. Renew Energy 2021;169:117–28. https://doi.org/10.1016/j.renene.2020.12.131.

[34] Yang Z, Liu Q, Zhang L, Dai J, Razmjooy N. Model parameter estimation of the PEMFCs using improved Barnacles Mating Optimization algorithm. Energy 2020;212:118738. https://doi.org/10.1016/j.energy.2020.118738.

[35] Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Trans Evol Comput 1997;1:67–82. https://doi.org/10.1109/4235.585893.

[36] Singh D, Dhillon JS. Ameliorated grey wolf optimization for economic load dispatch problem. Energy 2019;169:398–419. https://doi.org/10.1016/j.energy.2018.11.034.

[37] Sun X, Wang G, Xu L, Yuan H, Yousefi N. Optimal estimation of the PEM fuel cells applying deep belief network optimized by improved archimedes optimization algorithm. Energy 2021;237:121532. https://doi.org/10.1016/j.energy.2021.121532.

[38] Sadollah A, Sayyaadi H, Yadav A. A dynamic metaheuristic optimization model inspired by biological nervous systems: neural network algorithm. Appl Soft Comput 2018;71:747–82. https://doi.org/10.1016/j.asoc.2018.07.039.

[39] Fontes G, Turpin C, Astier S, Meynard TA. Interactions between fuel cells and power converters: influence of current harmonics on a fuel cell stack. IEEE Trans Power Electron 2007;22:670–8. https://doi.org/10.1109/TPEL.2006.890008.

[40] El-Fergany AA. Extracting optimal parameters of PEM fuel cells using Salp Swarm Optimizer. Renew Energy 2018;119:641–8. https://doi.org/10.1016/j.renene.2017.12.051.

[41] Rakhshani H, Rahati A. Snap-drift cuckoo search: a novel cuckoo search optimization algorithm. Appl Soft Comput 2017;52:771–94. https://doi.org/10.1016/j.asoc.2016.09.048.

[42] Yu K, Liang JJ, Qu BY, Cheng Z, Wang H. Multiple learning backtracking search algorithm for estimating parameters of photovoltaic models. Appl Energy 2018;226:408–22. https://doi.org/10.1016/j.apenergy.2018.06.010.

[43] Deng W, Xu J, Gao X-Z, Zhao H. An enhanced MSIQDE algorithm with novel multiple strategies for global optimization problems. IEEE Trans Syst Man Cybern, Syst 2020:1–10. https://doi.org/10.1109/TSMC.2020.3030792.

[44] Gupta S, Deep K. A hybrid self-adaptive sine cosine algorithm with opposition based learning. Expert Syst Appl 2019;119:210–30. https://doi.org/10.1016/j.eswa.2018.10.050.

[45] Gupta S, Deep K, Heidari AA, Moayedi H, Wang M. Opposition-based learning Harris hawks optimization with advanced transition rules: principles and analysis. Expert Syst Appl 2020;158:113510. https://doi.org/10.1016/j.eswa.2020.113510.

[46] Shekhawat S, Saxena A. Development and applications of an intelligent crow search algorithm based on opposition based learning. ISA (Instrum Soc Am) Trans 2020;99:210–30. https://doi.org/10.1016/j.isatra.2019.09.004.

[47] Dhargupta S, Ghosh M, Mirjalili S, Sarkar R. Selective opposition based grey wolf optimization. Expert Syst Appl 2020;151:113389. https://doi.org/10.1016/j.eswa.2020.113389.

[48] Rahnamayan S, Tizhoosh HR, Salama MMA. Opposition-based differential evolution. IEEE Trans Evol Comput 2008;12:64–79. https://doi.org/10.1109/TEVC.2007.894200.

[49] Tian M, Gao X. Differential evolution with neighborhood-based adaptive evolution mechanism for numerical optimization. Inf Sci 2019;478:422–48. https://doi.org/10.1016/j.ins.2018.11.021.

[50] Zamani H, Nadimi-Shahraki MH, Gandomi AH. CCSA: conscious neighborhood-based crow search algorithm for solving global optimization problems. Appl Soft Comput 2019:105583. https://doi.org/10.1016/j.asoc.2019.105583.

[51] Liang J, Qu B, Suganthan P, Chen Q. Problem definitions and evaluation criteria for the CEC 2015 competition on learning-based real-parameter single objective optimization. Singapore: Technical Report201411A, Computational Intelligence Laboratory, Zhengzhou University; 2014. Zhengzhou China and Technical Report, Nanyang Technological University.

[52] Hachana O. Accurate PEM fuel cells parameters estimation using hybrid artificial bee colony differential evolution shuffled complex optimizer. Int J Energy Res 2022;46:6383–405. https://doi.org/10.1002/er.7576.

[53] Mirjalili S, Lewis A. The whale optimization algorithm. Adv Eng Software 2016;95:51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008.

[54] Dhiman G, Kumar V. Seagull optimization algorithm: theory and its applications for large-scale industrial engineering problems. Knowl Base Syst 2019;165:169–96. https://doi.org/10.1016/j.knosys.2018.11.024.

[55] Qais MH, Hasanien HM, Alghuwainem S. Transient search optimization: a new meta-heuristic optimization algorithm. Appl Intell 2020;50:3926–41. https://doi.org/10.1007/s10489-020-01727-y.

[56] Civicioglu P. Backtracking search optimization algorithm for numerical optimization problems. Appl Math Comput 2013;219:8121–44. https://doi.org/10.1016/j.amc.2013.02.017.

[57] Kaur S, Awasthi LK, Sangal AL, Dhiman G. Tunicate Swarm Algorithm: a new bio-inspired based metaheuristic paradigm for global optimization. Eng Appl Artif Intell 2020;90:103541. https://doi.org/10.1016/j.engappai.2020.103541.

[58] Rao R. Int J Ind Eng Comput 2016;7:19–34.

[59] Dhiman G, Kaur A. STOA: a bio-inspired based optimization algorithm for industrial engineering problems. Eng Appl Artif Intell 2019;82:148–74. https://doi.org/10.1016/j.engappai.2019.03.021.

[60] Mirjalili S. SCA: a Sine Cosine Algorithm for solving optimization problems. Knowl Base Syst 2016;96:120–33. https://doi.org/10.1016/j.knosys.2015.12.022.

[61] Zhang Y. Chaotic neural network algorithm with competitive learning for global optimization. Knowl Base Syst 2021;231:107405. https://doi.org/10.1016/j.knosys.2021.107405.

[62] Houssein EH, Saber E, Ali AA, Wazery YM. Centroid mutation-based Search and Rescue optimization algorithm for feature selection and classification. Expert Syst Appl 2022;191:116235. https://doi.org/10.1016/j.eswa.2021.116235.

[63] Gong S-P, Khishe M, Mohammadi M. Niching chimp optimization for constraint multimodal engineering optimization problems. Expert Syst Appl 2022;198:116887. https://doi.org/10.1016/j.eswa.2022.116887.

[64] Wu J, Li N, Zhao Y, Wang J. Usage of correlation analysis and hypothesis test in optimizing the gated recurrent unit network for wind speed forecasting. Energy 2022;242:122960. https://doi.org/10.1016/j.energy.2021.122960.

[65] Hou G, Gong L, Hu B, Su H, Huang T, Huang C, et al. Application of fast adaptive moth-flame optimization in flexible operation modeling for supercritical unit. Energy 2022;239:121843. https://doi.org/10.1016/j.energy.2021.121843.