



Deep learning-accelerated optimization algorithm for controller parameters optimization of doubly-fed induction generators



Linfei Yin ^{a,c,d}, Xinghui Cao ^{a,c,d}, Senlin Wang ^{b,*}

^a School of Electrical Engineering, Guangxi University, Nanning, Guangxi, 530004, China

^b Quanzhou Institute of Equipment Manufacturing Haixi Institutes, Chinese Academy of Science, Quanzhou, 362200, China

^c Institute of Artificial Intelligence, Guangxi University, Nanning, Guangxi, 530004, China

^d Guangxi Key Laboratory of Intelligent Control and Maintenance of Power Equipment, Guangxi University, Nanning, 530004, China

ARTICLE INFO

Article history:

Received 31 July 2022

Received in revised form 26 September 2022

Accepted 2 November 2022

Available online 11 November 2022

Keywords:

Deep fully connected models

Gray wolf optimizer

Adaptive differential evolution

Global search

Parameter optimization

ABSTRACT

In this work, a cooperative Gray wolf Optimizer with adaptive differential Evolution (GOE) is proposed for the multimodal controller parameters optimization of doubly-fed induction generators (DFIGs) based on maximum power point tracking (MPPT) strategies. Moreover, the optimization process of the GOE is accelerated by a deep fully connected model (DFCM). The GOE contains a cooperative gray wolf optimizer (GWO) and adaptive differential evolution (ADE). The cooperative GWO contains alpha, beta, delta, and omega wolves to explore and exploit optimization problems and achieves optimization tasks wider and deeper than GWO. The ADE cooperates with the cooperative GWO to solve global optimization over continuous spaces. The simulation results on seven uni-model benchmark functions show that the GOE accelerated by DFCM obtains acceptable fitness values with 39.99% lesser computation time than the symmetry adapted stochastic search (SASS) algorithm and 80.72% lesser computation time than the Lévy flights-success-history based adaptive differential evolution with constraint handling technique (COLSHADE) algorithm, which are the winners of the CEC2020 Competition on Real-World Single Objective Constrained Optimization. Furthermore, the simulation results on DFIG with MPPT strategies in three real-world cases verify that the GOE accelerated by DFCM can effectively obtain global optimization solutions for non-smooth problems with 99.51% lesser average computation time than the SASS algorithm, 99.63% less than the COLSHADE algorithm, and 89.52% less than other methods. In addition, the accelerated GOE algorithm by DFCM has the feature of faster convergence.

© 2022 Elsevier B.V. All rights reserved.

1. Introduction

One of the important ways to achieve carbon neutrality in energy is using renewable energy security [1]. The exploitation and application of fossil fuels are the major reasons for global warming and air pollution, which attract numerous researchers to pay attention to carbon neutrality [2]. Hydropower generating sets, wind, and photovoltaic power are major renewable energy sources [3]. In addition, supplying power to rural places through long-distance electricity transmission lines is uneconomical with the high costs of grid expansion, rugged terrain [4], and unstable power in remote areas [5]. Hence, renewable sources are increasingly popular in satisfying electricity requirements [6]. Furthermore, renewable energy, such as wind power, reduces the dependence on fossil fuels in many countries [7]. One energy conversion is wind turbine (WT), such as doubly-fed induction

generators (DFIGs) [8], permanent magnetic synchronous generators [9], and advanced energy converters with lower-cost of energy converters [10].

Maximum power point tracking (MPPT) technology maximizes the mechanical energy of WTs for the wind speed (WS) varies [11]. MPPT systems with extensive nonlinearities and uncertainties are significant subjects for DFIG-based WTs [12]. Two approaches are possible to achieve MPPT.

The first way to achieve MPPT is to design a controller with excellent performance. For example, a fractional-order proportional-integral-derivative (PID) has been developed [13]. In addition, fuzzy and PID controllers have been combined to obtain MPPT [14]. Moreover, Elman neural network was proposed to obtain a quick response for MPPT [15]. Feedback linearization controllers have been achieved to obtain the global optimal energy tracking from different WSs [16]. Sliding-mode control has been applied to avoid accurate measurement for MMPT [17]. Besides, a nonlinear strategy has been presented for MPPT backstepping techniques with proof based on Lyapunov theory [18]. In addition, model predictive control with future prediction ability has been designed

* Corresponding author.

E-mail address: senlin16888@fjirsm.ac.cn (S. Wang).

Nomenclature	
Variables	
$\alpha, \beta, \delta, \omega$	Four types of gray wolves
β_{DFIG}	Pitch angle
b_{DFCM}	DFCM offsets
C_p	Power coefficient
e_{RMSE}	RMSE
$F_{i,g}$	Adaptive scaling factor
$f'_g, f_{\text{avg},g}, \text{ and } f_{\text{best},g}$	Individual, average, and global fitness values
$H_{t,\text{DFIG}}$ and $H_{g,\text{DFIG}}$	Inertia constants of WT and DFIG
$L_s, L_t, \text{ and } L_m$	Stator, rotor, and magnetizing inductances
N_p	Population size
n_{10}	Number values for evaluation
$w_b, w_s, \text{ and } w_r$	Electrical base, synchronous angle, and rotor angle speeds
w_m	Mechanical speed
$w_{m,\text{DFIG}}$	Rotational speed
w_{DFCM}	DFCM weights
P_m, P_e, P_s	Mechanical, electrical, active power
Q_s	Reactive power
R	Radius
R_s and R_r	Stator and rotor resistances
$\vec{\tau}_{\text{IGWO},1} \text{ and } \vec{\tau}_{\text{IGWO},2}$	Random numbers
ρ	Air density
t_m	Maximum iteration
$T_{e,\text{DFIG}}$	Electrical torque
$T_{m,\text{DFIG}}$	Mechanical torque
v_{wind}	Wind speed
\vec{X}_{IGWO}	Wolf position
X_p	Parent population
X_i	Child population

Abbreviations	
ADE	Adaptive differential evolution
COLSHADE	Lévy flights-success-history based adaptive differential evolution with constraint handling technique
CGWO	Cooperative GWO
DFIGs	Doubly-fed induction generators
DFCM	Deep fully connected model
DRPG	Drop change of the power grid voltage
GOE	CGWO with ADE
GWO	Gray wolf optimizer
MFO	Moth-flame optimization
MPPT	Maximum power point tracking
PID	Proportional–integral–derivative
PI	Proportional–integral
PSO	Particle swarm optimization
RSCs	Rotor side converters
RMSE	Root mean square error
SASS	Symmetry adapted stochastic search
SIWS	Step increase change of WS
SWWS	Sawtooth wave changes in WS
WS	Wind speed
WT	Wind turbine

to obtain MPPT [19]. Besides, quantum technology and deep reinforcement learning have been combined to obtain MPPT [20]. Multiple controllers of a WT have been coordinated for low cost and high control performances [21]. Although these designed controllers can have excellent control performances, these controllers still need much training before these controllers can be used [20].

The parameter optimization of a widely used controller is another way to achieve MPPT, such as PID controllers. Generally, a DFIG contains four proportional–integral (PI) controllers that need to be optimized [22]. Four approaches can be applied to adjust the parameters of four PIs of DFIG.

- (1) The manual trial-and-error method is a direct method to obtain initial variable ranges for the other three approaches, such as the controller parameters ranges [23]. However, the trial-and-error method is almost impossible to reach accurate control performances. Moreover, this trial-and-error method requires considerable labor and effort and is prone to be wrong.
- (2) Ziegler–Nichols can be utilized for industrial applications, such as position servo systems [24] and ore mining [25]. However, this method relies on the exact system transfer functions. Therefore, this method is well-suited for theoretical analysis rather than practical systems with small parameter differences.

(3) Optimization algorithms can be utilized to optimize controller parameters. For instance, particle swarm optimization (PSO) [26], parallel Monte Carlo [27], controller-based methods [28], gray wolf optimizer (GWO) [29], adaptive differential evolution (ADE) [30]. The process of PI controller parameters being optimized consumes substantial computational time. For example, the time for a single run of DFIG simulation is ten seconds. An optimization algorithm with 200 iterations and a population size of 100 needs 2.31 days to optimize a set of parameters. These optimization algorithms generally need to be run ten times to resist the stochastic property, which will take 23.15 days. Although these optimization algorithms can obtain a higher performance set of controller parameters, they are too time-consuming. These optimization algorithms need more time to optimize the DFIG controller parameters if the initialized upper and lower limit parameters are not chosen reasonably.

(4) This work proposes a novel method to optimize the DFIG controller parameters. In this work, an artificial intelligence approach is adopted to accelerate the optimization process of the optimization algorithm that does not depend on DFIG transfer functions. The deep fully connected models (DFCMs) have a great prediction or forecasting ability in numerous fields, such as load forecasting [31] and wind power output prediction [32]. Therefore, this work adopts DFCM to accelerate the optimization process of an optimization algorithm. Consequently, this accelerated optimization algorithm has high accuracy and does not depend on DFIG transfer functions.

With flexibility preponderances: (i) competent development and discovery; (ii) local optimal avoidance; (iii) flexibility behaviors in an unknown environment, GWO is an important method for global optimization for various systems [33], such as several distributed generation problems [34], and complex real-world problems [35]. With guidance/difference mechanisms, the ADE

has better optimization performance, such as multimodal problems [36] and local binary pattern-based problems [37]. Therefore, the advantages of GWO and ADE are combined in this work for practical engineering optimization problems. This work suggests a novel scheme named cooperative GWO (CGWO) with ADE (GOE) to obtain global maximized solutions with stability.

However, the GOE, which contains two optimization approaches, needs more computation time than the single optimization method. Therefore, deep learning with representation ability is applied to reduce the computation time in this work. Although the prediction accuracies of these DFCMs are not 100% [38], the inputs and outputs of fitness functions can be learned by DFCMs; thus, the optimization process of the GOE is accelerated by DFCMs in this work. This work contains the following contributions after the GOE accelerated by DFCMs is applied for DFIG.

- (1) This work presents a novel way of thinking about optimization methods. To the best of the authors' knowledge, artificial intelligence methods are applied for the first time to accelerate the optimization process of optimization algorithms. Furthermore, the model of optimization algorithm accelerated by machine learning is created in this work. As a result, the process of optimization algorithms is accelerated, the time consumed by optimization algorithms is reduced, and the iterative convergence process is accelerated.
- (2) After machine learning accelerates the intermediate process, the optimization algorithm is still utilized to optimize the parameters of DFIG controllers. Thus, the proposed accelerated optimization algorithm does not require a 100% accurate machine-learning algorithm to accelerate the optimization process. Thus, to the best of the authors' knowledge, (a) machine learning methods are employed for the first time to accelerate optimization algorithms; (b) machine learning algorithms do not need to be 100% accurate; (c) the deficiencies of machine learning algorithms that are not 100% accurate and optimization algorithms that are slow are solved simultaneously.
- (3) The GOE contains a CGWO and an ADE. The CGWO aims to balance the exploration and exploitation of parameter optimization problems. The ADE considers the correlation between multiple variables to a certain extent. In addition, the ADE considers the correlation between multiple controller parameters.
- (4) Multiple controller parameters are simultaneously optimized by the combined GOE. Each controller games with the other through the control system. The combined GOE can obtain better performance when controller parameters are simultaneously optimized.
- (5) After a DFCM of machine learning is utilized to replace parts of the optimization processes of the proposed GOE, the GOE computation time is decreased. Four group PI parameters of DFIG are optimized by the GOE with DFCMs simultaneously with better performance, lower control costs, and lesser computation time. As a result, the GOE with DFCMs can promote carbon neutrality with the higher control performance of WTs in a short optimization time.

This work applies the GOE to obtain PI factors of DFIG with MPPT. The work includes the following parts: the modeling of WTs (Section 2) and the proposed accelerated GOE (Section 3); in Section 3, the GOE for the optimum PI tuning of DFIG for MPPT is described; in Section 4, the result of the accelerated GOE is carried out and compared with other algorithms; the main conclusion and remarks for the accelerated GOE are summarized in Section 5.

2. Plant and fitness function of parameters optimization of DFIG controllers

The PI controller parameters determine the control performance of DFIG. The control error is reduced, and the objective function value is reduced if the optimization algorithm optimizes the PI parameters.

2.1. DFIG control plant

A DFIG-WT includes grid and rotor side converters (RSCs), a mechanical shaft system, a WT module, and a DFIG system [39] (Fig. 1). The generator of the shaft system contacts the WT. The equation of a generator-wind turbine system [40] is

$$T_{m,DFIG} - T_{e,DFIG} = Jdw_r/dt \quad (1)$$

where $T_{e,DFIG}$ and $T_{m,DFIG}$ are electrical and mechanical torques, respectively; w_r represents rotor speed; J represents WT system parameter. The relationship between mechanical P_m and electrical power P_e is

$$P_m - P_e = w_m J dw_m/dt \quad (2)$$

where w_m means mechanical speed. Generally, mechanical power is

$$P_m(\beta_{DFIG}) = \rho \pi R^2 C_p v_{wind}^3 / 2 \quad (3)$$

where R , ρ and v_{wind} are the radius of WT, air density, and WS, respectively; C_p represents power coefficient which relies on pitch angle β_{DFIG} as

$$\begin{aligned} C_p = & (60.0416/\lambda_i - 0.20704\beta_{DFIG} - 2.588)e^{-21/\lambda_i} \\ & + 0.0068Rw_{m,DFIG}/V_w, \end{aligned} \quad (4)$$

where $w_{m,DFIG}$ is the rotational speed. The DFIG model is [41]

$$\left\{ \begin{array}{l} v_{qs} = \frac{w_b}{L_s}(-R_1 i_{qs} + w_s L'_s i_{qs} + \frac{w_r}{w_s} e'_{qs} \\ \quad - \frac{1}{T_r w_s} e'_{ds} - \frac{L'_s di_{qs}}{w_b dt} + \frac{L_m}{L_{rr}} v_{qr}) \\ v_{ds} = \frac{w_b}{L_s}(-R_1 i_{qs} - w_s L'_s i_{qs} + \frac{w_r}{w_s} e'_{ds} \\ \quad + \frac{1}{T_r w_s} e'_{qs} - \frac{L'_s di_{ds}}{w_b dt} + \frac{L_m}{L_{rr}} v_{qr}) \\ e'_{qs} = \int w_b w_s [R_2 i_{ds} - \frac{1}{T_r w_s} e'_{qs} + (1 - \frac{w_r}{w_s} e'_{ds} - \frac{L_m}{L_{rr}} v_{qr})] dt \\ e'_{ds} = \int w_b w_s [-R_2 i_{ds} - \frac{1}{T_r w_s} e'_{qs} - (1 - \frac{w_r}{w_s} e'_{ds} + \frac{L_m}{L_{rr}} v_{qr})] dt \end{array} \right. \quad (5)$$

where w_b , w_s , and w_r are the electrical base, synchronous angle, and rotor angle speeds, respectively; L_s , L_r , and L_m mean stator, rotor, and magnetizing inductances, respectively; i_{dr} , i_{qr} , i_{ds} , and i_{qs} are current; λ and λ_{opt} are tip-speed-ratio and its optimized value, respectively; ω_b and ω_s mean electrical and synchronous angle speeds, respectively; R_s and R_r mean stator and rotor resistances, respectively.

The electromagnetic and mechanical torques (i.e., $T_{e,DFIG}$ and $T_{m,DFIG}$) respectively are

$$T_{e,DFIG} = (e'_{qs}/w_s)i_{qs} + (e'_{ds}/w_s)i_{ds} \quad (6)$$

$$T_{m,DFIG} = P_m/w_{m,DFIG} \quad (7)$$

The active power P_s is

$$P_s = v_{qs}i_{ds} - v_{ds}i_{qs} \quad (8)$$

Generally, the v_{ds} is neglected in MPPT systems [41]. Hence the P_s is

$$P_s = v_{qs}i_{ds} \quad (9)$$

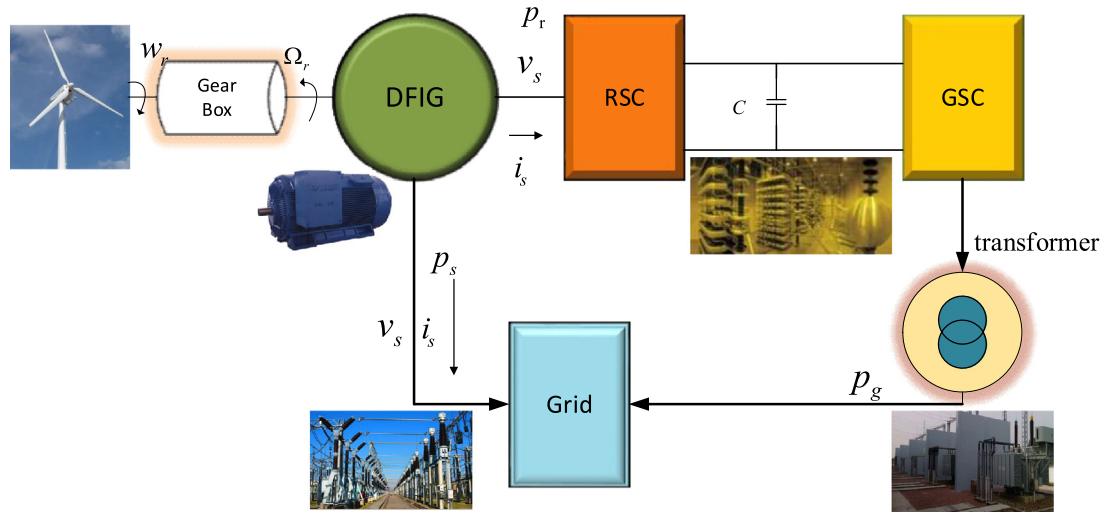


Fig. 1. DFIG-WT systems.

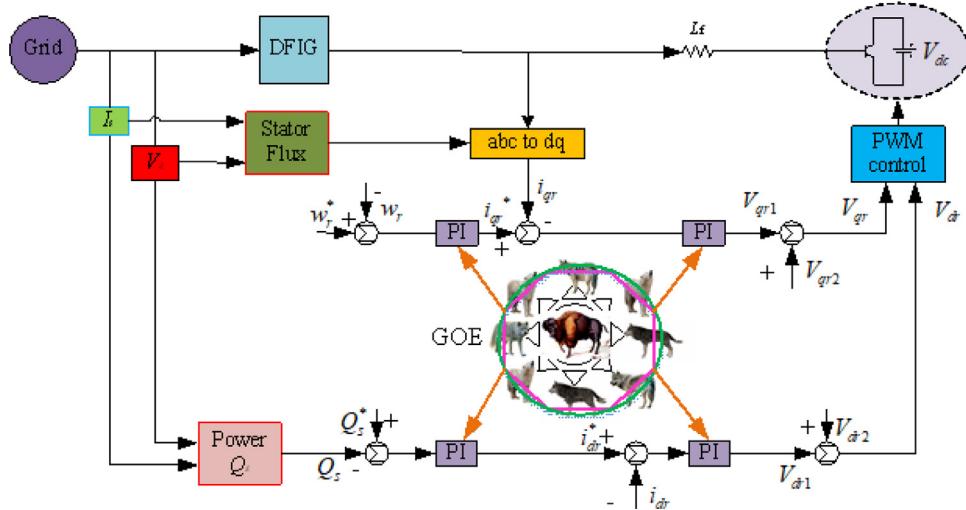


Fig. 2. Control framework of the proposed model.

The electromechanical dynamics is [41],

$$\begin{aligned} \frac{d\omega_m, \text{DFIG}}{dt} = & (P_m/\omega_m, \text{DFIG} - T_e, \text{DFIG} \\ & - D_{\text{DFIG}}\omega_m, \text{DFIG})/2(H_t, \text{DFIG} + H_g, \text{DFIG}) \end{aligned} \quad (10)$$

where ω_m, DFIG is rotational speed; H_t, DFIG and H_g, DFIG are inertia constants of WT and DFIG, respectively.

2.2. Fitness function of parameters optimization of DFIG controllers

Four inner PI loops are employed to obtain the best performance (Fig. 2). In the outer controlling loops, i_{qr}^* and i_{dr}^* are reference values. The outputs of final controllers v_{qr2} and v_{dr2} as

$$\begin{cases} v_{qr2} = \frac{w_s - w_r}{w_s} w_s ((1 - \frac{L_m^2}{L_s L_r}) L_r i_{dr} + \frac{L_m^2 (v_{qs} - R_s i_{qs})}{w_s L_m L_s}) \\ v_{dr2} = -\frac{w_s - w_r}{w_s} w_s (1 - \frac{L_m^2}{L_s L_r}) L_r i_{qr} \end{cases} \quad (11)$$

The optimal objective for a PI controller-based MPPT system model is

$$\begin{aligned} f(x) = & \sum \int_0^T (0.2 |Q_s - Q_s^*| + 0.004 |w_r - w_r^*| \\ & + w_1 |u_{qr}| + w_2 |u_{dr}|) dt \end{aligned}$$

$$\begin{aligned} & k_{pi \min} < k_{pi} < k_{pi \max} \\ & k_{li \min} < k_{li} < k_{li \max} \\ \text{s.t. } & \begin{cases} v_{wind \min} < v_{wind} < v_{wind \max}, \quad i = 1, 2, 3, 4. \\ u_{qr \min} < u_{qr} < u_{qr \max} \\ u_{dr \min} < u_{dr} < u_{dr \max} \\ Q_{s \min} < Q_s < Q_{s \max} \end{cases} \end{aligned} \quad (12)$$

where v_{wind} is wind speed; V_s is power grid voltage; Q_s is reactive power; T is the system operation sampling time. The GOE includes three guiding coefficients, including a , b , and c , which limit $[0, 1]$. The maximum iteration t_m is set to 100 to reduce the time to execute all algorithms properly. The system parameters of w_1 and w_2 are control costs parameters. Considering the friction factor, w_1 and w_2 are set to 1/16.

Eq. (12) considers the systemic reactive power Q_s , rotor speed w_r , and control costs $w_1 |u_{qr}| + w_2 |u_{dr}|$. The optimized PI controller parameters provide higher control performances with smaller control costs.

The optimization algorithm gives eight variables to be optimized during a certain iteration for a DFIG controller parameter optimization problem. These eight optimized variables are four

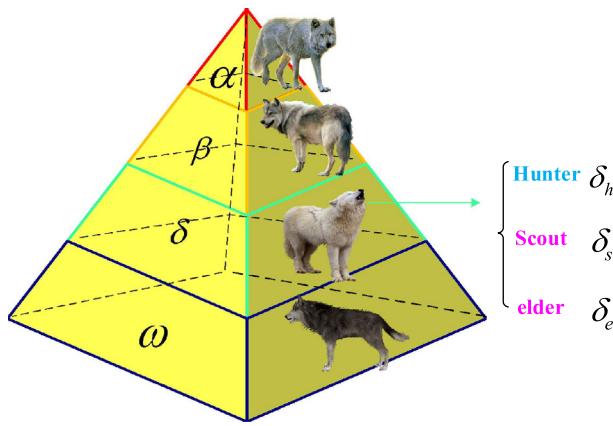


Fig. 3. Leadership hierarchy of gray wolves of GOE.

sets of PI parameters. The systemic reactive power is obtained after the Simulink simulation is completed. The objective function values of parameter optimization problems are achieved by Eq. (12) after system reactive power is obtained. Consequently, a set of PI parameters and objective function values are obtained.

3. Combined optimization accelerated by deep fully connected models

The proposed GOE contains CGWO and ADE for coordinated parameters optimization. Then, the GOE is combined with DFCMs of machine learning to reduce the calculation time of optimization processes.

3.1. Cooperative gray wolf optimizer

Compared with GWO, the CGWO is a stochastic algorithm that interacts with the random environment. Moreover, the CGWO has wider and deeper cooperation for optimization problems with global optimum searching, flexible leadership hierarchy, and remarkable performance in unknown environments. The major features of CGWO contain three points:

- (1) The gray wolves, which consist of four types (i.e., α , β , δ , and ω) aiming to mimic their behaviors, are grouped into three independent teams, i.e., the header team of alpha, the cooperative hunting team of beta, the team of omega and a random search team delta, respectively.
- (2) In cooperative hunting teams, delta wolves consist of three groups, i.e., the random scout, elder, and hunt groups. According to the fitness values in unknown environments, these groups obtain global solutions over six GWO. Thus, a more proper balance between discovery and development can be gained.
- (3) The header is primarily in charge of hunting wolves. Messages before hunting are given to alpha wolves. Meanwhile, older delta wolves have enough experiences to help the header in the cooperative hunting teams. Therefore, the global search ability is improved for deeper exploitation and hunting.

The primal gray wolves are at the peak of the food chain. Wolves are divided into four types, i.e., α , β , δ and ω wolves. The wolf α is the group leader, making decisions mostly responsible for hunting and waking up. The wolf β is subordinate which assists the wolf α in making decisions. The wolf ω is the lowest ranking and has to submit to other wolves. The wolf δ , which submits α and β wolves, includes scouts, elders, and hunters. The

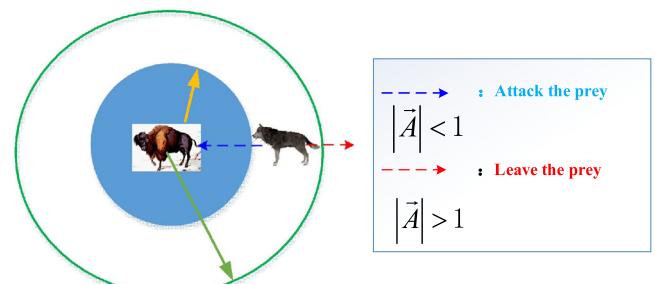


Fig. 4. Attracting or leaving for prey.

subscripts s , e and h stand for scout wolves, elder wolves, and hunt wolves, i.e., δ_s , δ_e , and δ_h , respectively (Fig. 3).

Compared with the GWO, the CGWO has a cooperative hunting team as

$$\vec{D}_{IGWO} = \left| \vec{C}_{IGWO} \vec{X}_{IGWO,p}(n) - \vec{X}_{IGWO}(n) \right| \quad (13)$$

$$\vec{X}_{IGWO}(n+1) = \vec{X}_{IGWO,p}(n) - \vec{A}_{IGWO} \vec{D}_{IGWO} \quad (14)$$

where n indicates iterations; $\vec{X}_{IGWO,p}$ indicates the prey position; \vec{X}_{IGWO} indicates the position of each wolf; \vec{A}_{IGWO} and \vec{C}_{IGWO} are coefficients as

$$\vec{A}_{IGWO} = 2 \vec{d}_{IGWO} \vec{r}_{IGWO,1} - \vec{d}_{IGWO} \quad (15)$$

$$\vec{C}_{IGWO} = 2 \vec{r}_{IGWO,2} \quad (16)$$

where $\vec{r}_{IGWO,1}$ and $\vec{r}_{IGWO,2}$ are random numbers, limited in $[0, 1]$; component \vec{d}_{IGWO} is the encircling random coefficient. If $|\vec{A}_{IGWO}| < 1$, gray wolves are forced to attract the prey; if $|\vec{A}_{IGWO}| > 1$, wolves diverge from the global goal to find fitter prey (Fig. 4).

Hunt wolves consist of the α , β , and δ wolves. The ω wolves are guided to update their locations to obtain the accurate location of potential prey (Fig. 5).

To mathematically simulate well-organized hunting, α , β , and δ_h wolves are advised to cooperate in hunting (Fig. 6). The locations are updated as

$$\left\{ \begin{array}{l} \vec{D}_\alpha = \left| \vec{C}_1 \vec{X}_\alpha - \vec{X}_{IGWO} \right| \\ \vec{D}_{\beta 1} = \left| \vec{C}_2 \vec{X}_{\beta 1} - \vec{X}_{IGWO} \right| \\ \vec{D}_{\beta 2} = \left| \vec{C}_2 \vec{X}_{\beta 2} - \vec{X}_{IGWO} \right| \\ \vec{D}_{\delta 1} = \left| \vec{C}_3 \vec{X}_{\delta_h 1} - \vec{X}_{IGWO} \right| \\ \vec{D}_{\delta 2} = \left| \vec{C}_3 \vec{X}_{\delta_h 2} - \vec{X}_{IGWO} \right| \\ \vec{D}_{\delta 3} = \left| \vec{C}_3 \vec{X}_{\delta_h 3} - \vec{X}_{IGWO} \right| \end{array} \right. \quad (17)$$

$$\left\{ \begin{array}{l} \vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \vec{D}_\alpha \\ \vec{X}_{21} = \vec{X}_{\beta 1} - \vec{A}_2 \vec{D}_{\beta 1} \\ \vec{X}_{22} = \vec{X}_{\beta 2} - \vec{A}_2 \vec{D}_{\beta 2} \\ \vec{X}_{31} = \vec{X}_{\delta_h 1} - \vec{A}_3 \vec{D}_{\delta 1} \\ \vec{X}_{32} = \vec{X}_{\delta_h 2} - \vec{A}_3 \vec{D}_{\delta 2} \\ \vec{X}_{33} = \vec{X}_{\delta_h 3} - \vec{A}_3 \vec{D}_{\delta 3} \end{array} \right. \quad (18)$$

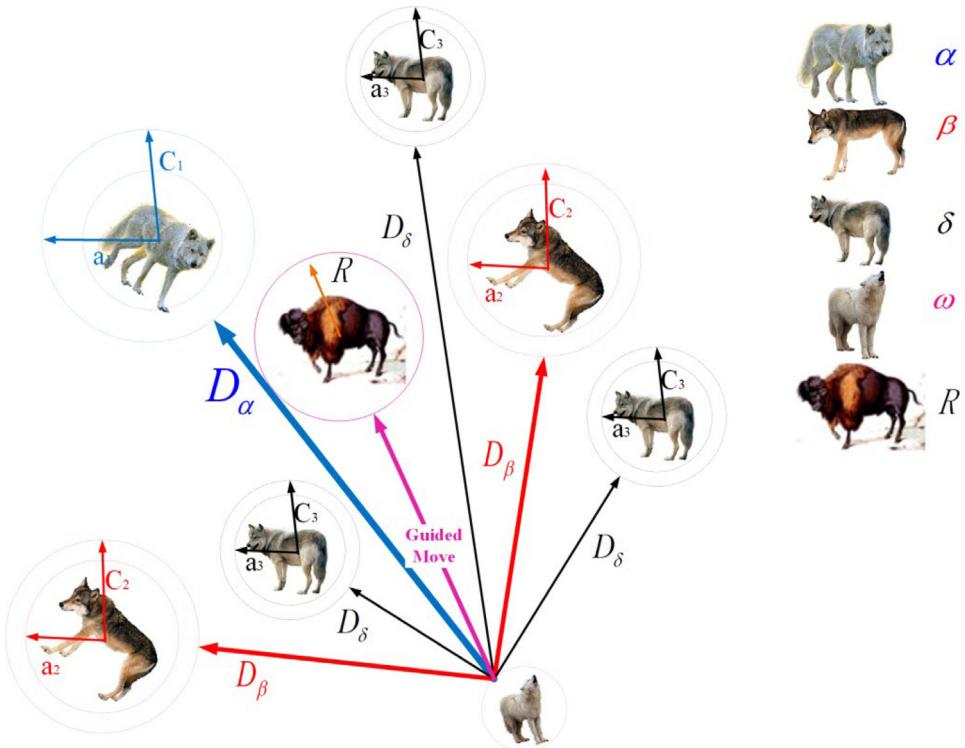
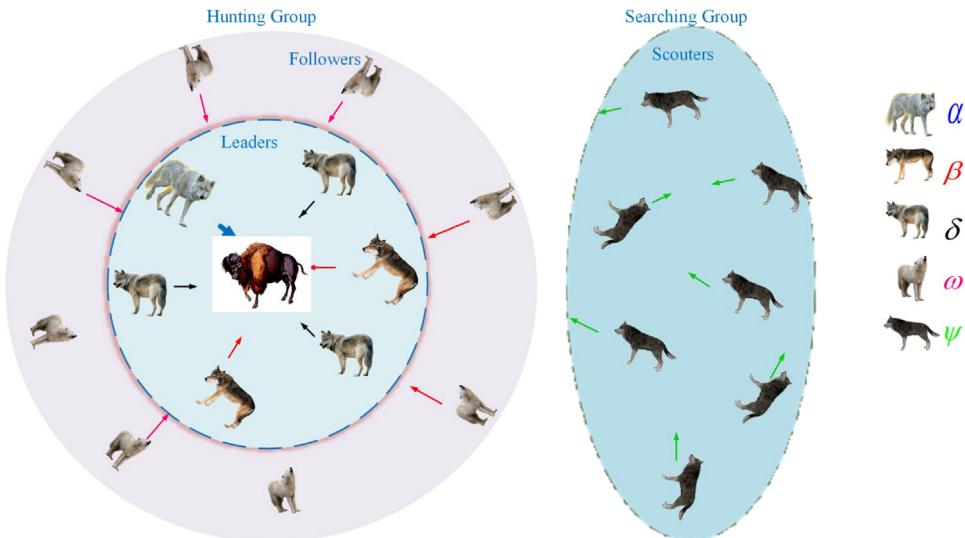
Fig. 5. Random location of ω wolves of GOE.

Fig. 6. Searching behavior of random scouts for deep exploration.

$$\begin{cases} \vec{X}_{IGWO}(n+1) = k_\alpha \vec{X}_1 + k_\beta \left(\frac{\vec{X}_{21} + \vec{X}_{22}}{2} \right) \\ + k_{\delta_h} \left(\frac{\vec{X}_{31} + \vec{X}_{32} + \vec{X}_{33}}{3} \right) \\ \sum_{i=\alpha, \beta, \delta_h} k_i = 1, k_i \geq 0 \end{cases} \quad (19)$$

where α , β , δ_h , k_α , k_β , and k_{δ_h} are wolves parameters, respectively.

To attain deep exploration, the searching behavior of a stochastic scout is

$$\vec{X}_{IGWO}(n+1) = \vec{X}_{IGWO}(n) + \vec{r}_{\delta_s} \quad (20)$$

where \vec{r}_{δ_s} represents a random reconnaissance, subject to controlled variables.

3.2. Adaptive differential evolution

Differential evolution has been employed to solve global optimization for many engineering fields [42]. After an adaptive operation is added, the ADE is powerful with few control parameters and contains variant populations, cross populations, and selection operations.

The initial population parameters are generated randomly by ADE as

$$\{x_i(0) | x_i(0) = [x_{i1}, x_{i2}, \dots, x_{iD}], x_{ij} = rb_j + (1 - r)a_j, j \in [1, D], i \in [1, N_p]\}, \quad (21)$$

where i is generation number, limits in $[0, 1]$; N_p represents the population size; r is a random number.

ADE generates variation in individuals by typical difference strategies. The strategy is shown by three steps, randomly selected three individuals, X_{ri} , $i = 1, 2, 3$.

The adaptive scaling factor $F_{i,g}$ is enhanced by the diversity of populations and guaranteed by the convergence of ADE to obtain optimal parameters. The adaptive scaling factor $F_{i,g}$ is updated as

$$F_{i,g} = \begin{cases} 1 - \frac{f_{\text{avg},g} - f'_g}{f_{\text{avg},g} - f_{\text{best},g}}, & \text{if } f_{\text{avg},g} > f'_g \\ \text{Cauchy}(-2, 0.5), & \text{otherwise} \end{cases}, \quad (22)$$

where random *Cauchy* ($-2, 0.5$) distribution includes the position parameter -2 and the scaling parameter 0.5 ; f'_g , $f_{\text{avg},g}$, and $f_{\text{best},g}$ are the g th individual, average, and global fitness values, respectively.

Location $X_i(g)$ is updated as

$$u_{ij} = \begin{cases} X_{ij}, & \text{if } r \geq P_C \text{ and } j \neq j_{\text{rand}} \\ V_{ij}, & \text{otherwise} \end{cases} \quad (23)$$

where j represents a random integer, ranging from 1 to d ; d represents the solution dimension; $P_{Ci,g+1}$ is the crossover probability of selecting a new individual for the mutant. To obtain the optimal value, the $P_{Ci,g+1}$ is

$$P_{Ci,g+1} = \begin{cases} P_{Ci,g}, & \text{if } f_{\text{avg},g} > f'_g \\ N(0.5, 0.5), & \text{otherwise} \end{cases}, \quad (24)$$

where $N(0.5, 0.5)$ is the Gaussian distribution; double 0.5 means that the variance and random values are the same.

A greedy strategy selects the individuals of ADE to obtain optimal next generation, as

$$X_{\text{nexti}} = \begin{cases} X_i, & \text{iff } (u_i) \geq f(X_i) \\ u_i, & \text{otherwise} \end{cases} \quad (25)$$

3.3. Deep fully connected models

DFCMs can learn the relationship between the inputs x and the outputs $f(x)$ of the fitness functions of optimization problems. Each fully connected layer of the DFCM represents outputs $f_{\text{pred}}(x)$ as [43]

$$f_{\text{pred}}(x) = \tanh(w_{\text{DFCM}}x + b_{\text{DFCM}}) \quad (26)$$

where w_{DFCM} and b_{DFCM} are DFCM weights and offsets, respectively; the prediction accuracy of the DFCM is evaluated by the root mean square error (RMSE) e_{RMSE} as [44]

$$e_{\text{RMSE}} = \sqrt{\frac{1}{n_{10}} \sum_{i=1}^{n_{10}} (f_{\text{pred},i}(x) - f_i(x))^2} \quad (27)$$

where n_{10} is the number values for evaluation; $f_{\text{pred},i}(x)$ and $f_i(x)$ are predict and real fitness values obtained by ten iterations of the GOE, respectively.

DFCMs replace the middle optimization parts of the optimization process of the GOE. DFCMs train the inputs and outputs of the first ten iterations of GOE. First, the well-trained DFCM predicts serval inputs with the same size of x . Then, the inputs with the corresponding smallest fitness value are selected as the initial wolf positions of the next iterations of the GOE. Finally, since the prediction accuracy of the well-trained DFCM may not be 100%, the last ten iterations are optimized by the conventional optimization method. As a result, the GOE optimizes only 20 iterations, and the total computation time of the proposed strategy is reduced by $\frac{t_m - 2n_{10}}{t_m} \times 100\%$; where t_m is the maximum iteration number of the GOE.

3.4. Steps of GOE accelerated by DFCMs for DFIG

Inspired by GWO and ADE, the hybridized GOE with the DFCM considers iteration speed and direction. The accelerated GOE is designed to perform optimum PI for the MPPT of DFIG-WT.

The updating steps of GOE accelerated by the DFCM for DFIG are listed below (Fig. 7).

Step 1. Initialize the parameters of DFIG, the weights of GOE, the parent population, mutant population, child population with random position, and the initialization parameters of DFCM.

(1.1) The parent population (X_p) is

$$X_p = \{X_1, X_2, \dots, X_k, \dots, X_m\}, \quad (28)$$

where k represents the serial time of individuals; m represents the population size; $k = 1, 2, 3, \dots, m$; X_k is $X_k = (X_{1,k}, X_{2,k}, \dots, X_{p,k}, \dots, X_{d,k})$, where p and d are individual serial numbers. The X is the PI parameter set of the four PI controllers of DFIG. That is to say, X is the parameter that need to be optimized by the GOE.

(1.2) The individual $X_{p,k}$ is

$$X_{p,k} = X_{p,k}(l) + (X_{p,k}(u) - rX_{p,k}(l)) \quad (29)$$

where $X_{p,k}(l)$ and $X_{p,k}(u)$ represent bottom and top boundaries, respectively; that is to say, $X_{p,k}(l)$ and $X_{p,k}(u)$ are PI parameters boundaries, respectively.

(1.3) The X_p classifies gray wolves as the leading groups, second groups, third individuals, and last groups, which are the α , β , δ and ω wolves, respectively.

Step 2. While ($t < n_{10}$), n_{10} is the running number of iterations before acceleration.

Step 3. Start Simulink simulation. Obtain system reactive power and control costs. Calculate the fitness function of each gray wolf by Eq. (12)

Step 4. Update parent population positions by Eqs. (13)–(25)

Step 5. Adaptively update $F_{i,g}$ and $P_{Ci,g+1}$

Step 6. Update mutant and child populations The best parent X_p value and the fitness function f are obtained. The gray wolves update their positions in the boundary constraints as

$$X_{ij} = \begin{cases} l_j, & \text{if } f(X_{ij}) < l_j \\ u_j, & \text{if } f(X_{ij}) > u_j, \quad j \in [1, d]; i \in [1, m], \\ X_{ij}, & \text{otherwise} \end{cases} \quad (30)$$

where l and u represent bottom and top boundaries, respectively.

Step 7. While $f(X_i) < f(X_p)$

Step 8. Convert child X_i to parent X_p

Step 9. End while

Step 10. Calculate ω wolf positions

Step 11. Calculate δ_2 wolf positions

Step 12. Output the optimal PI parameters of DFIG.

Step 13. End of the ($t < n_{10}$) iteration

Step 14. Input all the wolf positions (PI parameters) and fitness values (Eq. (12)) to the DFCM.

Step 15. Train the DFCM by inputs. Therefore, the relationship between the PI parameters and the objective function values containing the reactive power is learned by the DFCM.

Step 16. Input serval positions (PI parameters) to the well-trained DFCM; predict fitness values of these input positions.

Step 17. Select m positions with m smallest fitness values as initial positions of the next ten iterations of accelerated GOE

Step 18. While ($n_{10} < t \leq t_m - n_{10}$), t_m is the maximum number of iterations.

Step 19. Start Simulink simulation. Obtain system reactive power and control costs. Calculate the fitness function of each gray wolf by Eq. (12)

Step 20. Update parent population positions by Eqs. (13)–(25)

Step 21. Adaptively update $F_{i,g}$ and $P_{Ci,g+1}$

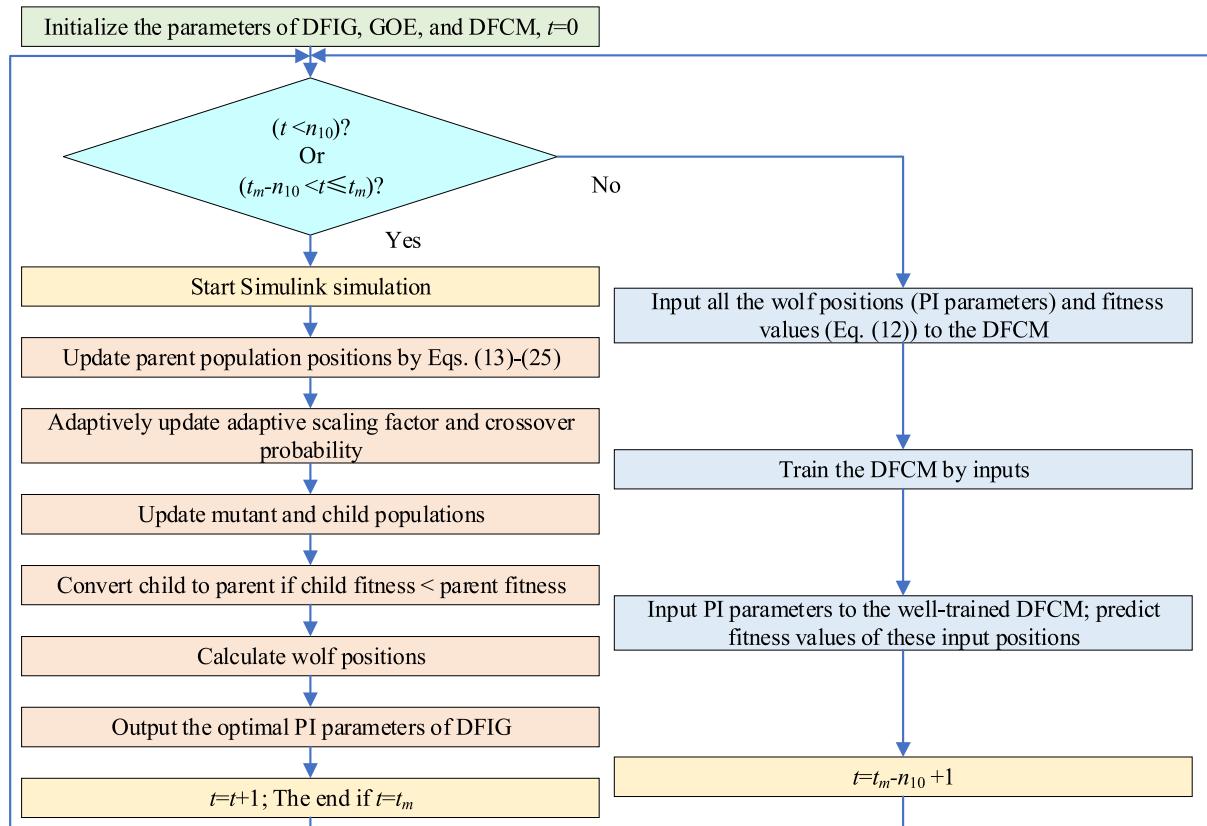


Fig. 7. Flow chart of the GOE accelerated by DFCMs.

- Step 22. Update mutant and child populations
- Step 23. While $f(X_i) < f(X_p)$
- Step 24. Convert child X_i to parent X_p
- Step 25. End while
- Step 26. Calculate ω wolves positions
- Step 27. Calculate δ_2 wolves positions
- Step 28. Output the optimal PI parameters of DFIG
- Step 29. End of the $(t_m - n_{10} < t \leq t_m)$ iteration
- Step 30. Return the best objective function value and the corresponding position (the best PI parameters). This GOE accelerated by the DFCM includes the following features.

- (1) The DFCM can reduce $(t_m - 2n_{10}) \times 100\% / t_m$ calculation time for complex optimization problems with long-time calculating fitness values. The accelerated GOE calculates only $2n_{10}$ iterations fitness values; the DFCM replaces other iterations; the calculation time of other iterations is far larger than the sum of the training time and predicting time used by the DFCM. The computation time of the whole optimization algorithm is reduced by reducing the number of iterations. Finding the optimal PI controller parameters for DFIG is accelerated as the computation time of the optimization algorithm is reduced.
- (2) The DFCM-based acceleration method can accelerate the optimization process because (a) the training process of DFCM can basically be computed by the graphics processing units; in contrast, the optimization process of conventional optimization algorithms basically needs the central processing units for calculation because the individual-to-individual coupling. For example, the update of the position of the gray wolf individuals requires the position of other better individuals in the previous and current iterations to be calculated (b) for a Simulink simulation with a long simulation time, the computation time to start and run

Simulink is much longer than the computation time of the optimization algorithm itself.

- (3) Compared with evolutionary computation, the ADE of the accelerated GOE reduces the complexity of evolutionary computation operations. Moreover, the ADE of the accelerated GOE has excellent performance, high reliability, efficiency, and robustness; the ADE of the accelerated GOE is an elegant parallel searching operation for unsolvable complex optimization problems and nonlinear non-solvable continuous problems.
- (4) The CGWO of the accelerated GOE selects the strongest wolf as the leader and then takes the prey under the leadership. As a result, the wolves are more effective at capturing prey. The leader wolf leads the wolves to capture the prey through continuous search, and the process is useful for optimizing the problem and finding the optimal global solution.

3.5. Convergence analysis of GOE accelerated by DFCMs

The general framework accelerated by DFCMs is an optimization algorithm framework. After several iterations, this algorithm can quickly search for optimization solutions. This ADE and this DFCM are the two key steps in the proposed algorithm.

First, the CGWO algorithm has convergence. After adding ADE, the convergence of this CGWO algorithm will be faster. Even if the ADE fails, the CGWO algorithm can find optimal solutions.

Secondly, the DFCM can learn the relationship between the independent variables and the objective function values. After learning, the DFCM can directly predict optimal independent variables. Although the optimal independent variables predicted by DFCM are not 100% accurate, DFCM can predict the suboptimal solutions near the optimal solutions. Therefore, this CGWO

Table 1
Benchmark functions.

Function	$[x_{\min}, x_{\max}]$	Minimum $F(x)$
$F_1(x) = \sum_{i=1}^{30} (x_i - d_i)^2$	$[-100, 100]$	0
$F_2(x) = \sum_{i=1}^{30} x_i - d_i + \prod_{i=1}^{30} x_i - d_i $	$[-10, 10]$	0
$F_3(x) = \sum_{i=1}^{30} \left(\sum_{j=1}^i (x_j - d_j) \right)^2$	$[-100, 100]$	0
$F_4(x) = \max \{ x_i - d_i \}, 1 \leq i \leq 30$	$[-100, 100]$	0
$F_5(x) = \sum_{i=1}^{29} \left[100 \left((x_{i+1} - d_{i+1}) - (x_i - d_i)^2 \right)^2 \right]$	$[-30, 30]$	0
$F_6(x) = \sum_{i=1}^{30} (x_i - d_i + 0.5)^2$	$[-100, 100]$	0
$F_7(x) = \sum_{i=1}^{30} i (x_i - d_i)^4$	$[-1.28, 1.28]$	0

algorithm can compensate for the non-100% accuracy problem after the DFCM prediction. This CGWO algorithm continues to find the optimal solution in the vicinity of the suboptimal solution after this DFCM prediction. After ten runs of the final CGWO algorithm, the proposed accelerated GOE accurately achieves the optimal solution. As a result, this DFCM only accelerates the iterative process of optimization approaches and does not modify the whole iterative framework.

However, the training time of this DFCM is much less than the time required for the iterative process of the optimization algorithm that many times. Therefore, the proposed algorithm speeds the controller parameter search process without changing the convergence of the overall controller parameter search process.

4. Experimental results

The GOE is realized in the RSC controller for benchmark functions and DFIG on the MATLAB/Simulink 7.10 with a computer with 2.9 GHz 16 GB RAM.

The parameters of these compared methods (i.e., genetic algorithm (GA), PSO, moth-flame optimization (MFO) [45], and GGWO [37]) are meticulously selected on a trial-and-error basis to attain optimal results. Besides, two winners of the CEC2020 Competition on Real-World Single Objective Constrained Optimization (i.e., symmetry adapted stochastic search (SASS) and Lévy flights-success-history based adaptive differential evolution with constraint handling technique (COLSHADE)) are compared with the GOE accelerated by DFCM. The details and default parameters of state-of-the-art methods can be found in Ref. [46]. All comparison algorithms, including these two award-winning optimization algorithms, are compared in two experiments of benchmark functions and DFIG for three scenarios. Since this study is the first to employ deep learning to accelerate optimization methods, no similar acceleration methods are available to compare with the acceleration algorithms proposed in this study.

4.1. Benchmark functions

Seven benchmark functions (Table 1) are applied in this case study to compare the optimization performances. The numbers of individuals and iterations of these comparison algorithms are set to 100. Note that, the random values d_i of these functions are the maximum x_{\max} times [0.8975, 0.0295, 1.4297, 0.7093, 1.5770, 1.9816, 0.2306, 0.2718, 0.6579, 1.6781, 0.7288, 0.8474, 1.0282, 0.4239, 1.9402, 1.4284, 1.4837, 0.3416, 1.7887, 1.5842, 0.9748, 0.0371, 0.8548, 1.7961, 1.7283, 0.2679, 1.0789, 0.4918, 1.8216, 0.2463]/100, which are not the same with Ref. [46]; the random values are not given in the Ref. [46]. The same random values are used when all compared algorithms are tested in the benchmark functions. The random values of the given benchmark functions in this work benefit future researchers to conduct fair comparisons. Since the upper limit of these independent variables is set to 100, a slight inattentiveness of the searched independent variables can

lead to a large fitness value. For example, the first searched value of function one is 20, which leads to a fitness value close to 400.

The optimized results, the computational time, and the convergence curves obtained by these algorithms (Fig. 8) show that: (i) the computation time required by the proposed algorithm is shorter than the time required by MFO, SASS, and COLSHADE methods; (ii) although taking more computational time, the SASS algorithm obtains the most optimal solution; (iii) although the code provided by the authors of the algorithms is employed, the time required by the COLSHADE algorithm is the largest among the compared algorithms; however, overall, the total time required by these algorithms is modest (iv) although the code provided by the authors of the algorithms is employed, the division of the optimal solution obtained by MFO is unstable; (v) the GOE accelerated by DFCM obtains acceptable fitness values with 39.99% lesser computation time than the SASS algorithm and 80.72% lesser computation time than the COLSHADE algorithm.

Note that: (1) if the maximum allowed functions evaluation is set as the maximum iteration number, the fitness values of the SASS method are very large; the maximum allowed functions evaluation of the SASS method in this case study is set as 1000×30 ; (2) if the population size of MFO is set 200, the fitness values are very large, the population size of the MFO in this case study is set to 10000.

4.2. Parameter optimization of doubly-fed induction generators

Three cases are considered: step increase change of WS (SIWS), sawtooth wave changes in WS (SWWS), and drop change of the power grid voltage (DRPG). The models described in Section 2 and applied in this study have been validated on the PSCAD/EMTDC software [47]. Fig. 9 shows gained results from these five algorithms over ten times running. Fig. 9 shows that GOE has the best convergence stability from different algorithms and balances development and exploration properly. The maximum iterations and population size are 100 and 200, respectively.

In Fig. 9, the fitness values of the accelerated GOE from 11 to 90 iterations are not given; the DFCM accelerates this process from 11 to 90 iterations. Fig. 9(a) shows that the accelerated GOE achieves smaller fitness values under ten times running. Fig. 9(a) red lines are the average fitness values obtained by these compared methods. The proposed accelerated optimization method obtains the minimized average fitness value (Fig. 9(a)).

The major parameters of the DFCM for accelerating the accelerated GOE are set as follows: gradient threshold method is L2 normalization; maximum epoch is 3000; learning rate is 0.001; training method is stochastic gradient descent method [39]; the unit numbers of the DFCM are [8 100 100 1] (Fig. 10); the training inputs are the wolf positions of previous ten iterations; the training outputs are fitness values of previous ten iterations.

The used time for training DFCM is 143.18 s. The training process of the deep model is given in Fig. 11.

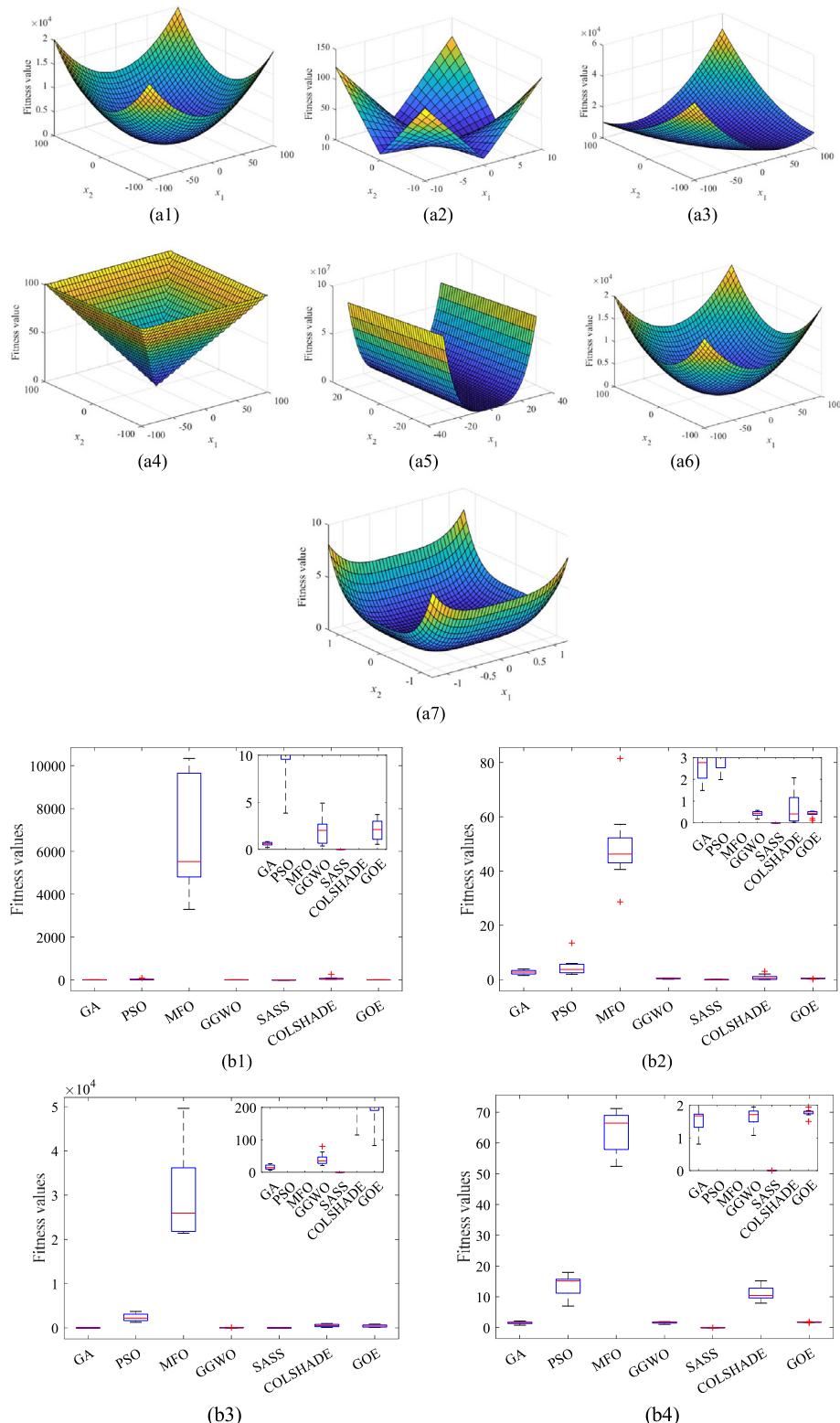
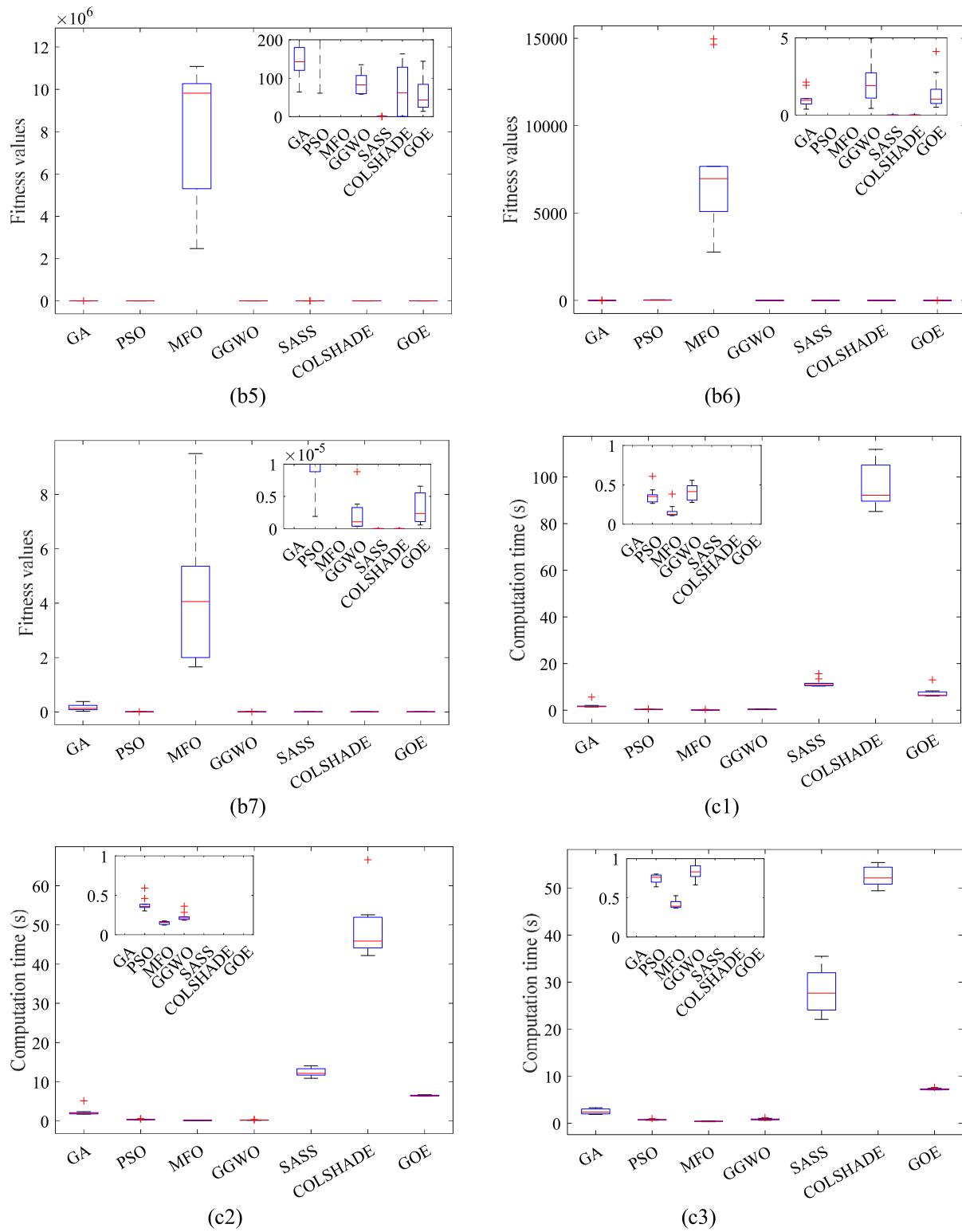


Fig. 8. Fitness values and computation time under benchmark functions: (a1)–(a7) fitness values; (b1)–(b7) obtained fitness values; (c1)–(c7) computation time; (d1)–(d7) best of convergence curves.

4.2.1. Wind speed of case 1

In this case, the value of WS is subdivided into three steps in the range of 8–10 m/s. A step wind contains all the frequency input information; in this case, three steps are considered for a complex case for the proposed accelerated method. Optimization results under the SIWS are shown in Fig. 12.

Fig. 12 shows that: (1) the accelerated GOE can modulate the reactive and active power more steadily and quickly compared with other algorithms, especially the SASS and COLSHADE algorithms; (2) since the accelerated GOE provides the maximum power factor, the best wind energy extraction can be realized.

**Fig. 8.** (continued).

4.2.2. Wind speed of case 2

In this case study, a sine input for a control system aims to test the controller responses for a single fixed frequency. The results under SWWS are shown in Fig. 13, which shows that: (i) the accelerated GOE withdraws the minimum control error; (ii) the accelerated GOE-optimized controller achieves better control performance under fixed frequency inputs.

4.2.3. Drop change of the voltage of power grid

The fault ride-through-based DIFG system is studied for wind generation into the power grid. The systemic voltage drops from normal value to 70% with 1 s are considered. Compared with all other algorithms, Fig. 14 shows that: (i) the accelerated GOE has the best fault ride-through capability with smoothest curves; (ii) the global search capability is the best; the accelerated GOE

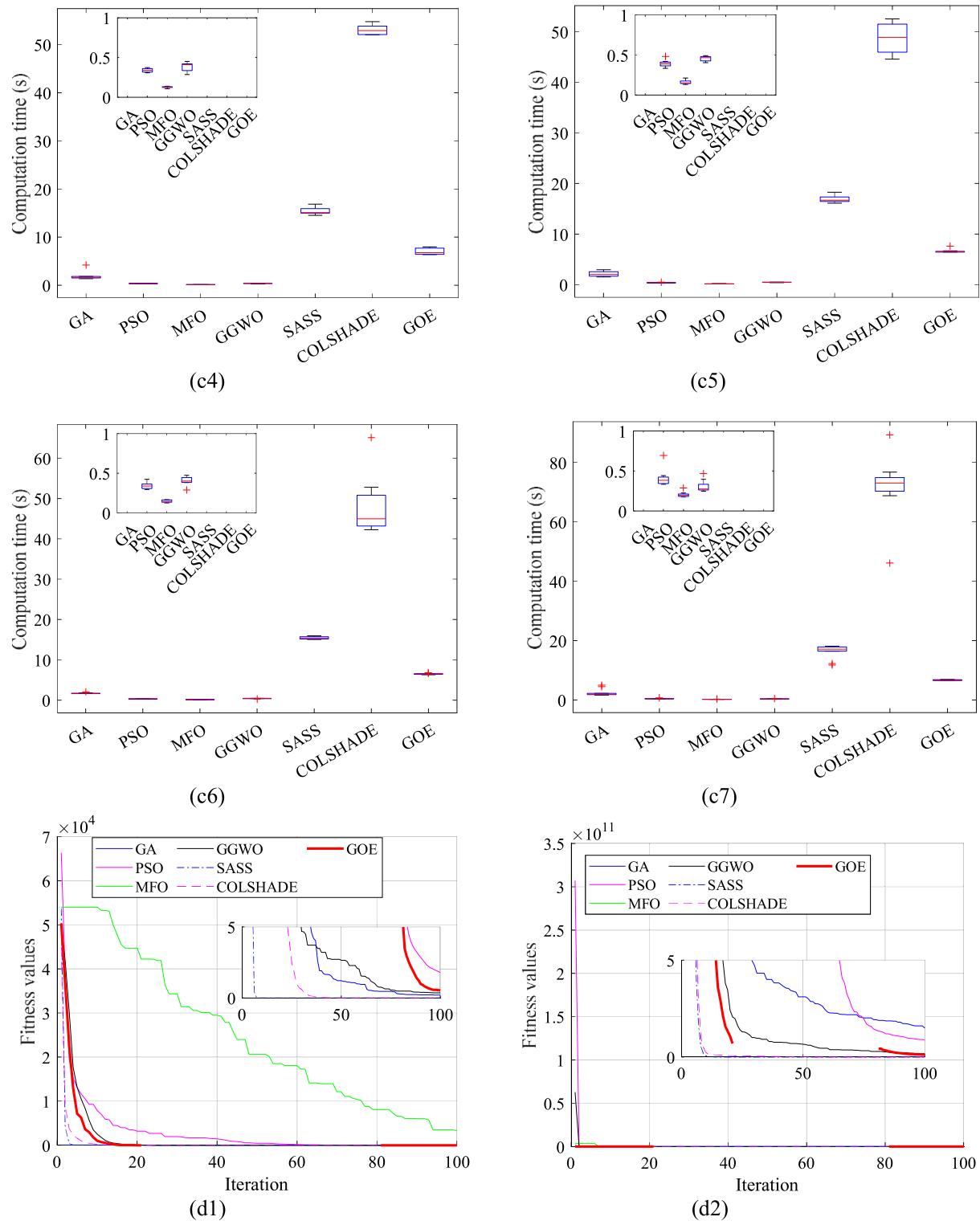


Fig. 8. (continued).

obtains better-coordinated controller parameters quickly; (iii) the feasibility and effectiveness of the accelerated GOE are verified under three complex cases in this work.

4.2.4. Summary of the three cases

After each method is run ten times under three complex cases, the optimized PI parameters are given in Table 2. Table 2 shows that numerous coordinated controller parameters could

obtain a similar control performance. The parameters optimization belongs to multimodal optimization problems. Therefore, although the accelerated GOE-optimized parameters are different from other optimized parameters, one solution to multimodal optimization problems is achieved by the GOE accelerated by the DFCM.

The simulation time is configured to the operating range as 50 s; the first 25 s is used as stability time. The calculation time of a one-time DFIG simulation is 6.57 s. Consequently, the calculation

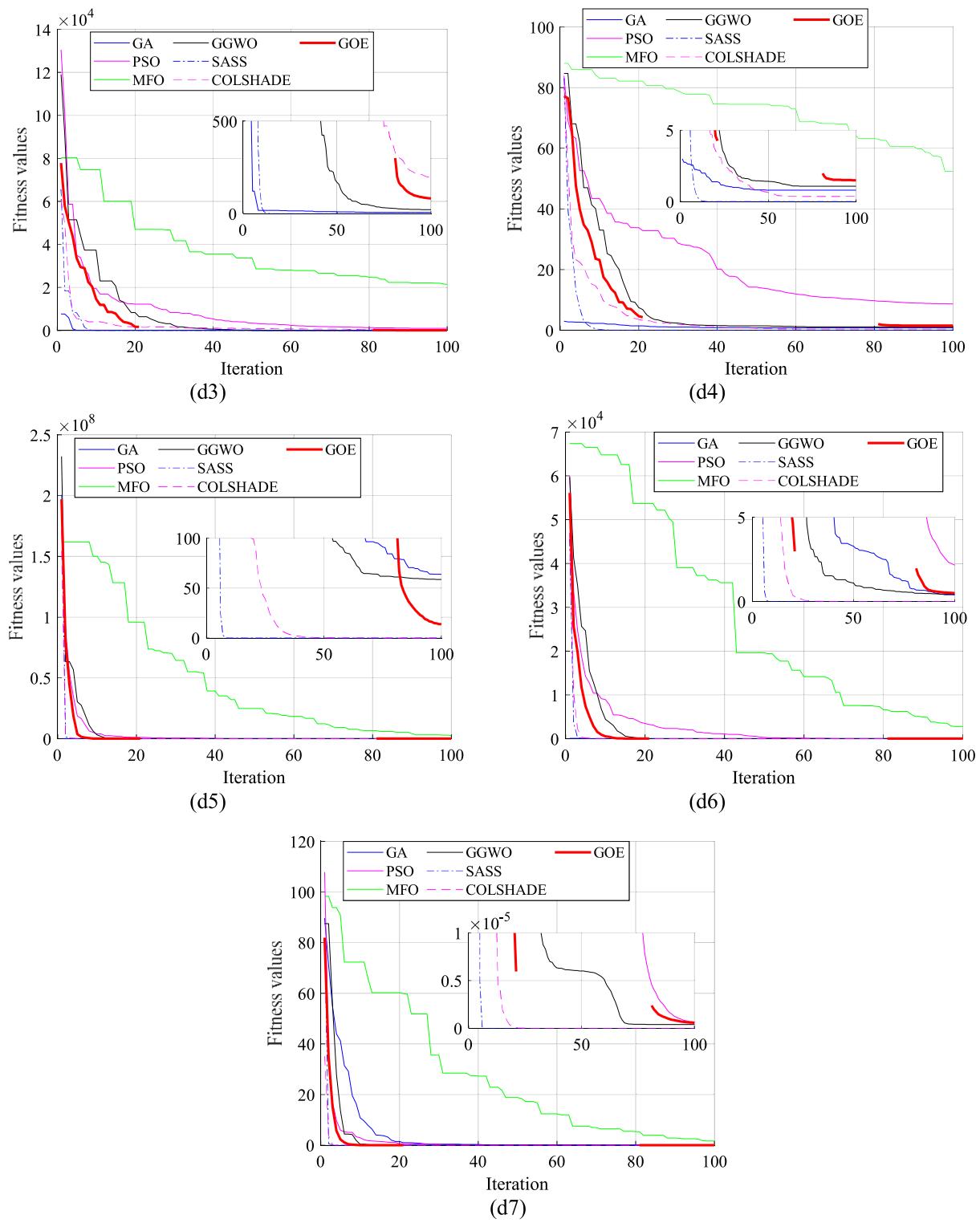


Fig. 8. (continued).

time of the fitness function with 100 iterations and 200 individuals is 131400 s or 1.52 days. Table 3 shows that the accelerated GOE with the DFCM obtains better solutions for complex practical engineering problems with 99.51% lesser average computation time than the SASS algorithm, 99.63% less than the COLSHADE algorithm, and 89.52% less than other methods.

This accelerated GOE algorithm reduces the execution time and obtains higher control performances. The proposed algorithm

compensates for both the slow optimization process and the problem of the non-100% accuracy of prediction algorithms.

Table 4 summarizes the absolute error of reactive power (IAE_Q) and rotor speed (IAE_{wr}) under three cases. The IAE_Q of GOE is the smallest in these three cases. The accelerated GOE with the DFCM has the highest comparative performance compared with other algorithms.

Table 5 clarifies all the control costs of the five algorithms. The most reasonable one is the MPPT system, on which the

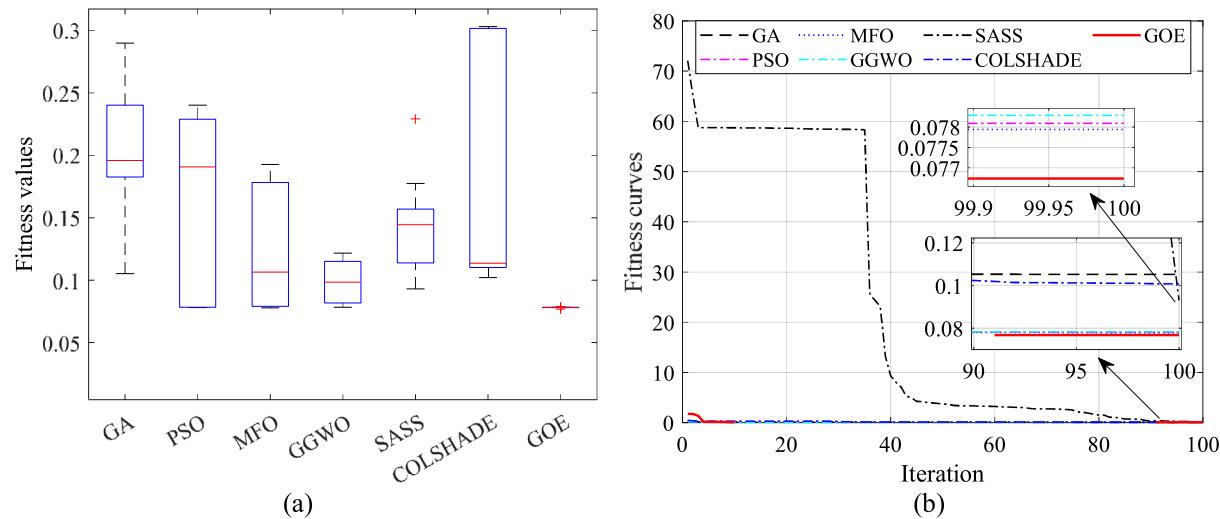


Fig. 9. Statistic results of five algorithms on three cases: (a) statistical fitness values; (b) fitness curves.

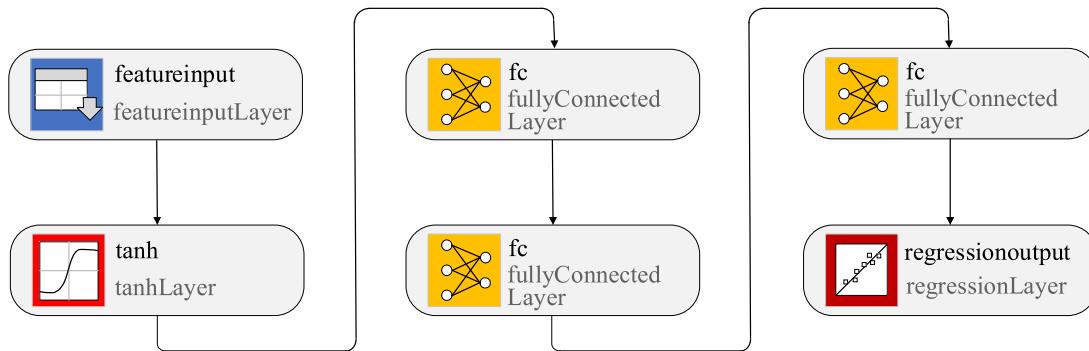


Fig. 10. Structure of the DFCM.

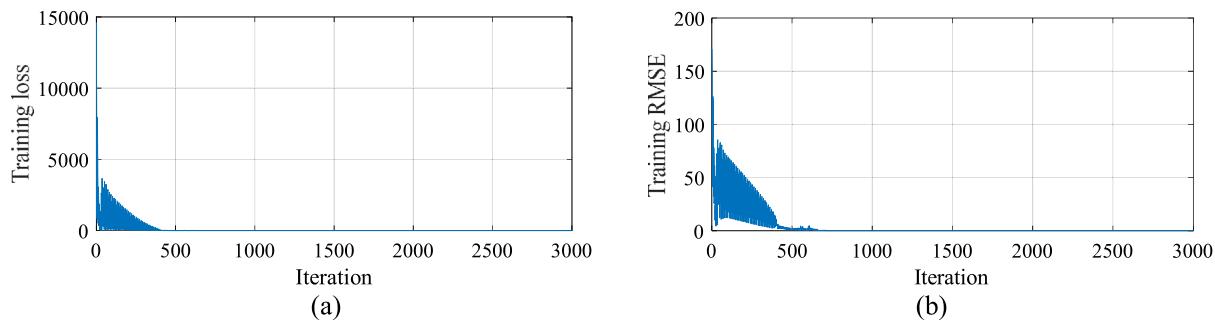


Fig. 11. Training process of DFCM: (a) training loss; (b) training RMSE.

Table 2
Best PI parameters gained by five algorithms.

Algorithm	$K_{p,wr}$	$K_{i,wr}$	$K_{p,iq}$	$K_{i,iq}$	$K_{p,Q}$	$K_{i,Q}$	$K_{i,id}$	$K_{p,id}$
GA	843.23	32.26	0.00680	0.00485	159.16	41.44	0.0189	0.00609
PSO	803.27	37.88	0.00781	0.00415	143.16	37.85	0.0271	0.00388
MFO	741.63	37.73	0.00763	0.00392	179.18	40.38	0.0259	0.00337
GGWO	782.26	36.88	0.00457	0.00305	210.36	39.45	0.0353	0.00309
SASS	744.67	39.99	0.00983	0.00150	117.19	34.92	0.0231	0.00259
COLSHADE	600.67	35.99	0.00603	0.00360	120.19	34.94	0.0251	0.00356
GOE	683.66	35.99	0.00803	0.00363	120.19	34.92	0.0251	0.00356

absolute reactive power costs and control costs are traded off. The accelerated GOE achieves a global optimum in most cases than other algorithms.

After ten runs of each method, Table 6 provides the worst, best, and mean fitness values, the skewness, kurtosis, smallest standard, and relative deviations, which show that: (1) the

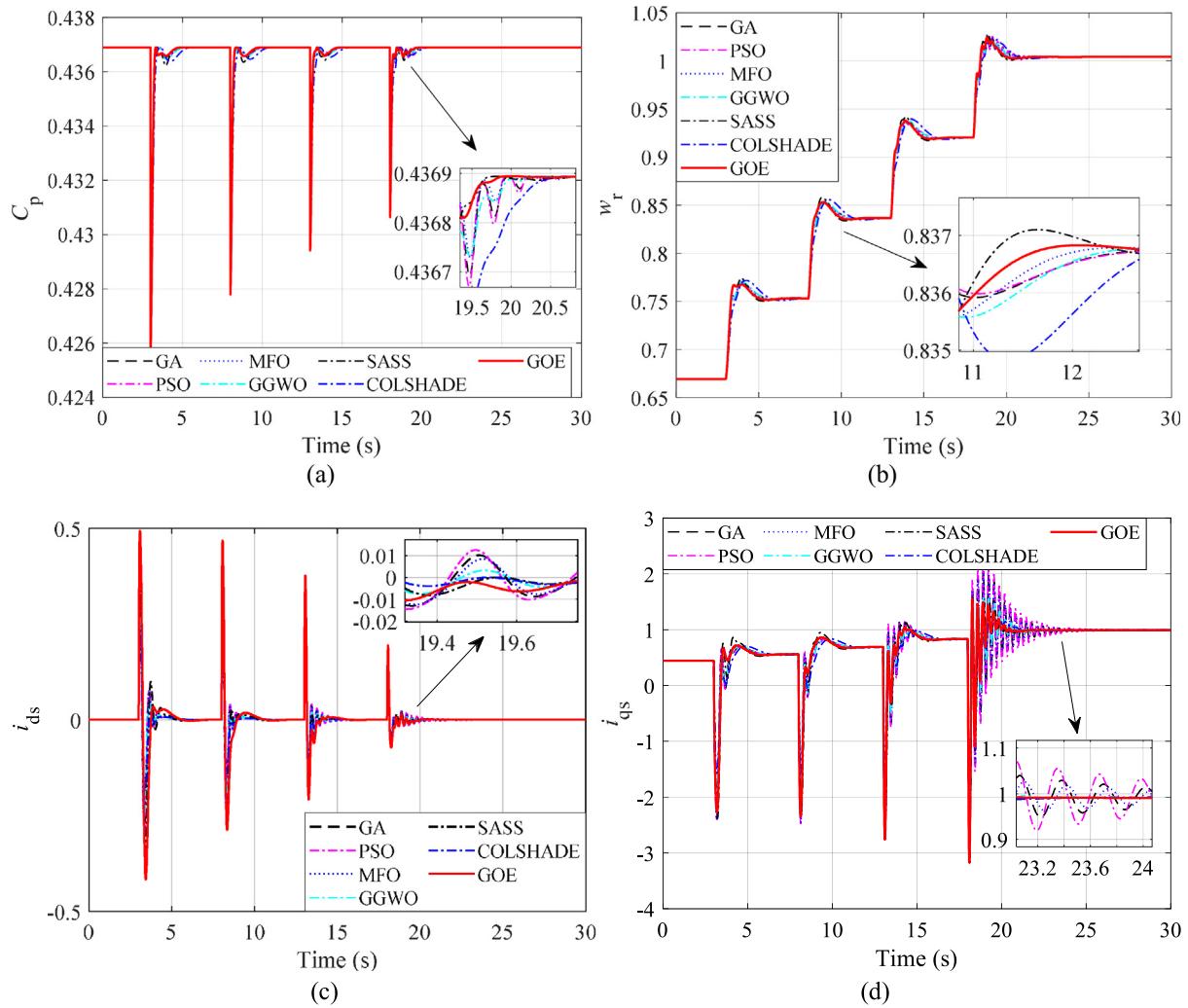


Fig. 12. System responses under SIWS: (a) power coefficient; (b) rotor speed; (c) d -axis current; (d) q -axis current.

Table 3
Total execution time in 10 runs.

Algorithm	Maximum execution time (s)	Minimum execution time (s)	Mean execution time (s)
GA	132580.79	131564.78	131689.76
PSO	132524.38	131563.57	131597.38
MFO	132879.56	131704.20	131897.56
GGWO	138575.23	132055.84	132656.62
SASS	951156.85	2524159.56	2798372.01
COLSHADE	4661213.51	3285188.06	3721459.65
GOE with the DFCM	16393.12	13394.93	13787.64

Table 4
Statistical results of IAE_Q and IAE_{wr} gained by five algorithms in three cases.

Algorithm	SIWS		SWWS		DRPG	
	IAE_Q	IAE_{wr}	IAE_Q	IAE_{wr}	IAE_Q	IAE_{wr}
GA	0.3043	0.1333	0.0420	0.0512	0.1144	0.00847
PSO	0.3146	0.1324	0.0444	0.0510	0.1157	0.00845
MFO	0.3071	0.1314	0.0411	0.0455	0.1144	0.00862
GGWO	0.2486	0.1396	0.0343	0.0520	0.0977	0.00825
SASS	0.3869	0.1402	0.0459	0.0372	0.1281	0.00876
COLSHADE	0.5272	0.1803	0.0708	0.0744	0.1108	0.00839
GOE	0.2155	0.1217	0.0323	0.0374	0.0929	0.00708

proposed accelerated GOE is more advantageous than the traditional solver algorithms; (2) the accelerated GOE obtains better solutions steadily and lower control costs with lesser computation time; (3) deviations obtained by the accelerated GOE mean

that the accelerated GOE provides the high convergence stability and reliability than others; (4) the local searching ability of GA and PSO solvers may not be good from the high standard

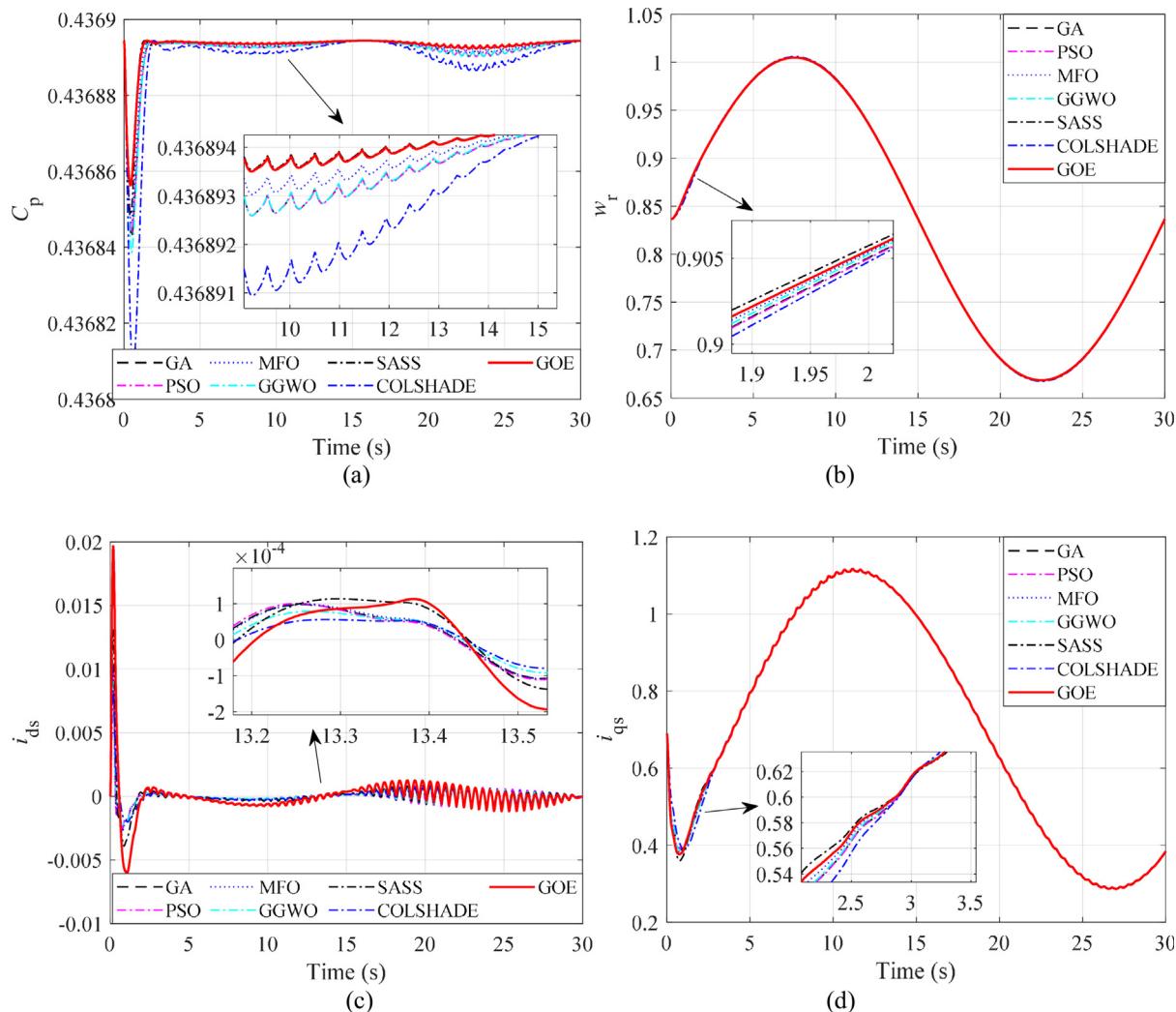


Fig. 13. System responses under SWWS: (a) power coefficient; (b) rotor speed; (c) d-axis current; (d) q-axis current.

Table 5
Control costs after ten runs of each method.

Algorithm	SIWS control cost (p.u.)		SWWS control cost (p.u.)		DRPG control cost (p.u.)	
	Power cost	Control cost	Power cost	Control cost	Power cost	Control cost
GA	0.10936	3.56875	0.25437	0.06012	0.29962	0.01635
PSO	0.10935	3.56873	0.25437	0.06014	0.29962	0.01650
MFO	0.10935	3.56872	0.25439	0.05967	0.29962	0.01640
GGWO	0.10942	3.56885	0.25436	0.06018	0.29963	0.01525
SASS	0.10938	3.56844	0.25443	0.05898	0.29965	0.01403
COLSHADE	0.10966	3.56917	0.25428	0.06218	0.29966	0.01315
GOE	0.10932	3.56780	0.25443	0.05912	0.29962	0.01048

Table 6
Fitness values under three cases after ten tests of each method.

Method	Worst	Best	Mean	Standard deviation	Skewness	Kurtosis	Relative standard deviation
GA	0.2900	0.1053	0.2095	0.0540	-0.2263	2.6870	0.2577
PSO	0.2403	0.0781	0.1657	0.0721	-0.2801	1.2836	0.4352
MFO	0.1928	0.0779	0.1208	0.0480	0.6434	1.7164	0.3969
GGWO	0.1216	0.0783	0.0984	0.0191	0.0198	1.0670	0.1943
SASS	0.2292	0.0931	0.1448	0.0400	0.6741	3.1158	0.2764
COLSHADE	0.3031	0.1019	0.1694	0.0919	0.8554	1.7558	0.5427
GOE	0.0788	0.0767	0.0781	0.0005	-1.7017	5.8978	0.0067

deviation of the recorded results among continuous solution candidates; (5) the accelerated GOE avoids the local search optimum more effectively than other algorithms.

In addition, the Friedman's and Kruskal-Wallis tests are 0.6760 and 1, respectively. The box plot of the Kruskal-Wallis test of found variables by these compared methods is given in Fig. 15.

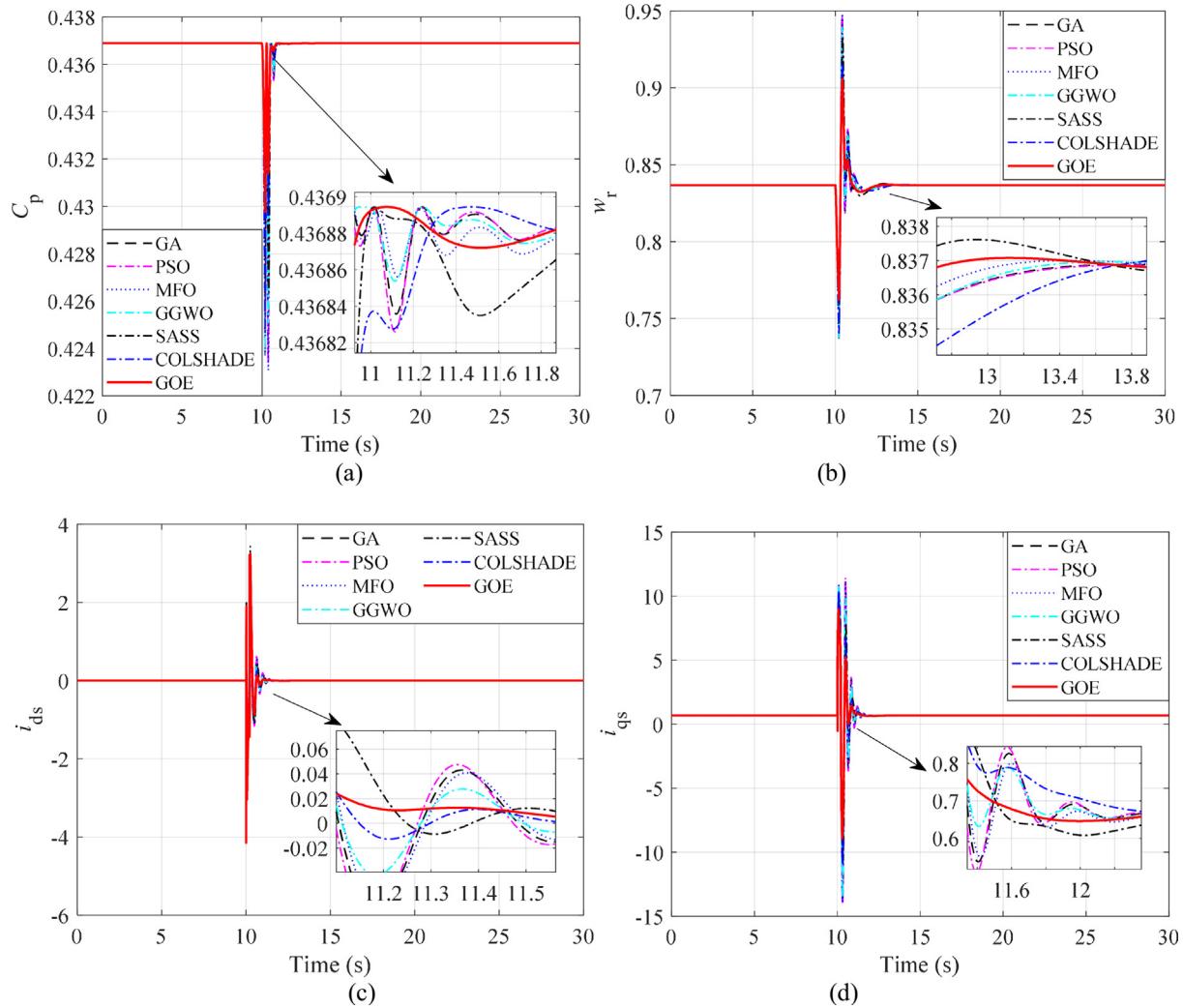


Fig. 14. System responses under DRPG: (a) power coefficient; (b) rotor speed; (c) d -axis current; (d) q -axis current.

From the convergence of the iteration curves, setting a number of iterations greater than or equal to 100 is sufficient for the optimization problem of the benchmark functions and DFIG experiments in this study. In general, the number of populations should be greater than twice the number of variables to be optimized; therefore, the number of populations of the optimization algorithms applied to the benchmark functions in this study should be greater than 60; and the number of populations of the optimization algorithms applied to the DFIG experiments in this study should be greater than 16.

In general, a relatively small number of training samples requires a larger number of maximum epochs to obtain a relatively satisfactory prediction accuracy of DFCM. When the learning rate is less than 0.01, a maximum epoch training process greater than 100 can compensate for the randomness of the training process. Different from convolutional neural networks, the training accuracy of DFCM is less dependent on the training function; therefore, all three commonly available training functions can be chosen. The number of neurons in DFCM should be greater than twice the number of inputs.

The longer the iterative process of the optimization algorithm is replaced by the proposed acceleration method of this study, the longer the time saved by the acceleration algorithm. However, a too short pre-processing of the optimization algorithm might lead to too homogeneous DFCM training samples, which in turn provokes a decrease in the accuracy of the DFCM. Therefore, after

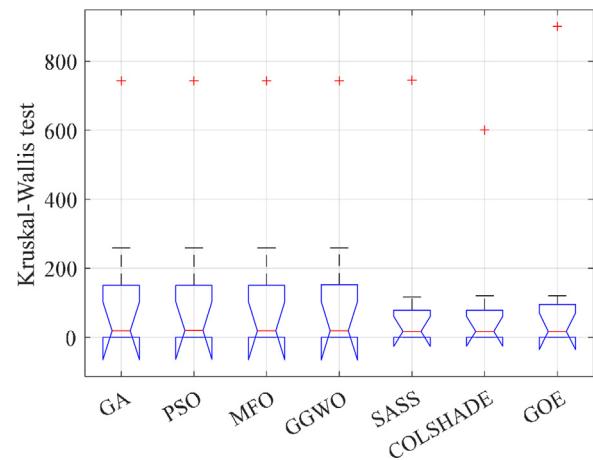


Fig. 15. Box plot of Kruskal-Wallis test of found variables by compared methods.

extensive experiments, the optimization algorithm has to provide training samples with at least five iterations to the proposed DFCM for training.

After the GOE accelerated by the DFCM is verified under three complex cases, the accelerated GOE has the following major limitations.

- (1) Three scenarios may lead to an ineffective acceleration of the proposed acceleration algorithm: (a) the time for solving the optimization process is short; functions with short optimization time are recommended without the proposed method for acceleration; (b) the training time of the resulting algorithm is long; when the training time of the DFCM is excessive because of the large number of variables to be optimized, the proposed DFCM can be replaced by convolutional neural networks to improve the prediction accuracy and convergence speed; (c) dirty data may lead to an explosion of the training gradient of the DFCM; the increase of the randomness of the optimization algorithm can reduce the dirty data provided for the training of the DFCM.
- (2) Although a fast and better optimization result is obtained in this study, the optimization results are obtained in the simulation model. As with all comparative optimization algorithms, the optimization search process for controller parameter optimization cannot be computed in a real DFIG because (a) the search process may find non-converging parameters and cause damage to the DFIG, and (b) the environment of a real-life DFIG cannot set the input for the three scenarios that change in real-time during the search process. However, the controller parameters found by the optimization algorithm can be applied to the real-life DFIG.
- (3) The penalty function of accelerated GOE is not effective when large infeasible regions exist in the search space.
- (4) The accelerated GOE has not been used for the general binary model. However, the accelerated GOE with the DFCM can adopt integral transfer functions or other operators to solve various binary problems.
- (5) The ADE of the accelerated GOE decreases the population diversity as the evolutionary generations number increases during iterative processes, converges to local minima prematurely, or causes the algorithm to stagnate, which makes the performance worse during the evolutionary process. Fortunately, the disadvantage of the ADE can be mitigated by the CGWO of the accelerated GOE.

5. Conclusions

A GOE with the DFCM is developed for DFIG-WTs with MPPT under various test conditions. According to case studies, the GOE with the DFCM can increase wind energy of effectively restore reliability levels with robust performance. Meanwhile, the control cost of GOE with the DFCM is smaller than that of other algorithms. Furthermore, the computation time is the least compared with other compared methods. The GOE with the DFCM includes the following advantages.

- (1) The GOE with the DFCM has four types of wolves for cooperative hunting, i.e., α , β , δ , and ω wolves, which aim to search the prey for working with the trade-off percent, has deeper exploitation ability with the ADE of GOE with the DFCM.
- (2) In an unknown environment, the GOE with the DFCM has a random scout team, including ω and δ wolves to search for possible prey randomly and a deeper exploitation ability with ADE compared with GA, PSO, and MFO. From the experiment results, the probability of gaining high-quality optimal results of GOE with the DFCM is the best; the fitness value of GOE with the DFCM is smaller than other approaches; the calculated time used by the GOE with the DFCM is less than other methods.

- (3) The GOE accelerated by the DFCM with global search capability rapidly solves non-smooth problems with low costs. Furthermore, the DFCM algorithm, which is not 100% accurate, provides a powerful alternative to the DFCM algorithm to accelerate the optimization process of the GOE optimization algorithm, which continues the optimization search based on the results predicted by the DFCM algorithm to search for the optimal solution more quickly.

In the future, the proposed accelerated GOE with the DFCM could be implemented by a field-programmable gate array in practical DFIG controllers; a multi-objective accelerated GOE could be designed for more complex controller parameters optimization problems. Furthermore, multi-objective accelerated GOE could be designed and implemented for more complex practical engineering problems, such as squirrel cage induction generators and direct-drive wind turbines. In addition, dynamic optimization problems could be considered by an improved GOE with the DFCM with dynamic tolerance and levy flights. Additionally, a faster training method could be designed to accelerate the training process of DFCM itself. Besides, the accelerated method could be lighted for fast optimization problems.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments

This work was supported by the Fujian Provincial Key Laboratory of Intelligent Identification and Control of Complex Dynamic System, China under Grant. 2022A0002, the Supporting project of STS program of Fujian Academy of Sciences under Grant. 2021T3022, the National Natural Science Foundation of China under Grant. 52107081 and the National Natural Science Foundation of Guangxi Province under Grant. AA22068071.

References

- [1] D. Fózér, M. Volanti, F. Passarini, P.S. Varbanov, J.J. Kleme, P. Mizsey, Bioenergy with carbon emissions capture and utilisation towards GHG neutrality: power-to-gas storage via hydrothermal gasification, *Appl. Energy* 280 (2020) 115923.
- [2] H. Wei, H. Zhang, D. Yu, Y. Wang, D. Ling, X. Ming, Short-term optimal operation of hydro-wind-solar hybrid system with improved generative adversarial networks, *Appl. Energy* 250 (2019) 389–403.
- [3] A. Khosravi, V. Olkkonen, A. Farsaei, S. Syri, Replacing hard coal with wind and nuclear power in Finland—impacts on electricity and district heating markets, *Energy* 203 (2020) 117884.
- [4] A.M. Omer, On the wind energy resources of Sudan, *Renew. Sustain. Energy Rev.* 12 (8) (2008) 2117–2139.
- [5] Z. Song, S. Feng, L. Zhang, Z. Hu, X. Hu, R. Yao, Economy analysis of second-life battery in wind power systems considering battery degradation in dynamic processes: real case scenarios, *Appl. Energy* 251 (2019) 113411.
- [6] D.A. Tejada-Arango, G. Morales-Espaa, S. Wogrin, E. Centeno, Power-based generation expansion planning for flexibility requirements, *IEEE Trans. Power Syst.* 35 (3) (2019) 2012–2023.
- [7] Y. Yang, J. Zhou, G. Liu, L. Mo, F. He, Multi-plan formulation of hydropower generation considering uncertainty of wind power, *Appl. Energy* 260 (2020) 114239.
- [8] S. Ebrahimkhani, Robust fractional order sliding mode control of doubly-fed induction generator (DFIG)-based wind turbines, *ISA Trans.* 63 (2016) 343–354.

- [9] G. Bi, G. Wang, G. Zhang, N. Zhao, D.G. Xu, Low-noise initial position detection method for Sensorless permanent magnet synchronous motor drives, *IEEE Trans. Power Electron.* 35 (12) (2020) 13333–13344.
- [10] F. Shi, D. Shu, Z. Yan, Z. Song, A shifted frequency impedance model of doubly-fed induction generator (DFIG)-based wind farms and its applications on S 2 SI analysis, *IEEE Trans. Power Electron.* 36 (1) (2020) 215–227.
- [11] C.R. Raghavendran, J.P. Roselyn, U.D. Sowmmiya, Devaraj, Effective power transfer and reduced-order generalized integrator sequence based fault ride through strategy in grid connected DFIG based WECS, *Int. J. Electr. Power Energy Syst.* 130 (4) (2021) 106809.
- [12] K. Shi, X. Yin, L. Jiang, Y. Liu, Y. Hu, H. Wen, Perturbation estimation based nonlinear adaptive power decoupling control for DFIG wind turbine, *IEEE Trans. Power Electron.* 35 (1) (2020) 319–333.
- [13] H.T. Do, T.D. Dang, H.V.A. Truong, K.K. Ahn, Maximum power point tracking and output power control on pressure coupling wind energy conversion system, *IEEE Trans. Ind. Electron.* 65 (2) (2018) 1316–1324.
- [14] M. Oussama, C. Abdelghani, C. Lakhdar, Efficiency and robustness of type-2 fractional fuzzy PID design using salps swarm algorithm for a wind turbine control under uncertainty, *ISA Trans.* 125 (2022) 72–84.
- [15] H. Cherif, A. Benakcha, I. Laib, S.E. Chehaidia, A. Menacer, B. Soudan, A.G. Olabi, Early detection and localization of stator inter-turn faults based on discrete wavelet energy ratio and neural networks in induction motor, *Energy* 212 (2020) 118684.
- [16] B. Yang, L. Jiang, L. Wang, W. Yao, Q.H. Wu, Nonlinear maximum power point tracking control and modal analysis of DFIG based wind turbine, *Int. J. Electr. Power Energy Syst.* 74 (2016) 429–436.
- [17] X. Zheng, Y. Feng, F. Han, X. Yu, Integral-type terminal sliding-mode control for grid-side converter in wind energy conversion systems, *IEEE Trans. Ind. Electron.* 66 (5) (2019) 3702–3711.
- [18] A. Hussain, M.M. Garg, M.P. Korukonda, S. Hasan, L. Behera, A parameter estimation based MPPT method for a PV system using Lyapunov control scheme, *IEEE Trans. Sustain. Energy* 10 (4) (2019) 2123–2132.
- [19] C. Kim, Y. Gui, C.C. Chung, Maximum power point tracking of a wind power plant with predictive gradient ascent method, *IEEE Trans. Sustain. Energy* 8 (2) (2017) 685–694.
- [20] J.P. Ram, N. Rajasekar, M. Miyatake, Design and overview of maximum power point tracking techniques in wind and solar photovoltaic systems: A review, *Renew. Sustain. Energy Rev.* 73 (2017) 1138–1159.
- [21] L. Yin, L. Chen, D. Liu, X. Huang, F. Gao, Quantum deep reinforcement learning for rotor side converter control of double-fed induction generator-based wind turbines, *Eng. Appl. Artif. Intell.* 106 (2021) 104451.
- [22] C. Du, Z. Yin, J. Liu, Y. Zhang, X. Sun, A speed estimation method for induction motors based on active disturbance rejection observer, *IEEE Trans. Power Electron.* 35 (8) (2020) 8439–8442.
- [23] D.C. Das, N. Sinha, A.K. Roy, Small signal stability analysis of dish-Stirling solar thermal based autonomous hybrid energy system, *Int. J. Electr. Power Energy Syst.* 63 (2014) 485–498.
- [24] H. Feng, W. Ma, C. Yin, D. Cao, Trajectory control of electro-hydraulic position servo system using improved PSO-PID controller, *Autom. Constr.* 127 (7) (2021) 103722.
- [25] A. Jpm, B. Jvfn, C. Efmf, B. Emaf, On the design and analysis of structured-ANN for online PID-tuning to bulk resumption process in ore mining system, *Neurocomputing* 402 (2020) 266–282.
- [26] X. Wang, P. Henshaw, D. Ting, J. Yan, Exergoeconomic analysis for a thermoelectric generator using mutation particle swarm optimization (M-PSO), *Appl. Energy* 294 (2021) 116952.
- [27] K. Han, T. Huang, L. Yin, Quantum parallel multi-layer Monte Carlo optimization algorithm for controller parameters optimization of doubly-fed induction generator-based wind turbines, *Appl. Soft Comput.* 112 (2021) 107813.
- [28] L. Yin, Q. Gao, Multi-objective proportional-integral-derivative optimization algorithm for parameters optimization of double-fed induction generator-based wind turbines, *Appl. Soft Comput.* 110 (2021) 107673.
- [29] B. Yang, X. Zhang, T. Yu, H. Shu, Z. Fang, Grouped grey wolf optimizer for maximum power point tracking of doubly-fed induction generator based wind turbine, *Energy Convers. Manage.* 133 (2017) 427–443.
- [30] A. Fathy, H. Rezk, D. Yousri, A robust global MPPT to mitigate partial shading of triple-junction solar cell-based system using manta ray foraging optimization algorithm, *Sol. Energy* 207 (2020) 305–316.
- [31] Z. Sheng, H. Wang, G. Chen, B. Zhou, J. Sun, Convolutional residual network to short-term load forecasting, *Appl. Intell.* 51 (4) (2021) 2485–2499.
- [32] H. Huang, R. Jia, X. Shi, J. Liang, J. Dang, Feature selection and hyper parameters optimization for short-term wind power forecast, *Appl. Intell.* 51 (10) (2021) 6752–6770.
- [33] X. Meng, J. Jiang, H. Wang, AGWO: advanced GWO in multi-layer perception optimization, *Expert Syst. Appl.* 173 (2021) 114676.
- [34] M.A. Al-Betar, M.A. Awadallah, M.M. Krishan, A non-convex economic load dispatch problem with valve loading effect using a hybrid grey wolf optimizer, *Neural Comput. Appl.* 32 (2020) 12127–12154.
- [35] J. Gurrola-Ramos, A. Hernández-Aguirre, O. Dalmau-Cedeño, COLSHADE for real-world single-objective constrained optimization problems, in: 2020 IEEE Congress on Evolutionary Computation, (CEC), IEEE, Glasgow, UK, 2020, pp. 1–8, <http://dx.doi.org/10.1109/CEC48606.2020.9185583>, 19–24.
- [36] Z.G. Chen, Z.H. Zhan, H. Wang, J. Zhang, Distributed individuals for multiple peaks: a novel differential evolution for multimodal optimization problems, *IEEE Trans. Evol. Comput.* 24 (4) (2020) 708–719.
- [37] H. Zhao, Z.H. Zhan, Y. Lin, X. Chen, X.N. Luo, J. Zhang, et al., Local binary pattern-based adaptive differential evolution for multimodal optimization problems, *IEEE Trans. Cybern.* 50 (7) (2020) 3343–3357.
- [38] J. Hu, H. Zhang, Y. Liu, R. Sutcliffe, J. Feng, BBW: a batch balance wrapper for training deep neural networks on extremely imbalanced datasets with few minority samples, *Appl. Intell.* 52 (6) (2022) 6723–6738.
- [39] L. Yin, Q. Gao, Proportional-integral-derivative optimization algorithm for double-fed induction generator with the maximum wind power tracking technique, *Soft Comput.* 25 (4) (2021) 3097–3111.
- [40] B. Babaghorbani, M.T. Beheshti, H.A. Talebi, A Lyapunov-based model predictive control strategy in a permanent magnet synchronous generator wind turbine, *Int. J. Electr. Power Energy Syst.* 130 (2021) 106972.
- [41] S. Ebrahimkhani, Robust fractional order sliding mode control of doubly-fed induction generator (DFIG)-based wind turbines, *ISA Trans.* 63 (2016) 343–354.
- [42] Y.Q. Cai, D.W. Wu, Y. Zhou, S.K. Fu, H. Tian, Y.Q. Du, Self-organizing neighborhood-based differential evolution for global optimization, *Swarm Evol. Comput.* 56 (2020) 100699.
- [43] T. Tuna, A. Beke, T. Kumbasar, Deep learning frameworks to learn prediction and simulation focused control system models, *Appl. Intell.* 52 (1) (2022) 662–679.
- [44] Q. Wang, M. Zhang, Y. Zhang, J. Zhong, V.S. Sheng, Location-based deep factorization machine model for service recommendation, *Appl. Intell.* 52 (2022) 9899–9918.
- [45] X.J. Lei, M. Fang, H. Fujita, Moth-flame optimization-based algorithm with synthetic dynamic PPI networks for discovering protein complexes, *Knowl.-Based Syst.* 172 (2019) 76–85.
- [46] J. Xu, L. Xu, Optimal stochastic process optimizer: A new metaheuristic algorithm with adaptive exploration-exploitation property, *IEEE Access* 9 (2021) 108640–108664.
- [47] W. Qiao, Dynamic modeling and control of doubly fed induction generators driven by wind turbines, 15–18 2009, in: 2009 IEEE/PES Power Systems Conference and Exposition, IEEE, Seattle, WA, USA, 2009, pp. 1–8.