Applied Soft Computing Journal 86 (2020) 105937

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

Enhanced whale optimization algorithm for maximum power point tracking of variable-speed wind generators

Mohammed H. Qais [a,*], Hany M. Hasanien [b], Saad Alghuwainem [a]

[a] Electrical Engineering Department, Faculty of Engineering, King Saud University, Riyadh 11421, Saudi Arabia
[b] Electrical Power and Machines Department, Faculty of Engineering, Ain Shams University, Cairo 11517, Egypt

ARTICLE INFO

Article history:
Received 22 March 2019
Received in revised form 22 September 2019
Accepted 7 November 2019
Available online 15 November 2019

Keywords:
Benchmark functions
Optimization methods
Whale optimization algorithm
Wind generator

ABSTRACT

This paper proposes an enhancement of the meta-heuristic whale optimization algorithm (WOA) for maximum power point tracking (MPPT) of variable-speed wind generators. First of all, twenty-three benchmark functions tested the enhanced whale optimization algorithm (EWOA). Then the statistical results of EWOA compared with the results of other algorithms (WOA, salp swarm algorithm (SSA), enhanced SSA (ESSA), grey wolf optimizer (GWO), augmented GWO (AGWO), and particle swarm optimization (PSO). Also, the non-parametric statistical test and convergence curves proved the superiority and the speed of the EWOA. After that, the EWOA and WOA are implemented to design optimal Takagi–Sugeno fuzzy logic controllers (FLCs) to enhance the MPPT control of variable-speed wind generators. Moreover, real wind speed data has confirmed the robustness of optimal EWOA-MPPT. In conclusion, the simulation results revealed that the EWOA is a promising algorithm to be applied for solving different engineering problems.

1. Introduction

In today's world, there is a significant revolution of emerging new meta-heuristic algorithms, which are inspired by natural behavior and its societies [1]. These meta-heuristic algorithms help in solving several engineering optimization problems and can reach easily with an efficient way to a global or near-to-global solution of such problem [2]. The merits of the meta-heuristic algorithms are (1) no initial conditions, (2) capable of avoiding stagnation at local optima, and (3) deal efficiently with the nonlinearity problem because of randomization process [3]. However, the conventional methods may stick into a local minimum under a fluctuation or nonlinearity condition of an optimization problem. However, according to the No Free Lunch theory [4], any meta-heuristic algorithm sometimes solves one optimization problem and in the meantime fails to solve other problems. Furthermore, each algorithm is firstly published as an original version, where the researchers can later improve and apply it to solve different engineering problems (e.g., particle swarm optimizer (PSO)) [5–7]. Therefore, different meta-heuristic algorithms, which are inspired by nature, have emerged to solve the optimization problems. Some of these algorithms imitate the physical laws such as thermal exchange [8], electromagnetic field [9], water evaporation [10], and lightning attachment [11]. Other algorithms stimulate society behavior, such as the colony of ants and bees [12,13] and fishes [14,15]. Furthermore, the researchers stimulated many algorithms either from the competition between creatures [16] or beings of creatures [17]. The decision-making [18] and learning manners of human [19] are other types of algorithms. Moreover, the fertilization [20,21] and growth [22] of plants are inspired too. Besides, other efficient algorithms are imitating Darwin's evolution concept, for example, genetic and differential evolution algorithms [23,24]. Finally, many meta-heuristic algorithms are inspiring the hunting procedure of predators, for example, wolves [25], coyotes [26], and humpback whales [27].

Whale optimization algorithm (WOA) is a new meta-heuristic algorithm published in 2016 by Mirjalili and Lewis [27]. The humpback whales hunt a school of fishes by diving in-depth, then make a bubble-net and swim-up in a spiral manner around the hunted fishes. The steps of WOA are searching for prey, encircling, then hunting the prey. The WOA is published as a first version and tested using twenty-three well-known benchmark functions. Many researchers have proposed an improvement to the WOA either by modification or hybridization with other algorithms to merge the merits of algorithms. Researchers have hybridized the WOA with many algorithms such as differential evolution (DE) algorithm [28–30], grey wolf optimizer (GWO) [31], moth flame optimization [32], and simulated annealing [33]. However, the hybridization between the two algorithms increases the computational complexity of optimization. Other researchers have

* Corresponding author.
E-mail addresses: mqais@ksu.edu.sa (M.H. Qais), hanyhasanien@ieee.org (H.M. Hasanien), saadalgh@ksu.edu.sa (S. Alghuwainem).

https://doi.org/10.1016/j.asoc.2019.105937
1568-4946/

improved the WOA using many methods such as the levy-flight strategy [34,35], chaotic maps [36–39], Levy flight and chaotic maps [40], and a local search strategy [41]. In addition to the original WOA, these improved algorithms are applied to solve many engineering problems as depicted in Table 1.

The WOA is applied to solve several engineering optimization problems such as modeling of photovoltaic (PV) modules [29,39,46–48], maximum power point tracking (MPPT) of partial shaded PVs [49,50], optimum location of Thyristor controlled series compensator and static VAR compensator [51], optimum power flow [52], automatic generation control of two interconnected power grids [53], communication systems [54,55], feature selection [56–59], flow shop scheduling [60], optimal Mamdani FLCs for vibration regulator of nonlinear steel building [61], optimal locating and sizing of batteries in power systems [43], tuning PID controllers on the automatic voltage regulators [62], elderly assessment of insulated gate bipolar transistor [63], and wind speed forecasting [64]. Based on the recent authors' knowledge, no one has applied the WOA to optimize the Takagi–Sugeno fuzzy logic controllers (TS-FLCs) in the control system of variable-speed wind generators.

The main contributions of this paper are summarized as follows:

(1) Enhancing WOA for higher wind power production using optimal TS-FLCs in the control system of variable speed wind generators.

(2) Twenty-three benchmark functions have tested the exploration and exploitation of the enhanced whale optimization algorithm (EWOA)).

(3) This study has compared the average and standard deviation results of EWOA with other algorithms (WOA [27], GWO [25], augmented GWO (AGWO) [65], salp swarm algorithm (SSA) [66], enhanced SSA (ESSA) [67], and PSO).

(4) The non-parametric statistical test (Wilcoxon signed-rank test) has verified the superiority of the EWOA among other algorithms at significance level 0.05.

(5) Real wind speed data has verified the robustness of optimal EWOA-MPPT control, where the simulation results revealed that the produced real power using EWOA-MPPT is higher than that obtained by using WOA-MPPT.

2. WOA algorithm

The WOA is motivated by the behavior of humpback whales to get their food [27]. Due to the big size of whales, they look for a school of fish. The whales begin hunting by swimming down to the deepness of sea then create bubble-net around the school of fish then swim-up in a spiral manner and get the food, as depicted in Fig. 1. Therefore, Mirjalili and Lewis proposed mathematical modeling for the WOA as follows: (i) looking for the food (exploration) as in (1) for $|A| > 1$, (ii) surrounding the food as in (2) for $|A| < 1$, and (iii) creating spiral bubble-net to hunt the food as in (3). Fig. 2-(a) describes the flowchart of WOA.

$$X_{t+1} = X_{rand} - A.D_{rand} \quad \text{if } |A| > 1$$
$$\text{where } D_{rand} = |C.X_{rand} - X_t| \tag{1}$$

$$X_{t+1} = X^* - A.D \quad \text{if } |A| < 1 \quad \text{where } D = \left|C.X^* - X_t\right| \tag{2}$$

where

$$A = 2.a.r - 2,\ a = 2 - 2(t/t_{\max}),\ \text{and } C = 2.r$$

$$X_{t+1} = X^* + D'.e^{bl}.\cos(2\pi l) \tag{3}$$

where

$$D' = \left|X^* - X_t\right|, l = (a_2 - 1)^* rand + 1,\ \text{and } a_2 = -1 - t/t_{\max}$$

where Factor A is varying randomly between $[-2, 2]$, l is a random number in $[-1,1]$, t is the current iteration, t_{max} is the permissible iterations, b is a tuned number which is proposed to be 1, and r is a random number distributed uniformly between [0, 1].

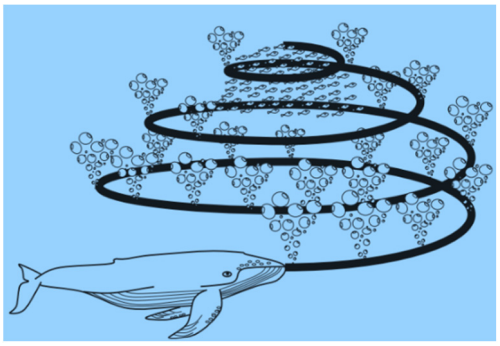

Fig. 1. Spiral bubble-net behavior of humpback whales [27].

3. EWOA algorithm

In this section, new modifications are proposed to enhance the WOA to get an uncomplicated and efficient algorithm named EWOA. First of all, the exploration stage of the WOA relies on the random entity selection from the population as in (1), which is not possible in some of the engineering applications that have only one entity in the population. Therefore, the exploration and the surrounding stages of the WOA are modified based on (4), where a cosine function ($1 \geq \cos(2\pi l) \geq -1$) is used to control the searching and to encircle the food instead of factors ($2 \geq A \geq -2$). For the exploitation stage, making parameter (b) to be an integer random number between [0, 500] will improve it. Fig. 2-(b) illustrates the flowchart of the EWOA. By comparing the flowcharts of the WOA and EWOA, it is evident that EWOA has less conditional statements and parameters.

$$X_{t+1} = X^* - \cos(2\pi l).D_{new} \quad \text{if } r_1 < 0.5 \text{ where}$$
$$D_{new} = (C.X^* - X_t) \tag{4}$$

$$X_{t+1} = X^* + D'.e^{bl}.\cos(2\pi l) \quad \text{if } r_1 \geq 0.5 \text{ where}$$
$$D' = \left|X^* - X_t\right| \tag{5}$$

where
$b = randi(500)$, $l = (a_2 - 1)^* rand + 1$, $a_2 = -1 - t/t_{\max}$, and r_1 is a random number between [0, 1]. The motivations of enhancing the WOA are: (i) the EWOA has a lower number of equations; (ii) It is problematic to use WOA efficiently in PSCAD/EMTDC software due to the using of only one search agent; (iii) the parameter b is integer random in EWOA, while it is constant in WOA.

4. Verification of the EWOA

This paper confirmed the robustness of the proposed EWOA using twenty-three test functions that are listed in Tables 2–4 [68]. These functions are categorized to examine the exploration and exploitation of the EWOA. The functions, which listed in Table 2, are uni-modal functions, which tests the exploitation performance of EWOA. Also, the functions in Tables 3 and 4 are multi-modal functions, which checks the exploration performance of EWOA. The modeling and testing of EWOA are done using MATLAB R2017b [69] on a PC Intel Core i-7, RAM 8 GB, Windows 7). This paper compared the average and standard deviation results of EWOA for 23 test functions with the results of other recent algorithms like WOA, GWO, SSA, PSO, AGWO, and ESSA. The average and standard deviation results are computed for 50 independent runs, 30 search agents, and 500 number of maximum iterations. The specific parameters of the compared algorithms are depicted in Table 5.

For the exploration test, it is evident that the EWOA hits best scores (5/7) compared with other algorithms, as in Table 6. For

Table 1
Some of the proposed methods for WOA improvement.

Algorithm	Year	Method	Complexity	Application
IWOA [28]	2019	WOA + DE	High	Benchmark functions
DEWCO [30]	2019	WOA + DE + chaotic maps	High	Benchmark functions
QIWOA [42]	2019	WOA + Quadratic interpolation	Moderate	Benchmark functions
IWOA [43]	2019	Gaussian + DE + crowding factor	Very high	Locating Electric vehicles charging stations
CWOA [36]	2019	WOA + Ten chaotic maps	Moderate	Permanent magnet motor modeling
BWOA [40]	2019	WOA + Levy Flight + chaotic maps	High	Benchmark functions + engineering
CCMWOA [44]	2019	WOA + Chaotic maps + Gaussian mutation	High	Benchmark functions
IWOA [45]	2019	Elite Opposition-Based Learning + DE + support vector machine	High	Feature selection in Arabic sentimentality study
MWOA [34]	2018	WOA + Levy flight	High	Benchmark functions
IWOA [46]	2018	Two prey searching technique	Moderate	Photovoltaic modeling
Proposed EWOA		**Reducing and Modifying the equations of WOA and reducing its control parameters**	**Low**	**Maximum power point tracking of wind generators + benchmark functions**

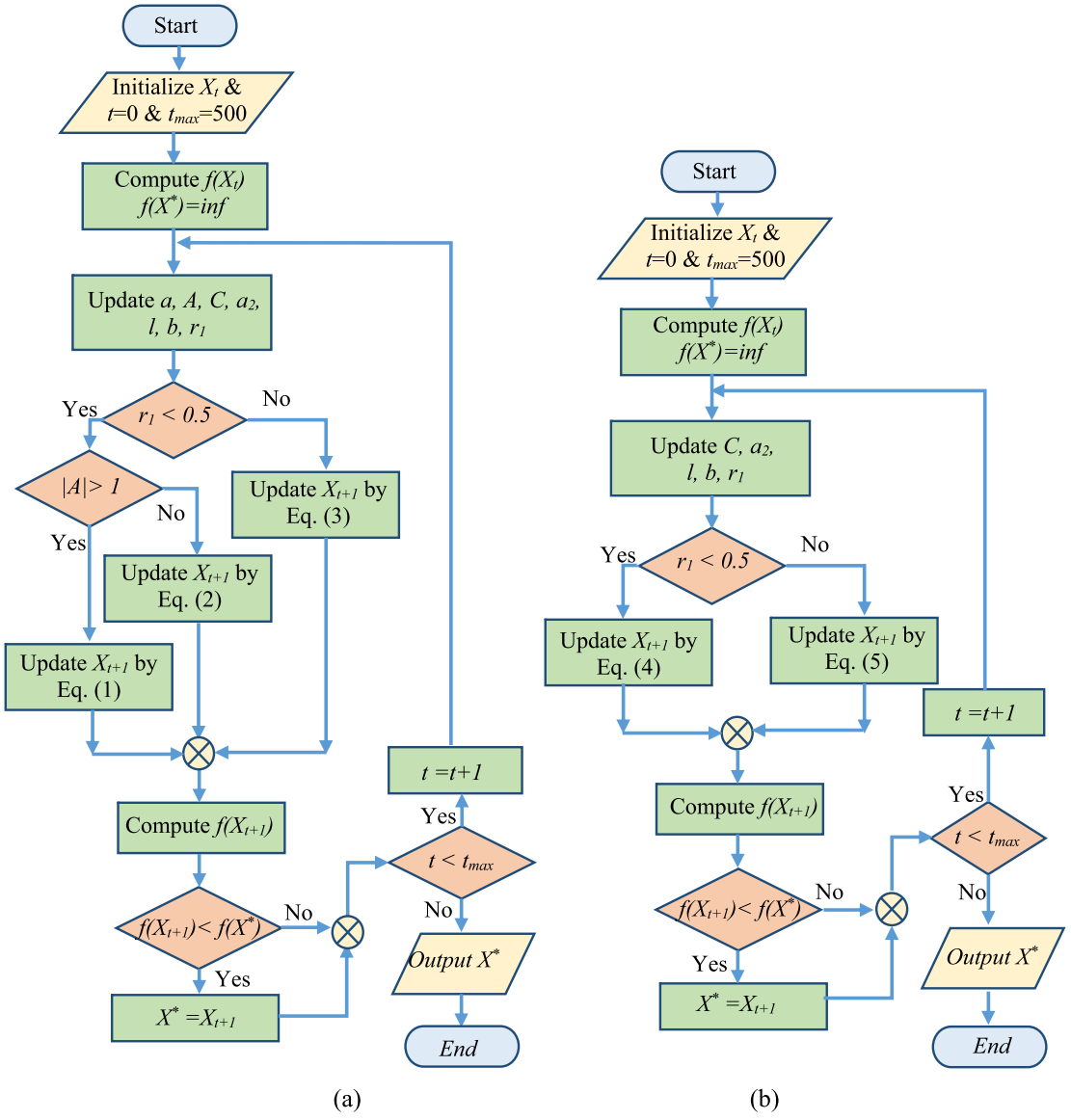

Fig. 2. Flowchart of (a) WOA; (b) EWOA.

the exploitation test, the EWOA hits the best minimum (7/16) as compared with other algorithms, as shown in Tables 7 and 8. This paper used the non-parametric sign test (Wilcoxon signed-rank test) to compute the rank of each algorithm for all benchmark functions. Table 9 shows that EWOA has the 1st rank among other algorithms. Furthermore, the Wilcoxon signed-rank test used to compute the p-value at 5% significant level to validate the superiority and significance of the EWOA. Table 10 shows the p-values of EWOA versus other algorithms, where all p-values are <0.05 except the p-value of EWOA vs. ESSA is <1. Table 11 shows the total CPU time of the compared algorithms for all 23 benchmark functions, where PSO has the smallest time. On

Table 2
Uni-modal functions.

Function	Dimension	Range	$f_{\min}$
$f_1 = \sum_{i=1}^{n} x_i^2$	30	[−100, 100]	0
$f_2 = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[−10, 10]	0
$f_3 = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	30	[−100, 100]	0
$f_4 = \max_i \{\lvert x_i \rvert, 1 \le i \le n\}$	30	[−100, 100]	0
$f_5 = \sum_{i=1}^{n} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[−30, 30]	0
$f_6 = \sum_{i=1}^{n} (x_i + 0.5)^2$	30	[−100, 100]	0
$f_7 = \sum_{i=1}^{n} i x_i^4 + random[0, 1)$	30	[−1.28, 1.28]	0

the other hand, the convergence of EWOA and other algorithms toward the best solution is shown in Fig. 3. The convergence is plotted with the function evaluations (Fes = Search agents × Iterations = 30 × 500 = 15 000), where EWOA converges faster to the best solution than other algorithms. It is found that the proposed EWOA is very competitive to other algorithms.

5. MPPT of variable speed wind generators

In this section, the offered EWOA and WOA are applied to improve the MPPT control strategy of variable speed wind generators. The grid-connected gearless permanent magnet synchronous generator driven by a variable speed wind turbine (VSWT-PMSG) is modeled and simulated using PSCAD/EMTDC software. The grid-connected VSWT-PMSG model, as illustrated in Fig. 4. Wind turbines arrest wind power and transform it into mechanical energy. Then this mechanical energy is transformed into electrical energy using PMSG. Finally, the electrical energy is transmitted to the grid through frequency converters.

5.1. Wind turbine modeling

The leading type of wind turbines for high wind power construction is the horizontal-axis wind turbine. It converts the kinetic wind energy to rotational energy by its blades. The aerodynamic power P_w of air density ρ flowing at speed v_w via a circular area A, as illustrated in (6)

$$P_w = \frac{1}{2}\rho A v_w^3 \tag{6}$$

Therefore, the rotational (mechanical) power P_M of the turbine shaft, as illustrated in (7)

$$P_M = P_w C_P \tag{7}$$

where the relation between P_M and P_w is called a power coefficient C_P, which depends on the pitch angle β and tip speed ratio (TSR) or λ as illustrated in (8). The TSR is the relation of blade tip velocity to the wind velocity, as illustrated in (10) [70].

$$C_P(\lambda, \beta) = 0.73\left(\frac{151}{\lambda_i} - 0.58\beta - 0.002\beta^{2.14} - 13.2\right) e^{-\frac{18.4}{\lambda_i}} \tag{8}$$

$$\frac{1}{\lambda_i} = \frac{1}{\lambda + 0.02\beta} - \frac{0.03}{1 + \beta^3} \tag{9}$$

$$\lambda = \frac{R\omega_m}{v_w} \tag{10}$$

where R is the blade span, and ω_m is the mechanical shaft velocity.

Regarding (7), the maximum traced power P_{max} by wind turbines can be attained by the optimum value of power coefficient C_{Popt} and λ_{opt} where the value of wind velocity is uncontrollable, and the swept area of blades is constant as illustrated in (11).

$$P_{\max} = \frac{1}{2}\rho A\left(\frac{\omega_m R}{\lambda_{opt}}\right)^3 C_{Popt} \tag{11}$$

5.2. PMSG modeling

The shaft of PMSG linked to the wind turbine without a gearbox to eliminate its losses. Therefore, the gearless drive-train model considered as a single-mass shaft model as in (12)

$$J\frac{d\omega_m}{dt} + D\omega_m = T_m - T_e \tag{12}$$

Table 3
Multi-modal functions.

Function	Dim	Range	$f_{\min}$
$f_8 = \sum_{i=1}^{n} x_i \sin(\sqrt{\lvert x_i \rvert})$	30	[−500, 500]	−418.9829 × D
$f_9 = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[−5.12, 5.12]	0
$f_{10} = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{j=1}^{n} x_j}\right) - \exp\left(\frac{1}{n}\cos(2\pi x_j)\right) + 20 + e$	30	[−32, 32]	0
$f_{11} = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[−600, 600]	0
$f_{12} = \frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{n}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	[−50, 50]	0
$f_{13} = 0.1\{10\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\}$ $+ \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	[−50, 50]	0

Table 4
Fixed-dimension multi-modal functions.

Function	Dim	Range	f_{min}
$f_{14}=(\frac{1}{500}+\sum_{i=1}^{25}\frac{1}{i+\sum_{j=1}^{2}(x_i-a_{ij})^6})^{-1}$	2	[−65, 65]	1
$f_{15}=\sum_{i=1}^{11}[a_i-\frac{x_i({b_i}^2+b_ix_2)}{{b_i}^2+b_ix_3+x_4}]^2$	4	[−5, 5]	0.00030
$f_{16}=4{x_1}^2-2.1{x_1}^4+\frac{1}{3}{x_1}^6+x_1x_2-4{x_2}^2+4{x_2}^4$	2	[−5, 5]	−1.0316
$f_{17}=(x_2-\frac{5.1}{4\pi^2}{x_1}^2+\frac{5}{\pi}x_1-6)^2+10(1-\frac{1}{8\pi})\cos(x_1)+10$	2	[−5, 5]	0.398
$f_{18}=[1+(x_1+x_2+1)^2(19-14x_1+3{x_1}^2-14x_2+6x_1x_2+3{x_2}^2)]\times[30+(2x_1-3x_2)^2\times(18-32x_1+12{x_1}^2+48x_2-36x_1x_2+27{x_2}^2)]$	2	[−2, 2]	3
$f_{19}=-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2)$	3	[1, 3]	−3.86
$f_{20}=-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})^2)$	6	[0, 1]	−3.32
$f_{21}=-\sum_{i=1}^{5}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0, 10]	−10.1532
$f_{22}=-\sum_{i=1}^{7}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0, 10]	−10.4028
$f_{23}=-\sum_{i=1}^{10}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0, 10]	−10.5363

Table 5
Specific parameters of algorithms for 500 iterations and 30 populations.

Algorithm	Specific parameters	Value
WOA	Constants b, a, and a2	1, [2,0], [−1,−2]
PSO	Inertia weight w, c1, and c2	[0.5,0.3], 2, and 2
GWO	Constant a	[2,0]
AGWO	Constant a	[2,1]
EWOA	Constants a2	[−1,−2]

where T_m is the mechanical torque (N m), J is the inertia moment of the shaft (kg m^2), D is the damping of the shaft (kg m^2/s), ω_m is the mechanical speed of the shaft (rad/s), and T_e is the electrical torque from the generator (N m).

The stator voltages of generator e_{sd} and e_{sq} are expressed in the direct axis and quadrature axis (dq) as illustrated in (8)

$$\begin{pmatrix}e_{sd}\\e_{sq}\end{pmatrix}=-R_s\begin{pmatrix}i_{sd}\\i_{sq}\end{pmatrix}-\frac{d}{dt}\begin{pmatrix}L_di_{sd}\\L_qi_{sq}\end{pmatrix}+\omega_e\begin{pmatrix}-L_qi_{sq}\\L_di_{sd}+\lambda_r\end{pmatrix} \quad (13)$$

where λ_r is the flux of permanent magnet, L_d and L_q are the dq-axis self-inductances of stator windings, R_s is the resistance of stator windings, ω_e is the electrical angular speed (rad/s), and i_{sd} and i_{sq} are the stator currents. The electrical torque of PMSG as illustrated in (14)

$$T_e=\frac{3}{2}(\frac{p}{2})(\lambda_ri_{sq}-(L_d-L_q)i_{sd}i_{sq}) \quad (14)$$

where p is the poles number.

5.3. Frequency converter

The variable-speed wind generator is linked to the grid via a frequency converter, as illustrated in Fig. 4. This converter includes two voltage source converters (VSCs) connected in back to back way, where one of them operates as a rectifier, and the other operates as an inverter [71]. The VSC is a two-level converter made of six insulated-gate bipolar transistors (IGBTs) paralleled with anti-aliased diodes A DC-link capacitor (40 mF) is connecting the converters [72]. The cascaded controller activates the converters by pulse-width-modulation (PWM) signal, as shown in Fig. 5 [73]. The rectifier, which connected to the PMSG, is responsible for maximum power point tracking (MPPT) from the wind turbine generation system.

5.4. MPPT control

In order to run the generator side converter in a rectifier mode, the flux oriented control is used to produce PWM to trigger the IGBTs of the rectifier. Cascaded control of outer and inner loops is used to produce dq-axis reference voltages, as shown in Fig. 5. The d-axis current i_{sd} is controlled to be zero for a unity power factor operation as in (15). In (16), the real power is proportional to the q-axis current i_{sq}. Therefore, controlling the real power (P_{pmsg}) with the maximum power P_{max} generates the reference q-axis current, as expressed in (17). The produced reference dq-axis voltages, as in (18), converted to ABC waveforms, then compared with the triangular waveform to generate PWM. Three fuzzy logic controllers (FLCs) are used to regulate the input errors for MPPT and a unity power factor operation, as indicated in Fig. 5.

$$i_{sd}^*=(k_{p3}+\frac{k_{i3}}{s})(Q_{ref}-Q_{pmsg})=0 \quad (15)$$

$$P_{pmsg}=w_rT_e=w_r\frac{3}{2}(\frac{p}{2})(\lambda_ri_{sq}) \quad (16)$$

$$i_{sq}^*=(k_{p1}+\frac{k_{i1}}{s})(P_{\max}-P_{pmsg}) \quad (17)$$

$$\begin{aligned}v_{sd}^*&=-(k_{p2}+\frac{k_{i2}}{s})(i_{sd}^*-i_{sd})-\omega_eL_qi_{sq}\\v_{sq}^*&=-(k_{p4}+\frac{k_{i4}}{s})(i_{sq}^*-i_{sq})+\omega_e(L_di_{sd}+\lambda_r)\end{aligned} \quad (18)$$

5.5. Fuzzy logic controller

The grid-linked wind generator is an extensive nonlinear system and exposed to abnormal conditions, for example, voltage and frequency deviations and short-circuits. Therefore, the fuzzy

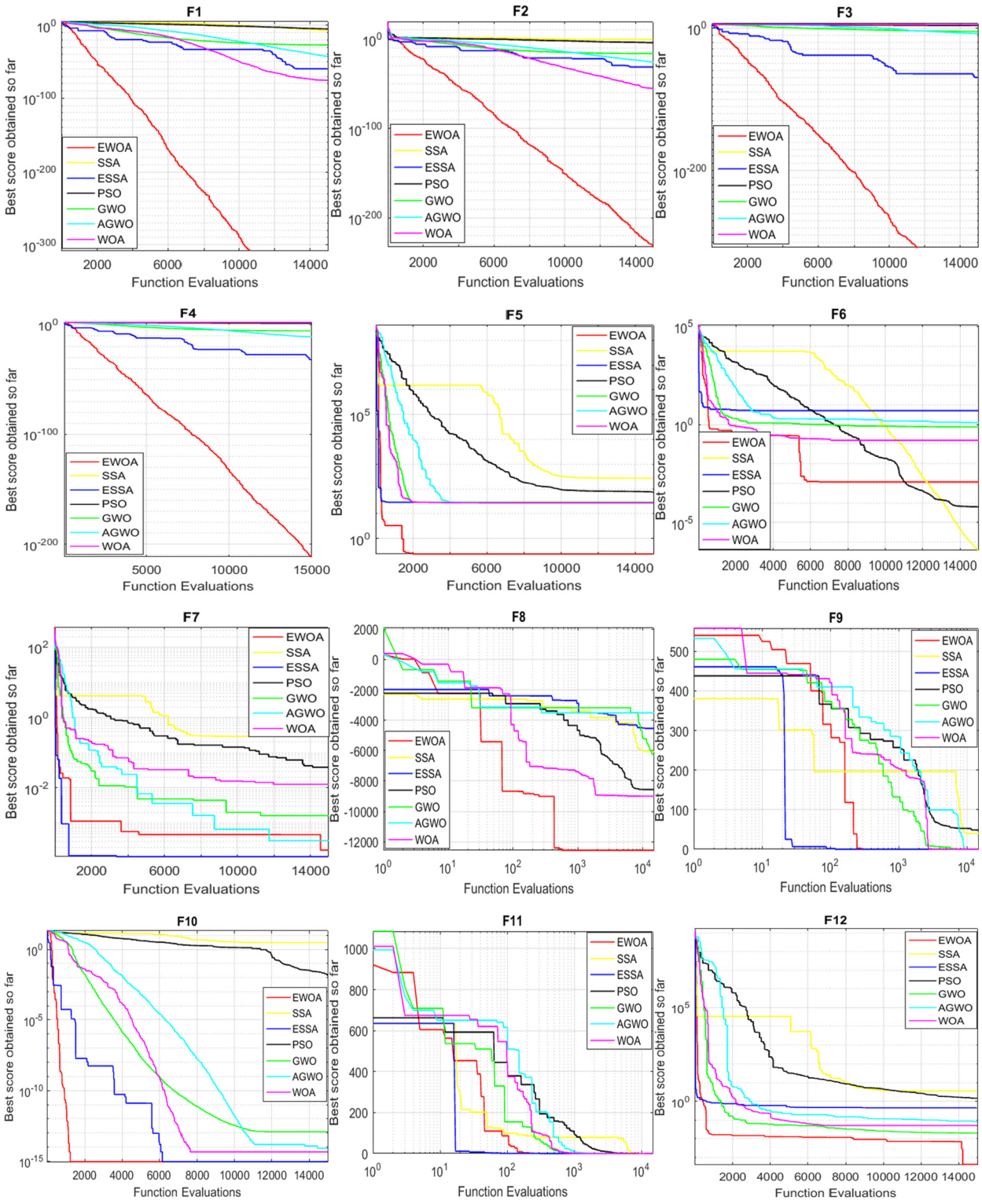

Fig. 3. Convergence behavior of tested algorithms.

logic controller (FLC) is an insensitive controller to these deviations. However, the design of linguistics of FLCs needs expertise people [74]. There are two forms of FLCs: (1) Mamdani FLC: that involves experience and human familiarity, and (2) Sugeno FLC: requires optimization algorithms [75,76]. The Fuzzy process consists of fuzzifying the crisp inputs, if-rules, and de-fuzzify the fuzzy sets to crisp the output data.

The crisp inputs are error *e(k),* and change of error $\Delta e(k)$ and the output is *Z*. The crisp data are converted to fuzzy data in the interval [0, 1] by Gaussian membership functions (MFs) as in (19).

$$\mu(x) = e^{[-0.5(\frac{x-m}{\sigma})^2]} \tag{19}$$

where x is the crisp data input of FLC, m is the mean, and σ is the standard deviation.

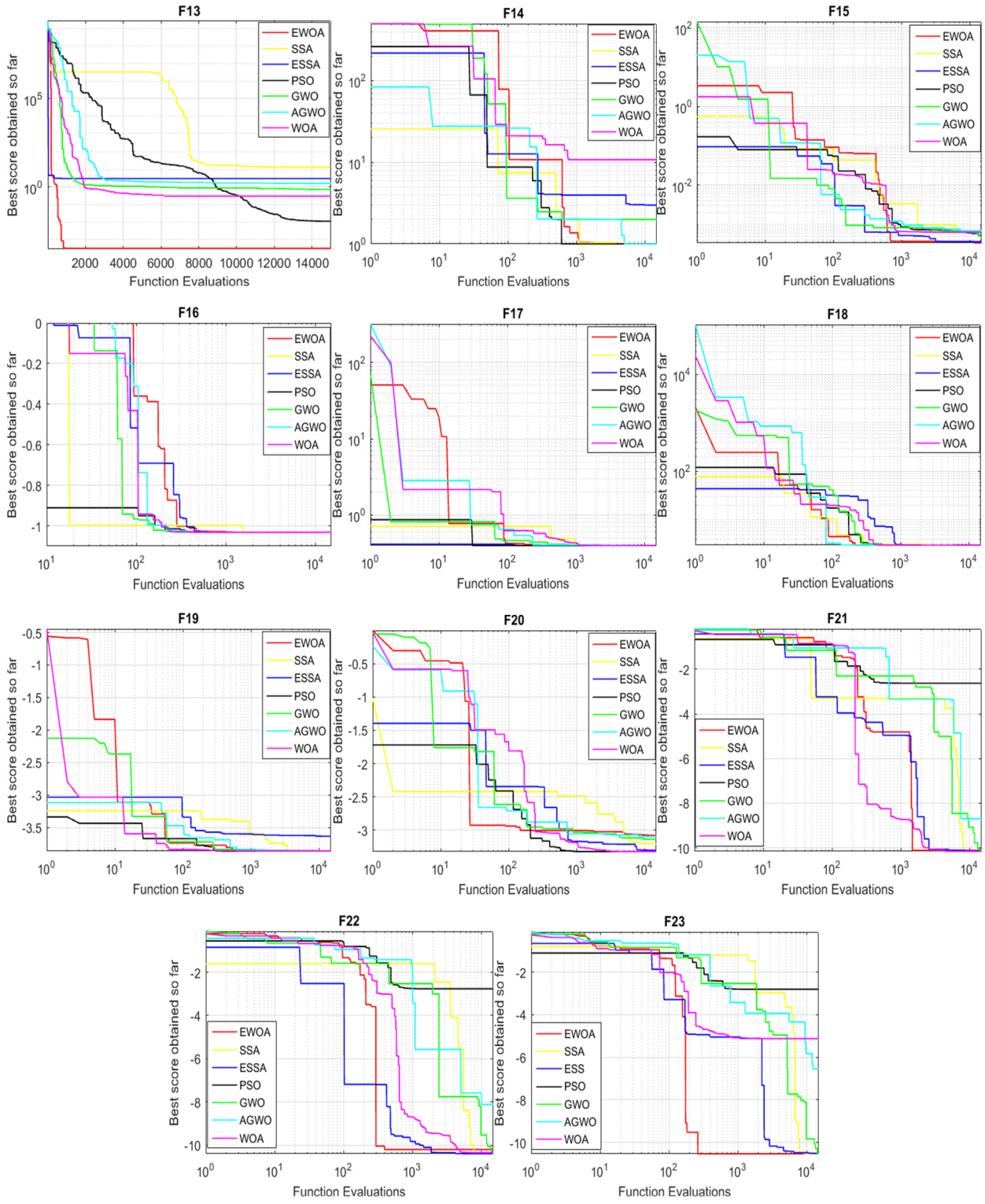

Fig. 3. (*continued*).

The fuzzified inputs are controlled using if-rule statements as in (20), where (*and*) is expressed mathematically by finding the minimum of the input MFs as in (21). The parameter z is a function of x_1 and x_2.

$$If\ x_1\ is\ \mu_1(x_1)\ and\ x_2\ is\ \mu_2(x_2)\ then\ z = f(x_1, x_2) \tag{20}$$

$$w = \min(\mu(x_1), \mu(x_2)) \tag{21}$$

Finally, the output Z of FLC expressed in (22) [77]

$$z = \frac{\sum_{i=1}^{n} w_i z_i}{\sum_{i=1}^{n} w_i} \tag{22}$$

5.6. Optimization methodology

In this section, this paper utilized the EWOA and WOA in obtaining the optimal rules and MFs of Takagi–Sugeno FLCs. The

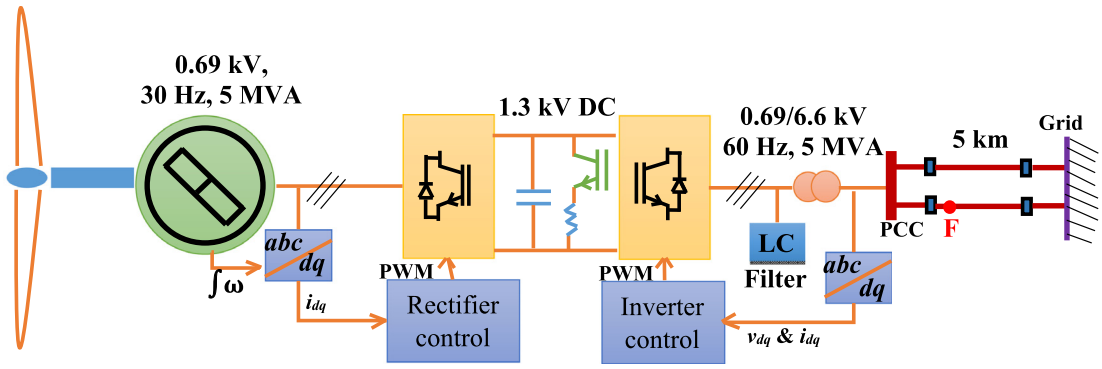

Fig. 4. Grid-linked VSWT-PMSG model.

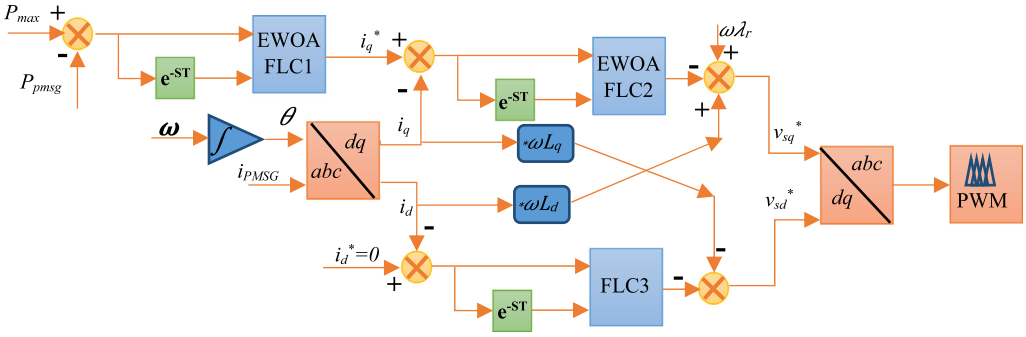

Fig. 5. Rectifier side control for MPPT.

Table 6
Statistical results (average and standard deviations) for uni-modal functions.

		EWOA	WOA	SSA	GWO	PSO	ESSA	AGWO
f1	av	**0.00E+00**	1.00E−75	3.29E−07	1.27E−27	1.31E−05	5.12E−38	6.69E−44
	std	0.00E+00	2.10E−75	5.92E−07	3.11E−27	2.39E−05	2.44E−37	1.33E−43
f2	av	**1.40E−195**	5.50E−51	1.9111	8.52E−17	0.0076	2.01E−25	8.35E−27
	std	0.00E+00	1.44E−50	1.6142	6.62E−17	0.0262	7.74E−25	9.41E−27
f3	av	**0.00E+00**	4.61E+04	1.50E+03	2.43E−05	2.13E+03	3.36E−35	2.98E−08
	std	0.00E+00	1.11E+04	707.0529	8.14E−05	878.7605	1.79E−34	1.08E−07
f4	av	**3.16E−201**	48.8782	2.44E−05	7.69E−07	16.4104	4.25E−18	1.25E−10
	std	0.00E+00	34.2237	1.89E−05	6.51E−07	4.3603	2.1E−17	5.70E−10
f5	av	**2.17E−01**	28.0558	136.5676	27.1786	106.6465	28.8803	26.9654
	std	2.81E−01	0.5234	1.54E+02	0.814	104.3333	0.0298	0.6973
f6	av	2.38E−02	0.41044	**1.72E−07**	0.70757	5.59E−05	3.3387	1.4381
	std	3.36E−02	0.28467	2.44E−07	0.3632	1.42E−04	0.8243	0.3314
f7	av	1.78E−04	1.43E−03	0.169	1.72E−03	6.05E−02	**8.83E−05**	1.42E−03
	std	1.73E−04	1.15E−03	0.0686	1.10E−03	0.0206	6.95E−05	8.21E−04

Table 7
Statistical results (average and standard deviations) for multi-modal functions.

		EWOA	WOA	SSA	GWO	PSO	ESSA	AGWO
f8	av	**−12 569.45**	−9747.22	−7457.2	−5776.12	−9347.8	−4830.1	−3633.36
	std	0.16322	1507.278	634.6745	682.0101	754.4838	709.3098	442.5815
f9	av	**0**	**0**	55.4523	3.1496	58.0837	**0**	0.91397
	std	0	0	18.2751	4.0294	13.1635	0	5.006
f10	av	**8.88E−16**	3.38E−15	2.8401	1.03E−13	0.9353	**8.88E−16**	9.30E−15
	std	0	1.75E−15	0.6581	1.60E−14	0.9909	0	2.55E−15
f11	av	**0**	0.022093	0.2291	6.08E−03	1.97E−02	**0**	1.18E−03
	std	0	0.069865	0.1295	0.0111	0.0232	0	0.0045
f12	av	**7.71E−04**	0.01915	6.8274	0.03669	0.9099	0.2724	0.1024
	std	2.20E−04	0.01614	2.7192	0.0175	1.0678	0.1555	0.0312
f13	av	**1.44E−02**	0.61302	21.3116	0.62239	0.119	2.8525	1.1287
	std	2.49E−02	0.266088	16.9894	0.2837	0.2396	0.3519	0.2193

Table 8
Statistical results (average and standard deviations) for fixed dimension multi-modal functions.

		EWOA	WOA	SSA	GWO	PSO	ESSA	AGWO
f14	av	1.2627	1.3952	1.1301	4.917	**0.998**	3.9752	2.3104
	std	0.68538	0.69402	0.5659	4.125	0	3.6295	2.4804
f15	av	9.58E−04	**5.72E−04**	2.70E−03	5.08E−03	3.20E−03	0.0024	1.84E−03
	std	7.81E−04	3.24E−04	5.43E−03	0.0086	0.0071	0.0053	0.005
f16	av	**−1.0316**	**−1.0316**	**−1.0316**	**−1.0316**	**−1.0316**	**−1.0316**	**−1.0316**
	std	1.15E−12	4.2E−07	5.78E−14	2.64E−08	6.71E−16	6.52E−06	4.17E−06
f17	av	**0.3979**	**0.3979**	**0.3979**	**0.3979**	**0.3979**	**0.3979**	0.3981
	std	8.24E−07	7.58E−06	1.26E−14	8.86E−07	0	1.18E−05	1.39E−05
f18	av	7.5034	**3**	**3**	**3**	**3**	3.0002	**3**
	std	4.9322	4.22E−15	2.66E−13	4.27E−05	1.26E−15	0.000369	3.07E−05
f19	av	−3.7529	−3.85616	**−3.8628**	−3.861	−3.8625	−3.8261	−3.859
	std	0.15693	0.002706	1.13E−11	0.0023	0.0014	0.0533	0.0033
f20	av	−3.0353	−3.2441	−3.22526	**−3.2568**	−3.2518	−3.1703	−3.1858
	std	0.23885	0.176653	0.05772	0.0885	0.1086	0.1191	0.1199
f21	av	−10.0167	−8.3578	−7.33763	−9.3955	−6.2948	**−10.1478**	−6.7613
	std	0.2213	3.629551	3.3158	2	2.9249	0.0143	1.7626
f22	av	−10.228	−7.2907	−8.48023	−10.2241	−7.0398	**−10.4004**	−7.1128
	std	0.27626	3.829202	3.094025	0.9701	3.5751	0.005	1.9801
f23	av	−10.3471	−6.8742	−8.64195	−10.0838	−7.7011	**−10.5331**	−8.1292
	std	0.11708	2.414737	3.1316	1.7514	3.6189	0.0051	1.0174

Table 9
Rank test of algorithms using the Wilcoxon signed-rank test.

	EWOA	WOA	SSA	GWO	PSO	ESSA	AGWO
f1	1	2	6	5	7	4	3
f2	1	2	7	5	6	4	3
f3	1	7	5	4	6	2	3
f4	1	7	5	4	6	2	3
f5	1	4	7	3	6	5	2
f6	3	4	1	5	2	7	6
f7	2	4	7	5	6	1	3
f8	1	2	4	5	3	6	7
f9	2	2	6	5	7	2	4
f10	1.5	3	7	5	6	1.5	4
f11	1.5	6	7	4	5	1.5	3
f12	1	2	7	3	6	5	4
f13	1	3	7	4	2	6	5
f14	3	4	2	7	1	6	5
f15	2	1	5	7	6	4	3
f16	4	4	4	4	4	4	4
f17	4.5	1	4.5	7	4.5	4.5	2
f18	7	3	3	3	3	6	3
f19	7	5	4	2	3	6	1
f20	7	3	4	1	2	6	5
f21	2	4	5	3	7	1	6
f22	2	5	4	3	7	1	6
f23	2	7	4	3	6	1	5
Sum	58.5	85	115.5	97	111.5	86.5	90
Algorithm rank	**(1)**	**(2)**	**(7)**	**(5)**	**(6)**	**(3)**	**(4)**

Table 10
The p-values using Wilcoxon rank sum test for all functions (f_1-f_{23}) at significance $\alpha = 0.05$.

Comparison	p-value	Comparison	p-value
EWOA vs. WOA	0.0361	EWOA vs. PSO	0.0063
EWOA vs. SSA	0.0017	EWOA vs. ESSA	0.0936
EWOA vs. GWO	0.0396	EWOA vs. AGWO	0.0475

Table 11
Total CPU time for all function ($f_1 - f_{23}$).

	EWOA	WOA	SSA	GWO	PSO	ESSA	AGWO
CPU time (s)	3.503	3.522	4.671	5.1794	2.727	4.9667	4.847
Rank	**(2)**	(3)	(4)	(7)	**(1)**	(6)	(5)

optimization methodology is summarized as follows: (1) proposing the cost function *J*, as in (23); (2) applying online optimization to minimize the cost function by updating the parameters of Takagi–Sugeno FLCs. The optimization process is explained in Fig. 6. The Sugeno FLC is designed as 5 × 5 Gaussian MFs, where each MF has two parameters (m, σ). The mean m of five MFs are {−1, −0.5, 0, 0.5, 1}, while EWOA and WOA will optimize σ. The number of σ for the input's MF for FLC ($1 \times 5 \times 2 = 10$), and the rules are ($5 \times 5 = 25$). Also, three gain parameters (k_e, $k_{\Delta e}$, and k_z) are used to amplify the input and output signal of FLCs. The sum of the previous parameters for each FLC is (10+25+3 = 38).

5.6.1. Cost function

The Power system is an extensive nonlinear system, then it is hard to construct mathematical modeling for its cost function. Therefore, in this paper, an online optimization is used to minimize the integral squared-error (ISE) of the measured output real wind power (P_{pmsg}) compared with the demanded maximum power (P_{max}). Then the cost function in this study is the ISE of power error, as expressed in (23). The ISE is reduced by measuring P_{pmsg} and comparing it with P_{max}, then updating the 72nd parameters of FLC1 and FLC2, as depicted in Fig. 6. The parameters of FLCs lie in the range $0.005 \le \sigma \le 2$, $-2 \le z \le 2$, and gain factors $0.05 \le k_{e,\Delta e,z} \le 10$.

$$J(x_1, x_2 \ldots x_{76}) = \int_0^t (P_{\max} - P_{pmsg})^2 dt \tag{23}$$

5.6.2. Optimization process

In this section, the flowchart process of finding the optimum parameters of Sugeno FLCs is shown in Fig. 6. The EWOA, WOA, Sugeno FLCs, and grid-linked VSWT-PMSG are modeled using PSCAD/EMTDC program. The optimization process begins with initializing the parameters of Sugeno FLCs and assesses the cost function using the EWOA or WOA then these algorithms update the parameters of Sugeno FLCs. The number of optimization iterations is 2000.

5.6.3. Optimization results

In this section, the convergence curves of EWOA and WOA are illustrated in Fig. 7. The EWOA converges faster than WOA to the best minimal. Furthermore, the EWOA have gained a lower value of the cost function (0.516234644) than that obtained by using the WOA (0.534056674). The obtained optimal parameters of Sugeno FLCs are listed in Tables 12 and 13. Also, the standard

Table 12
Optimal standard deviations and control rules of FLC1 using EWOA.

e	Δe				
	(−1, 0.016045)	(−0.5, 0.90524)	(0, 0.042719)	(0.5, 0.712543)	(1, 0.005)
(−1, 1.765615)	0.060763774	2	−0.154308148	0.507649453	1.077030014
(−0.5, 0.005)	−0.080334197	−0.415047471	−0.083952012	−0.008641999	0.640620883
(0, 0.005)	1.560599201	0.330417554	1.179103002	0.359712048	−0.087119681
(0.5, 0.014646)	0.495321321	−0.031904792	0.264479909	0.962619817	−0.017362488
(1, 0.005)	0.073283938	−0.049189361	−0.001817271	−0.169521776	1.516985395

Table 13
Optimal standard deviations and control rules of FLC2 using EWOA.

e	Δe				
	(−1, 0.043217)	(−0.5, 0.084122)	(0, 1.326148)	(0.5, 0.035743)	(1, 0.96471)
(−1, 0.012387)	−0.028003168	−0.145111788	−0.045015233	−1.353974731	0.230734452
(−0.5, 0.452082)	−2	−1.26601232	−2	−0.023829808	−0.274670013
(0, 0.275122)	−0.423670372	−0.040482105	0.003265903	−0.002755811	0.099417236
(0.5, 0.721378)	−0.012573265	−0.005734818	1.519584671	−0.882633962	−0.017744969
(1, 0.006446)	−0.081830408	−0.103716248	0.701835551	−0.000462848	0.158707836

Table 14
Optimal gain parameters of FLCs using EWOA.

	K_e	$K_{\Delta e}$	K_z
FLC 1	10	2.646555052	1.494634238
FLC 2	5.763653999	0.618603699	5.318642585

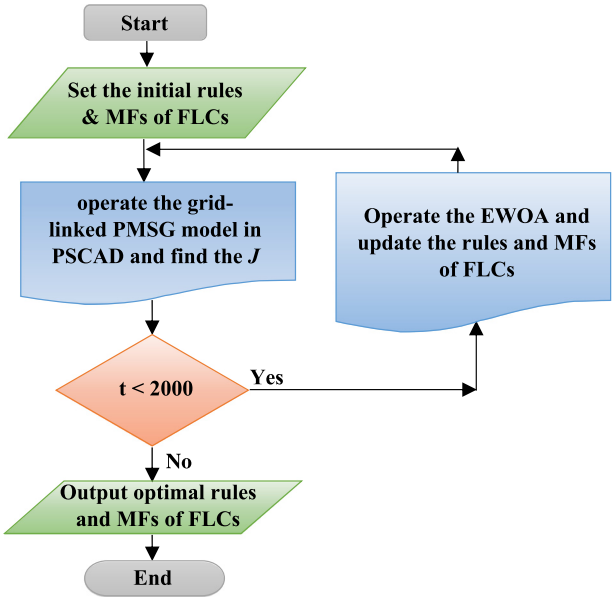

Fig. 6. Optimization methodology of FLCs.

deviation of inputs of FLC1 and 2 are illustrated in Figs. 8 and 9. The gain factors of FLC1 and FLC2 are listed in Table 14. These optimal parameters of FLCs can be expressed in linguistic form for Mamdani FLCs, for example, values that near (2) are large positive (LP) and values near zero (ZE), and values near (−2) are large negative (LN).

5.7. Simulation results of MPPT

In this Section, the optimal Sugeno FLCs are utilized for performing the MPPT control strategy of grid-linked VSWT-PMSG that is illustrated in Fig. 4. The studied system is modeled and simulated using PSCAD/EMTDC program. The real variable wind speed, which is measured in Zaafarana wind farm, is used to investigate the MPPT of designed FLCs. The wind speed data is measured on 28 November 2017 from 12:00 pm to 10:00 pm, where each 10 (min) is considered 10 (s) in this simulation study. For more investigations, a wide range of wind speed variation between 9.3 m/s to 13.8 m/s is used, as illustrated in Fig. 10-a. Besides, the designed rated wind speed of the wind generator is 12 m/s. If the wind speed is above the rated speed, then the pitch angle of blades will be increased to reduce the amount of incident wind, as illustrated in Fig. 10-b, to prevent the rotor speed from exceeding the rated rotational speed (1 pu) as illustrated in Fig. 10-c. The delivered real power to the grid is illustrated in Fig. 10-d. The power produced using EWOA-MPPT (0.961 pu) is higher than that obtained by using WOA-MPPT (0.942 pu). Extra 2% of power is generated using the EWOA.

6. Conclusion

This paper offers a straightforward enhancement to the searching and exploitation of the whale optimization algorithm (WOA). Besides, the control parameters of WOA are reduced. Twenty-three well-known benchmark functions tested the effectiveness of the proposed EWOA. Where the statistical average and standard deviation results of the EWOA compared with that of other algorithms (WOA, GWO, SSA, ESSA, AGWO, and PSO). For further verification, the non-parametric statistical test (Wilcoxon signed-rank test) utilized in testing the superiority of EWOA. On the other hand, the EWOA and WOA are applied to design optimal Takagi–Sugeno FLCs to improve the MPPT control of variable speed wind generator. The grid-linked PMSG driven by a variable-speed wind turbine is modeled and simulated using PSCAD/EMTDC software. The experimental wind speed data are used to verify the robustness of EWOA-MPPT control. The obtained simulation results revealed that the delivered real power to the grid using EWOA-MPPT is higher than that obtained by using WOA-MPPT. Therefore, the simplicity and the attained results encourage the researchers to apply the proposed EWOA for solving different engineering problems. For future work, the EWOA will be applied to improve the MPPT of grid-connected and partial shaded photovoltaic systems.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.asoc.2019.105937.

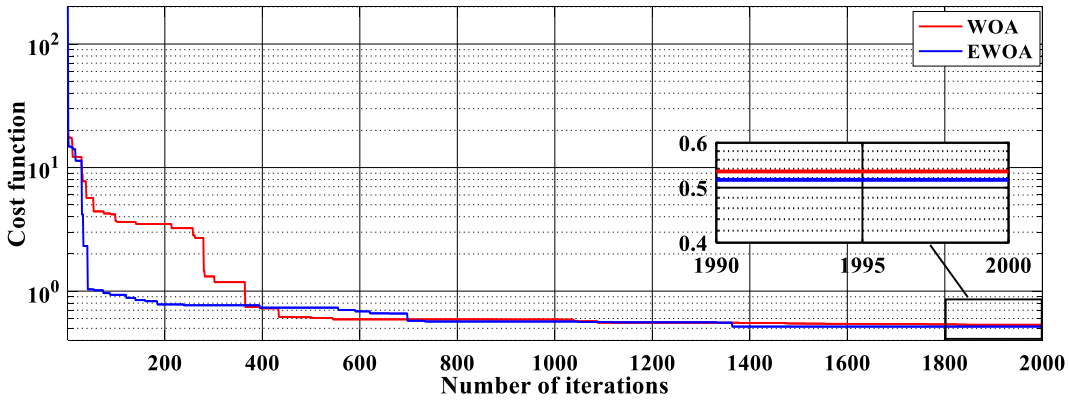

Fig. 7. Convergence curves of EWOA and WOA.

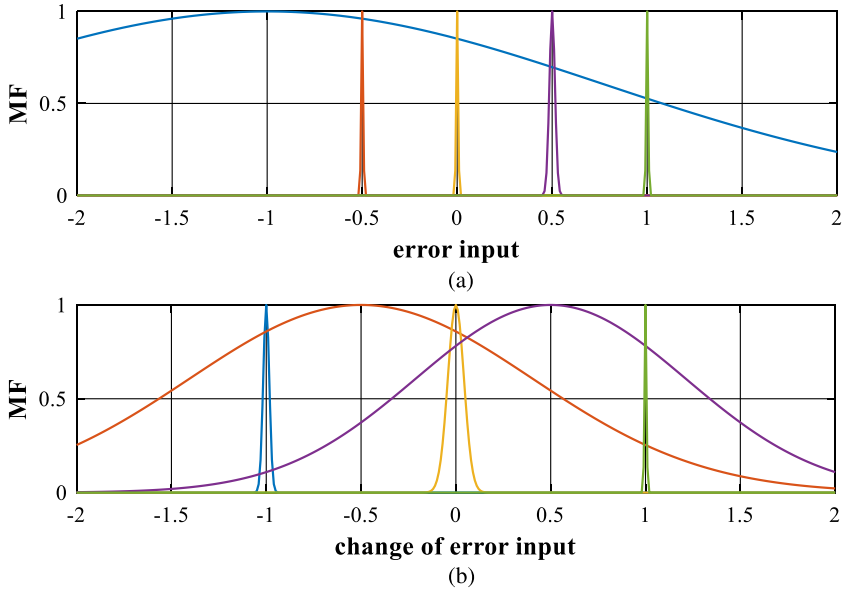

Fig. 8. Standard deviation of Gaussian membership functions of FLC 1.

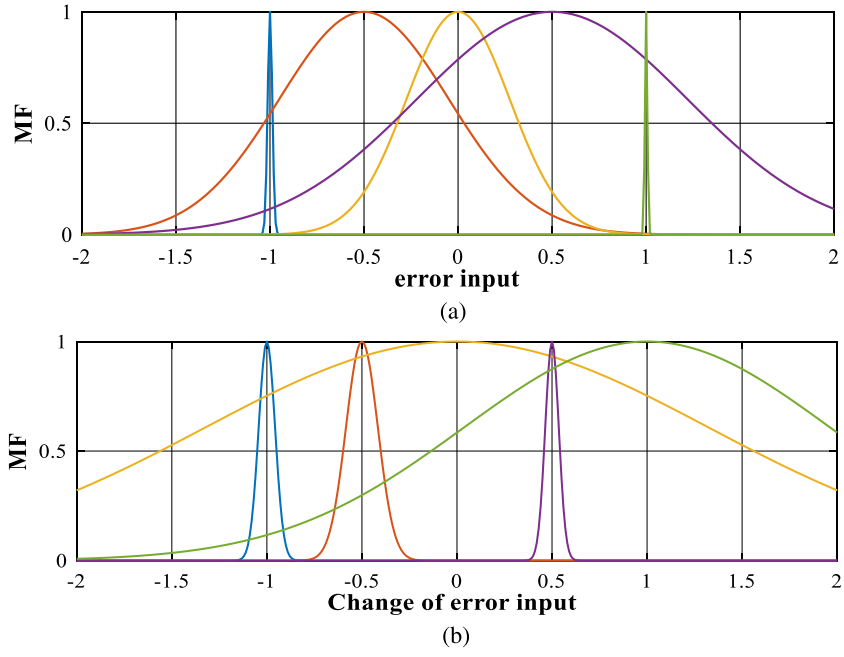

Fig. 9. Standard deviation of Gaussian membership functions of FLC 2.

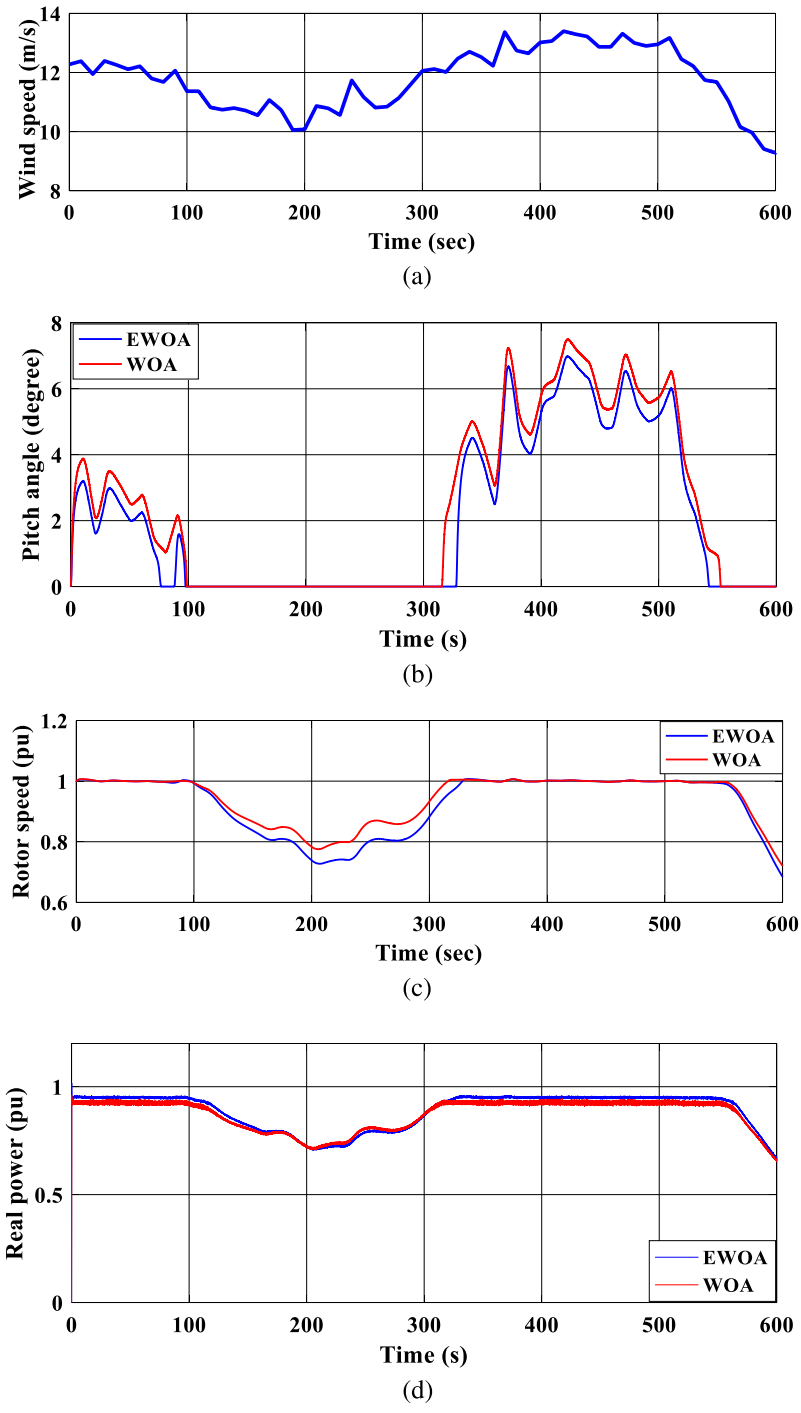

Fig. 10. Time response of grid-linked VSWT-PMSG: (a) wind speed; (b) pitch angle; (c) rotor speed; (d) real power.

Acknowledgment

The authors extend their gratitude to the Deanship of Scientific Research at King Saud University for funding this paper through research group No (RG-1440-049).

References

[1] S.J. Nanda, G. Panda, A survey on nature inspired metaheuristic algorithms for partitional clustering, Swarm Evol. Comput. 16 (2014) 1–18, http://dx.doi.org/10.1016/j.swevo.2013.11.003.

[2] V. Rafe, M. Moradi, R. Yousefian, A. Nikanjam, A meta-heuristic solution for automated refutation of complex software systems specified through graph transformations, Appl. Soft Comput. 33 (2015) 136–149, http://dx.doi.org/10.1016/j.asoc.2015.04.032.

[3] W.J.S. Gomes, A.T. Beck, R.H. Lopez, L.F.F. Miguel, A probabilistic metric for comparing metaheuristic optimization algorithms, Struct. Saf. 70 (2018) 59–70, http://dx.doi.org/10.1016/j.strusafe.2017.10.006.

[4] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1997) 67–82, http://dx.doi.org/10.1109/4235.585893.

[5] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Neural Networks, 1995. Proceedings. IEEE Int. Conf. 1995: pp. 1942–1948. http://dx.doi.org/10.1109/ICNN.1995.488968.

[6] N.A.Ab. Aziz, Z. Ibrahim, M. Mubin, S.W. Nawawi, M.S. Mohamad, Improving particle swarm optimization via adaptive switching asynchronous – synchronous update, Appl. Soft Comput. 72 (2018) 298–311, http://dx.doi.org/10.1016/j.asoc.2018.07.047.

[7] H. Shi, S. Liu, H. Wu, R. Li, S. Liu, N. Kwok, Y. Peng, Oscillatory particle swarm optimizer, Appl. Soft Comput. 73 (2018) 316–327, http://dx.doi.org/10.1016/j.asoc.2018.08.037.

[8] A. Kaveh, A. Dadras, A novel meta-heuristic optimization algorithm: Thermal exchange optimization, Adv. Eng. Softw. 110 (2017) 69–84, http://dx.doi.org/10.1016/j.advengsoft.2017.03.014.

[9] H. Abedinpourshotorban, S. Mariyam Shamsuddin, Z. Beheshti, D.N.A. Jawawi, Electromagnetic field optimization: A physics-inspired metaheuristic optimization algorithm, Swarm Evol. Comput. 26 (2016) 8–22, http://dx.doi.org/10.1016/j.swevo.2015.07.002.

[10] A. Kaveh, T. Bakhshpoori, Water evaporation optimization: A novel physically inspired optimization algorithm, Comput. Struct. 167 (2016) 69–85, http://dx.doi.org/10.1016/j.compstruc.2016.01.008.

[11] A.F. Nematollahi, A. Rahiminejad, B. Vahidi, A novel physical based meta-heuristic optimization method known as lightning attachment procedure optimization, Appl. Soft Comput. 59 (2017) 596–621, http://dx.doi.org/10.1016/j.asoc.2017.06.033.

[12] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, IEEE Comput. Intell. Mag. 1 (2006) 28–39, http://dx.doi.org/10.1109/MCI.2006.329691.

[13] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (2007) 459–471, http://dx.doi.org/10.1007/s10898-007-9149-x.

[14] E. Jahani, M. Chizari, Tackling global optimization problems with a novel algorithm – Mouth Brooding Fish algorithm, Appl. Soft Comput. 62 (2018) 987–1002, http://dx.doi.org/10.1016/j.asoc.2017.09.035.

[15] C.S. Fonseca, F.A.B.S. Ferreira, F. Madeiro, Vector quantization codebook design based on Fish School Search algorithm, Appl. Soft Comput. 73 (2018) 958–968, http://dx.doi.org/10.1016/j.asoc.2018.09.025.

[16] Y. Sharafi, M.A. Khanesar, M. Teshnehlab, COOA: Competitive optimization algorithm, Swarm Evol. Comput. 30 (2016) 39–63, http://dx.doi.org/10.1016/j.swevo.2016.04.002.

[17] F. Fausto, E. Cuevas, A. Valdivia, A. González, A global optimization algorithm inspired in the behavior of selfish herds, Biosystems 160 (2017) 39–55, http://dx.doi.org/10.1016/j.biosystems.2017.07.010.

[18] Q. Zhang, R. Wang, J. Yang, K. Ding, Y. Li, J. Hu, Collective decision optimization algorithm: A new heuristic optimization method, Neurocomputing 221 (2017) 123–137, http://dx.doi.org/10.1016/j.neucom.2016.09.068.

[19] L. Wang, H. Ni, R. Yang, P.M. Pardalos, X. Du, M. Fei, An adaptive simplified human learning optimization algorithm, Inf. Sci. (Ny). 320 (2015) 126–139, http://dx.doi.org/10.1016/j.ins.2015.05.022.

[20] X.-S. Yang, Flower Pollination Algorithm for Global Optimization BT - Unconventional Computation and Natural Computation, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 240–249.

[21] M.H. Qais, H.M. Hasanien, S. Alghuwainem, Identification of electrical parameters for three-diode photovoltaic model using analytical and sunflower optimization algorithm, Appl. Energy. 250 (2019) 109–117, http://dx.doi.org/10.1016/j.apenergy.2019.05.013.

[22] Q.Q. Li, K. Song, Z.C. He, E. Li, A.G. Cheng, T. Chen, The artificial tree (AT) algorithm, Eng. Appl. Artif. Intell. 65 (2017) 99–110, http://dx.doi.org/10.1016/j.engappai.2017.07.025.

[23] J.H. Holland, Genetic algorithms, Sci. Am. 267 (1992) 66–72.

[24] R. Storn, K. Price, Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (1997) 341–359, http://dx.doi.org/10.1023/A:1008202821328.

[25] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61, http://dx.doi.org/10.1016/j.advengsoft.2013.12.007.

[26] M.H. Qais, H.M. Hasanien, S. Alghuwainem, A.S. Nouh, Coyote optimization algorithm for parameters extraction of three-diode photovoltaic models of photovoltaic modules, Energy 187 (2019) 116001, http://dx.doi.org/10.1016/j.energy.2019.116001.

[27] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67, http://dx.doi.org/10.1016/j.advengsoft.2016.01.008.

[28] S. Mostafa Bozorgi, S. Yazdani, IWOA: An improved whale optimization algorithm for optimization problems, J. Comput. Des. Eng. 6 (2019) 243–259, http://dx.doi.org/10.1016/j.jcde.2019.02.002.

[29] G. Xiong, J. Zhang, X. Yuan, D. Shi, Y. He, G. Yao, Parameter extraction of solar photovoltaic models by means of a hybrid differential evolution with whale optimization algorithm, Sol. Energy 176 (2018) 742–761, http://dx.doi.org/10.1016/j.solener.2018.10.050.

[30] M.A. Elaziz, S. Mirjalili, A hyper-heuristic for improving the initial population of whale optimization algorithm, Knowl.-Based Syst. 172 (2019) 42–63, http://dx.doi.org/10.1016/j.knosys.2019.02.010.

[31] A.N. Jadhav, N. Gomathi, WGC: Hybridization of exponential grey wolf optimizer with whale optimization for data clustering, Alexandria Eng. J. 57 (2018) 1569–1584, http://dx.doi.org/10.1016/j.aej.2017.04.013.

[32] M.A. El Aziz, A.A. Ewees, A.E. Hassanien, Whale Optimization Algorithm and Moth-Flame Optimization for multilevel thresholding image segmentation, Expert Syst. Appl. 83 (2017) 242–256, http://dx.doi.org/10.1016/j.eswa.2017.04.023.

[33] M.M. Mafarja, S. Mirjalili, Hybrid Whale Optimization Algorithm with simulated annealing for feature selection, Neurocomputing 260 (2017) 302–312, http://dx.doi.org/10.1016/j.neucom.2017.04.053.

[34] Y. Sun, X. Wang, Y. Chen, Z. Liu, A modified whale optimization algorithm for large-scale global optimization problems, Expert Syst. Appl. 114 (2018) 563–577, http://dx.doi.org/10.1016/j.eswa.2018.08.027.

[35] Y. Ling, Y. Zhou, Q. Luo, Lévy Flight trajectory-based whale optimization algorithm for global optimization, IEEE Access 5 (2017) 6168–6186, http://dx.doi.org/10.1109/ACCESS.2017.2695498.

[36] D. Yousri, D. Allam, M.B. Eteiba, Chaotic whale optimizer variants for parameters estimation of the chaotic behavior in Permanent Magnet Synchronous Motor, Appl. Soft Comput. 74 (2019) 479–503, http://dx.doi.org/10.1016/j.asoc.2018.10.032.

[37] G. Kaur, S. Arora, Chaotic whale optimization algorithm, J. Comput. Des. Eng. 5 (2018) 275–284, http://dx.doi.org/10.1016/j.jcde.2017.12.006.

[38] G.I. Sayed, A. Darwish, A.E. Hassanien, A new chaotic whale optimization algorithm for features selection, J. Classification 35 (2018) 300–344, http://dx.doi.org/10.1007/s00357-018-9261-2.

[39] D. Oliva, M. Abd El Aziz, A. Ella Hassanien, Parameter estimation of photovoltaic cells using an improved chaotic whale optimization algorithm, Appl. Energy. 200 (2017) 141–154, http://dx.doi.org/10.1016/j.apenergy.2017.05.029.

[40] H. Chen, Y. Xu, M. Wang, X. Zhao, A balanced whale optimization algorithm for constrained engineering design problems, Appl. Math. Model. 71 (2019) 45–59, http://dx.doi.org/10.1016/j.apm.2019.02.004.

[41] M. Abdel-Basset, G. Manogaran, D. El-Shahat, S. Mirjalili, A hybrid whale optimization algorithm based on local search strategy for the permutation flow shop scheduling problem, Future Gener. Comput. Syst. 85 (2018) 129–145, http://dx.doi.org/10.1016/j.future.2018.03.020.

[42] Y. Sun, T. Yang, Z. Liu, A whale optimization algorithm based on quadratic interpolation for high-dimensional global optimization problems, Appl. Soft Comput. 85 (2019) 105744, http://dx.doi.org/10.1016/j.asoc.2019.105744.

[43] H. Zhang, L. Tang, C. Yang, S. Lan, Locating electric vehicle charging stations with service capacity using the improved whale optimization algorithm, Adv. Eng. Inform. 41 (2019) 100901, http://dx.doi.org/10.1016/j.aei.2019.02.006.

[44] J. Luo, H. Chen, A.A. Heidari, Y. Xu, Q. Zhang, C. Li, Multi-strategy boosted mutative whale-inspired optimization approaches, Appl. Math. Model. 73 (2019) 109–123, http://dx.doi.org/10.1016/j.apm.2019.03.046.

[45] M. Tubishat, M.A.M. Abushariah, N. Idris, I. Aljarah, Improved whale optimization algorithm for feature selection in arabic sentiment analysis, Appl. Intell. 49 (2019) 1688–1707, http://dx.doi.org/10.1007/s10489-018-1334-8.

[46] G. Xiong, J. Zhang, D. Shi, Y. He, Parameter extraction of solar photovoltaic models using an improved whale optimization algorithm, Energy Convers. Manag. 174 (2018) 388–405, http://dx.doi.org/10.1016/j.enconman.2018.08.053.

[47] M. Abd Elaziz, D. Oliva, Parameter estimation of solar cells diode models by an improved opposition-based whale optimization algorithm, Energy Convers. Manag. 171 (2018) 1843–1859, http://dx.doi.org/10.1016/j.enconman.2018.05.062.

[48] O.S. Elazab, H.M. Hasanien, M.A. Elgendy, A.M. Abdeen, Parameters estimation of single- and multiple-diode photovoltaic model using whale optimisation algorithm, IET Renew. Power Gener. 12 (2018) 1755–1761, http://dx.doi.org/10.1049/iet-rpg.2018.5317.

[49] N. Kumar, I. Hussain, B. Singh, B.K. Panigrahi, MPPT In dynamic condition of partially shaded PV system by using WODE technique, IEEE Trans. Sustain. Energy 8 (2017) 1204–1214, http://dx.doi.org/10.1109/TSTE.2017.2669525.

[50] H.M. Hasanien, Performance improvement of photovoltaic power systems using an optimal control strategy based on whale optimization algorithm, Electr. Power Syst. Res. 157 (2018) 168–176, http://dx.doi.org/10.1016/j.epsr.2017.12.019.

[51] S. Raj, B. Bhattacharyya, Optimal placement of TCSC and SVC for reactive power planning using Whale optimization algorithm, Swarm Evol. Comput. 40 (2018) 131–143, http://dx.doi.org/10.1016/j.swevo.2017.12.008.

[52] D. Prasad, A. Mukherjee, G. Shankar, V. Mukherjee, Application of chaotic whale optimisation algorithm for transient stability constrained optimal power flow, IET Sci. Meas. Technol. 11 (2017) 1002–1013, http://dx.doi.org/10.1049/iet-smt.2017.0015.

[53] H.M. Hasanien, Whale optimisation algorithm for automatic generation control of interconnected modern power systems including renewable energy sources, IET Gener. Transm. Distrib. 12 (2018) 607–614, http://dx.doi.org/10.1049/iet-gtd.2017.1005.

[54] A. Kaur, S. Sharma, A. Mishra, Nature-inspired optimisation algorithms assisted realisation of green communication via CR:a comparison study, IET Commun. 12 (2018) 2511–2520, http://dx.doi.org/10.1049/iet-com.2018.5327.

[55] C. Zhang, X. Fu, L.P. Ligthart, S. Peng, M. Xie, Synthesis of broadside linear aperiodic arrays with sidelobe suppression and null steering using whale optimization algorithm, IEEE Antennas Wirel. Propag. Lett. 17 (2018) 347–350, http://dx.doi.org/10.1109/LAWP.2018.2789919.

[56] M. Mafarja, S. Mirjalili, Whale optimization approaches for wrapper feature selection, Appl. Soft Comput. 62 (2018) 441–453, http://dx.doi.org/10.1016/j.asoc.2017.11.006.
[57] Y. Zheng, Y. Li, G. Wang, Y. Chen, Q. Xu, J. Fan, X. Cui, A novel hybrid algorithm for feature selection based on whale optimization algorithm, IEEE Access 7 (2019) 14908–14923, http://dx.doi.org/10.1109/ACCESS.2018.2879848.
[58] G.I. Sayed, A. Darwish, A.E. Hassanien, A new chaotic whale optimization algorithm for features selection, J. Classification 35 (2018) 300–344, http://dx.doi.org/10.1007/s00357-018-9261-2.
[59] M.M. Mafarja, S. Mirjalili, Hybrid Whale Optimization Algorithm with simulated annealing for feature selection, Neurocomputing 260 (2017) 302–312, http://dx.doi.org/10.1016/j.neucom.2017.04.053.
[60] M. Abdel-Basset, G. Manogaran, D. El-Shahat, S. Mirjalili, A hybrid whale optimization algorithm based on local search strategy for the permutation flow shop scheduling problem, Future Gener. Comput. Syst. 85 (2018) 129–145, http://dx.doi.org/10.1016/j.future.2018.03.020.
[61] M. Azizi, R.G. Ejlali, S.A. Mousavi Ghasemi, S. Talatahari, Upgraded Whale Optimization Algorithm for fuzzy logic based vibration control of nonlinear steel structure, Eng. Struct. 192 (2019) 53–70, http://dx.doi.org/10.1016/j.engstruct.2019.05.007.
[62] A.M. Mosaad, M.A. Attia, A.Y. Abdelaziz, Whale optimization algorithm to tune PID and PIDA controllers on AVR system, Ain Shams Eng. J. (2019) http://dx.doi.org/10.1016/j.asej.2019.07.004.
[63] L.-L. Li, J. Sun, M.-L. Tseng, Z.-G. Li, Extreme learning machine optimized by whale optimization algorithm using insulated gate bipolar transistor module aging degree evaluation, Expert Syst. Appl. 127 (2019) 58–67, http://dx.doi.org/10.1016/j.eswa.2019.03.002.
[64] J. Wang, P. Du, T. Niu, W. Yang, A novel hybrid system based on a new proposed algorithm—Multi-Objective Whale Optimization Algorithm for wind speed forecasting, Appl. Energy 208 (2017) 344–360, http://dx.doi.org/10.1016/j.apenergy.2017.10.031.
[65] M.H. Qais, H.M. Hasanien, S. Alghuwainem, Augmented grey wolf optimizer for grid-connected PMSG-based wind energy conversion systems, Appl. Soft Comput. 69 (2018) 504–515, http://dx.doi.org/10.1016/j.asoc.2018.05.006.
[66] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems, Adv. Eng. Softw. 114 (2017) 163–191, http://dx.doi.org/10.1016/j.advengsoft.2017.07.002.
[67] M.H. Qais, H.M. Hasanien, S. Alghuwainem, Enhanced salp swarm algorithm: Application to variable speed wind generators, Eng. Appl. Artif. Intell. 80 (2019) 82–96, http://dx.doi.org/10.1016/j.engappai.2019.01.011.
[68] J.G. Digalakis, K.G. Margaritis, On benchmarking functions for genetic algorithms, Int. J. Comput. Math. 77 (2001) 481–506, http://dx.doi.org/10.1080/00207160108805080.
[69] Mathworks, Matlab2016b, 2016, https://www.mathworks.com.
[70] M.H. Qais, H.M. Hasanien, S. Alghuwainem, A grey wolf optimizer for optimum parameters of multiple PI controllers of a grid-connected PMSG driven by variable speed wind turbine, IEEE Access 6 (2018) 44120–44128, http://dx.doi.org/10.1109/ACCESS.2018.2864303.
[71] M.H. Qais, H.M. Hasanien, S. Alghuwainem, Low voltage ride-through capability enhancement of grid-connected permanent magnet synchronous generator driven directly by variable speed wind turbine: a review, J. Eng. 2017 (2017) 1750–1754, http://dx.doi.org/10.1049/joe.2017.0632.
[72] M.H. Qais, H.M. Hasanien, S. Alghuwainem, Output power smoothing of grid-connected permanent-magnet synchronous generator driven directly by variable speed wind turbine: a review, J. Eng. 2017 (2017) 1755–1759, http://dx.doi.org/10.1049/joe.2017.0633.
[73] Y. Errami, M. Ouassaid, M. Maaroufi, A performance comparison of a nonlinear and a linear control for grid connected PMSG wind energy conversion system, Int. J. Electr. Power Energy Syst. 68 (2015) 180–194, http://dx.doi.org/10.1016/j.ijepes.2014.12.027.
[74] C.M. Salgado, J.L. Viegas, C.S. Azevedo, M.C. Ferreira, S.M. Vieira, J.M.C. Sousa, Takagi–Sugeno Fuzzy modeling using mixed Fuzzy clustering, IEEE Trans. Fuzzy Syst. 25 (2017) 1417–1429, http://dx.doi.org/10.1109/TFUZZ.2016.2639565.
[75] N. Siddique, H. Adeli, Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing, John Wiley & Sons Ltd, Chichester, UK, 2013.
[76] J.M. Keller, D. Liu, D.B. Fogel, Fundamentals of Computational Intelligence: Neural Networks, Fuzzy Systems, and Evolutionary Computation, IEEE, 2016, http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7547452.
[77] S.N. Sivanandam, S. Sumathi, S.N. Deepa, Introduction to Fuzzy Logic using MATLAB, Springer, London, UK, 2007.