Engineering Applications of Artificial Intelligence 92 (2020) 103666

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

A novel and effective optimization algorithm for global optimization and its engineering applications: Turbulent Flow of Water-based Optimization (TFWO)

Mojtaba Ghasemi [a], Iraj Faraji Davoudkhani [b], Ebrahim Akbari [c,*], Abolfazl Rahimnejad [d], Sahand Ghavidel [e], Li Li [e]

[a] Shiraz University of Technology, Shiraz, Iran
[b] Department of Electrical Engineering, Islamic Azad university, Khalkhal, Iran
[c] Young Researchers and Elite Club, Ayatollah Amoli Branch, Islamic Azad University, Amol, Iran
[d] School of Engineering, University of Guelph, Guelph, Canada
[e] Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, Australia

ARTICLE INFO

Keywords:
Real-parameter global optimization
Classical optimization algorithms
Turbulent Flow of Water-based Optimization (TFWO)
Economic Load Dispatch (ELD)

ABSTRACT

In this study we present a new and effective grouping optimization algorithm (namely, the Turbulent Flow of Water-based Optimization (TFWO)), inspired from a nature search phenomenon, i.e. whirlpools created in turbulent flow of water, for global real-world optimization problems. In the proposed algorithm, the problem of selecting control parameters is eliminated, the convergence power is increased and the algorithm have a fixed structure. The proposed algorithm is used to find the global solutions of real-parameter benchmark functions with different dimensions. Besides, in order to further investigate the effectiveness of TFWO, it was used to solve various types of nonlinear Economic Load Dispatch (ELD) optimization problems in power systems and Reliability–Redundancy Allocation Optimization (RRAO) for the overspeed protection system of a gas turbine, as two real-world engineering optimization problems. The results of TFWO are compared with other algorithms, which provide evidence for efficient performance with superior solution quality of the proposed TFWO algorithm in solving a great range of real-parameter benchmark and real-world engineering problems. Also, the results prove the competitive performance and robustness of TFWO algorithm compared to other state-of-the-art optimization algorithms in this study. The source codes of the TFWO algorithm are publicly available at https://github.com/ebrahimakbary/TFWO.

1. Introduction

Over the recent years, many real-world optimization problems required optimization of a large number of control variables with various constraints. Optimization means finding the best strategy to minimize or maximize objective function of a given problem. Mathematical programming techniques, for example Nelder and Mead simplex-based optimization algorithm (Nelder and Mead, 1965), and trust-region quadratic-based models (Winfield, 1969), and also nature-inspired optimization algorithms were used to optimize different problems with some objective functions under some specific constraints and with real-world different complexities such as non-linearity, mixed-integer nature, non-differentiability, etc. Due to the huge size of real-parameter optimization problems, classical mathematical programming techniques do not generally provide good solutions for different real-world or real-parameter optimization problems under some specific limits (Ghasemi et al., 2016a,b).

The meta-heuristic optimization algorithms are the second group of the proposed techniques, which are inspired and modeled by principles and concepts of real-world nature, like collective birds and animal behaviors. Some of these nature-inspired algorithms that were proposed and applied to solve global numerical optimization problems are Genetic Algorithm (GA) (Mitchell, 1999), Particle Swarm Optimization (PSO) (Eberhart and Kennedy, 1995), Differential Evolution (DE) (Storn and Price, 1997), Cooperative co-evolution (CC) with differential grouping (Omidvar et al., 2013), a DE with a successful-parent-selecting framework (Bhattacharjee et al., 2014), adaptive ranking mutation operator based DE (Gong et al., 2014), Honey Bees Optimization (MBO) (Abbass, 2001), Glowworm Swarm Optimization (GSO) (Krishnanand and Ghose, 2009), Opposition-Based Learning Optimization (OBO) (Tizhoosh, 2005), Harmony Search (HS) optimizer (Lee and Geem, 2005), Cat Swarm Optimization (CSO) (Chu and Tsai,

* Corresponding author.
E-mail address: ebrahimakbary@gmail.com (E. Akbari).

https://doi.org/10.1016/j.engappai.2020.103666
Received 8 July 2018; Received in revised form 25 February 2020; Accepted 14 April 2020
Available online 6 May 2020
0952-1976/

2007), Seeker Optimization Algorithm (SOA) (Dai et al., 2006), Cultural Algorithms (CAs) (Ali et al., 2013), Invasive Weed Optimization (IWO) (Mehrabian and Lucas, 2006), Imperialist Competitive Algorithm (ICA) (Atashpaz-Gargari and Lucas, 2007), Artificial Bee Colony (ABC) (Karaboga and Basturk, 2007), Firefly Algorithm (FFA) (Yang, 2010), Biogeography-Based Optimization (BBO) (Simon, 2008), Intelligent Water Drops (IWDs) (Hosseini, 2007), a hybrid Superior Solution Guided PSO with local search methods (Wu et al., 2014), a new and efficient optimizer (CFA) by Franklin's and Coulomb's laws theory (Ghasemi et al., 2018), Gravitational Search Algorithm (GSA) (Rashedi et al., 2009), Group Search Optimizer (GSO) (He et al., 2009), Wind Driven Optimization (WDO) algorithm (Bayraktar et al., 2013), Chemical Reaction Optimization (CRO) (Lam and Li, 2009), Hunting Search (HuS) algorithm (Oftadeh et al., 2010), Ant Colony Optimization (ACO) (Socha and Dorigo, 2008), Big Bang-Big Crunch (Erol and Eksin, 2006), Teaching–Learning-Based Optimization (TLBO) algorithm (Rao et al., 2011), Grey-Wolf Optimizer (GWO) (Mirjalili et al., 2014), States of Matter Search (SMS) (Cuevas et al., 2014), Group Counseling Optimizer (GCO) (Eita and Fahmy, 2010), Mine Blast Algorithm (MBA) (Sadollah et al., 2013), a nature-inspired meta-heuristic Water Wave Optimization (WWO) (Zheng, 2015), Joint Operations Algorithm (JOA) (Sun et al., 2016), Back-tracking Search Optimization (BSA) (Civicioglu, 2013), a modified new self-organizing hierarchical PSO with jumping time-varying acceleration coefficients (Ghasemi et al., 2020), Jaya (Rao, 2016), Soccer League Competition (SLC) (Moosavian and Roodsari, 2013), Bird Swarm Algorithm (BSA) (Meng et al., 2015), Cuckoo Optimization Algorithm (COA) (Rajabioun, 2011), Phasor Particle Swarm Optimization (PPSO) (Ghasemi et al., 2019), metaheuristic optimization techniques for optimal design of electrical machines (Hemmati and Rahideh, 2017; Hemmati et al., 2019), Gradient Evolution (GE) (Kuo and Zulvia, 2015), and Opposition-Based Algorithm (OBA) (Seif and Ahmadi, 2015).

Regarding the No Free Lunch theorem (NFL) (Wolpert and Macready, 1997), there is no meta-heuristic algorithm which is appropriate for all optimization problems. In other words, a particular meta-heuristic algorithm may yield favorable results for some problems, but it shows weak performance on some other problems. Therefore, in this paper, a new optimization method based on Turbulent Flow of Water-based Optimization (TFWO) is presented. This method has been inspired by the natural and random behavior of vortices in rivers, seas and oceans. We also investigate the application of the proposed algorithm in solving benchmark and real-world problems. The rest of the paper is organized as follows: The proposed Turbulent Flow of Water-based Optimization is introduced in Section 2. Section 3 presents and discusses the results on the benchmark test functions and examines the behavior of the proposed algorithm and compare it with other methods. In the second part of simulation study, in Section 4, the proposed TFWO algorithm is used for solving two real-world engineering problem, namely various types of nonlinear Economic Load Dispatch in power systems and Reliability–Redundancy Allocation Optimization for the overspeed protection system of a gas turbine. In Section 5 the conclusions for the proposed TFWO algorithm are given.

2. Turbulent Flow of Water-based Optimization (TFWO)

2.1. Introduction to turbulent flow of water: the whirlpool Theory

Turbulent flow of water through a narrow path in a circular form, for example around a submerged rock, may result in forming a whirlpool. This circulation and flow is affected by gravity force. As a result, flowing water moves downwards in a spiral path. As this continues, the speed of water flow is accelerated, and a small hole is created in the center which makes this flow even faster. And, when water is pulled into the hole (opening), a spinning motion starts, hence a whirlpool is created.

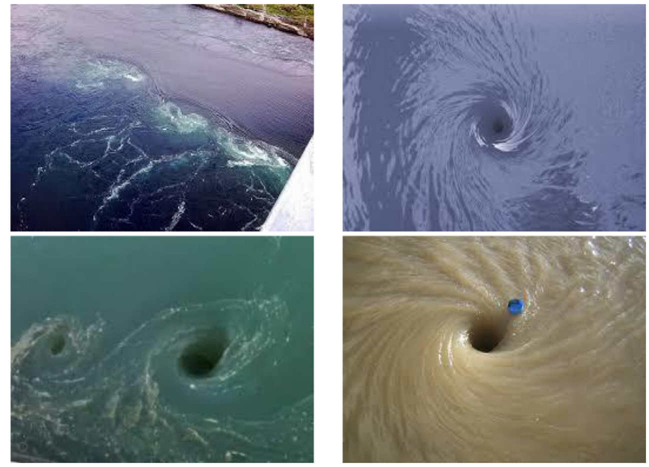

Fig. 1. Show generate several whirlpools with different forms and strengths in nature.

2.2. TFWO algorithm

Whirlpool is a random behavior of nature that occurs in rivers, seas, and oceans. In whirlpools, the center of whirlpool acts as a sucking hole, and pulls the objects and particles around it towards its middle and its inside, i.e. it applies centripetal force onto them. A whirlpool is in fact a volume of moving water mainly created by ocean tide. Whirlpool form in places where there are some small ridges besides each other on the surface of the streamlet. The flowing water collides with these ridges and then moves around itself. This way, the water in a narrow path moves around itself in the ridges and slowly is amalgamated around this circuit and forms a funnel. This flowing of water occurs due to centrifugal force. Most of whirlpools are not that much strong. Fig. 1 show several whirlpools with different forms and strengths. As seen, sometimes we have some whirlpools next to each other that have interactions in addition to their effects on their surrounding objects and particles.

2.2.1. Whirlpool model for optimization purposes

Here we use several whirlpools (or several groups of whirlpools with the center of the best member of the group as the hole of the whirlpool, which has the role of applying centripetal force on the other members), which absorb the objects and particles around them according to their distances and suck them and try to maintain them right in the center. This means that whirlpool apply centripetal force on all objects and particles (here the group members) around them. Centripetal force is defined as a force that is applied on a moving object in a circular path, and its direction is towards the center of the motion path of the object and perpendicular to it. Centrifugal force is an apparent force whose direction is the opposite of that of centripetal force. Centripetal force changes the moving direction of the object without changing its velocity. For a moving object around a circular path, there must be a centripetal force to counteract with it. The speedier the circular move, the greater centripetal force to maintain the object. Mathematical expression of centripetal force was first introduced by a Dutch physician, Christian Huygens, in 1659. For an object which moves in a circular path with constant velocity, the radius of the circle (r) is equal to multiplication of the mass of object (m) and squared velocity (v) divided by the centripetal force (F_c). In addition, the mass and centrifugal force, and also interaction between adjacent whirlpools and the objects cause resistance and deviation in this traction and suction. At first, we divide the algorithm's population into N_{Wh} groups or whirlpool sets, where the best member of each group represents the traction strength of that whirlpool. The best member of each group is set in the center of the corresponding whirlpool, and cause establishment of the hole

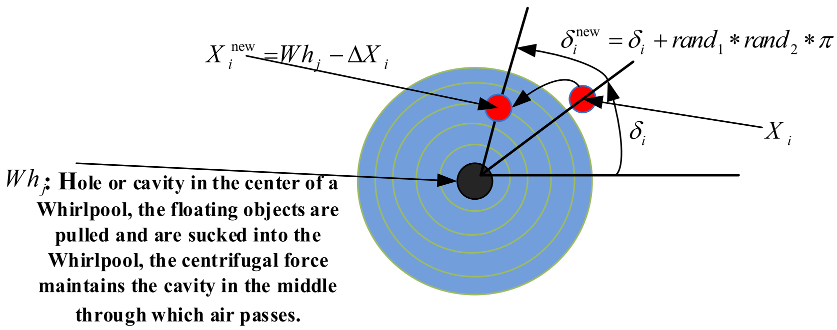

Fig. 2. The proposed model of whirlpool for optimization purposes.

and traction force in the center of the whirlpool. The best member of the group tries to unify the particle's positions with their whirlpool's centers (the position of the best member of the group), and pull them down into the hole. The whirlpools around the objects make a distance in the opposite direction of the object towards the whirlpool's hole (the best member of the group) up to a specific value, and somehow reduce the traction and centripetal forces and move it further away from the center of the whirlpool. The centrifugal force may cause an adverse effect on this centripetal force and/or break the objects or change them.

Moreover, bigger and stronger whirlpools pull or push the smaller and weaker ones according to the direction and distance, and effect on each other's positions, and change the positions of the other whirlpools.

In the following, the implementation and performance steps of this algorithm are presented and modeled based on the aforementioned paragraphs:

2.2.2. Formation of whirlpools:

At first, the initial population (X^0, comprising N_p members) of the algorithm is divided equally between N_{Wh} groups or whirlpool sets, and then the strongest member of each whirlpool set (the member with better objective function values f ()) is considered as the whirlpool which pulls the objects (X, including N_p- N_{Wh} objects).

2.2.3. The effects of whirlpools on objects of its set and other whirlpools:

Each whirlpool (*Wh*) acts as a sucking well or hole, and tends to unify the positions of objects in its set (X) with its central position by applying a centripetal force on them, and plunge them into its well. Therefore, jth whirlpool with its local position on Wh_j, acts in such a way that unifies the position of ith object (X_i) with that of itself, i.e. $X_i = Wh_j$. Nonetheless, other whirlpools cause some deviations (ΔX_i), according to distance between them (Wh-Wh_j) and the objective values (f ()). Hence, the new position of ith object would be equal to $X_i^{new} = Wh_j - \Delta X_i$, where the effects of the whirlpools on their objects are shown in Fig. 2.

As it is seen from Fig. 2, the objects (X) move with their special angle (δ) around their whirlpool's center and approach it. So, this angle at each moment (*iteration* in the algorithm) is changing:

$$\delta_i^{\text{new}} = \delta_i + rand_1 * rand_2 * \pi. \tag{1}$$

To model and calculate ΔX_i, the farthest and nearest whirlpools, i.e. the whirlpools with the most and the least weighed distance from all objects, are calculated based on Eq. (2), and then ΔX_i is calculated using Eq. (3). Eq. (4) is used to update the particle's position.

$$\Delta_t = f\left(Wh_t\right) * \left|Wh_t - sum\left(X_i\right)\right|^{0.5} \tag{2}$$

$$\begin{aligned} \Delta X_i = &\left(\cos\left(\delta_i^{\text{new}}\right) * rand\left(1, D\right) * \left(Wh_f - X_i\right)\right. \\ &- \sin\left(\delta_i^{\text{new}}\right) * rand\left(1, D\right) * \left.\left(Wh_w - X_i\right)\right) \\ &* \left(1 + \left|\cos\left(\delta_i^{\text{new}}\right) - \sin\left(\delta_i^{\text{new}}\right)\right|\right); \end{aligned} \tag{3}$$

$$X_i^{\text{new}} = Wh_j - \Delta X_i \tag{4}$$

where Wh_f and Wh_w are the whirlpools with minimum and maximum values of Δ_t, respectively; and δ_i is the ith object's angle.

The pseudo-codes 1 and 2 summarize the proposed model for updating object's position:

Pseudo-code 1:

for t = 1 : N_{Wh}
$\Delta_t = f\left(Wh_t\right) * \left|Wh_t - sum\left(X_i\right)\right|^{0.5}$
end
$Wh_f = Wh_t$ with minimum value of Δ_t
$Wh_w = Wh_t$ with maximum value of Δ_t

$\delta_i^{\text{new}} = \delta_i + rand_1 * rand_2 * \pi$
$\Delta X_i =$
$\left(\cos\left(\delta_i^{\text{new}}\right) * rand\left(1, D\right) * \left(Wh_f - X_i\right)\right.$
$\left.- \sin\left(\delta_i^{\text{new}}\right) * rand\left(1, D\right) * \left(Wh_w - X_i\right)\right)$
$* \left(1 + \left|\cos\left(\delta_i^{\text{new}}\right) * - \sin\left(\delta_i^{\text{new}}\right)\right|\right);$
$X_i^{\text{new}} = Wh_j - \Delta X_i;$

Pseudo-code2:

$X_i^{new} = \min\left(\max\left(X_i^{new}, X^{min}\right), X^{max}\right);$
$if\ f\left(X_i^{new}\right) <= f\left(X_i\right)$
$X_i = X_i^{new};$
$f\left(X_i\right) = f\left(X_i^{new}\right);$
end

2.2.4. Centrifugal force

Although the centripetal force pulls the moving objects towards their whirlpool, the centrifugal force takes them away from the corresponding center. Based on Newton's first law of motion, an object at rest stays at rest and an object in motion stays in motion with the same speed and in the same direction unless acted upon by an unbalanced force. Centrifugal force (or FE_i) sometimes overcomes the centripetal or traction force of the whirlpool, and randomly transfers the object to a new position. Here we have modeled the centrifugal force (as Eq. (5)), which occurs randomly in one dimension of the decision variables. To achieve this, at first the centrifugal force is calculated based on the angle between object and the whirlpool (as Eq. (5)), and if this force is greater than a random value (with uniform distribution in the range

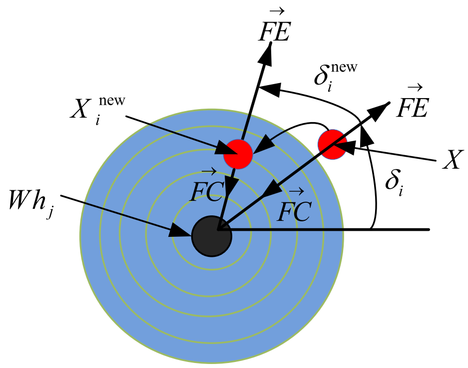

Fig. 3. The various types of forces in a whirlpool.

[0,1]), the centrifugal action is performed for the selected a randomly selected dimension, as Eq. (6).

$$FE_i = \left(\left(\cos\left(\delta_i^{new}\right)\right)^2 * \left(\sin\left(\delta_i^{new}\right)\right)^2\right)^2 \tag{5}$$

$$x_{i,p} = x_p^{min} + rand * \left(x_p^{max} - x_p^{min}\right) \tag{6}$$

The pseudo-code 3 summarize this phenomenon, which is also expressed schematically in Fig. 3.:

Pseudo-code 3:

$FE_i = \left(\left(\cos\left(\delta_i^{new}\right)\right)^2 * \left(\sin\left(\delta_i^{new}\right)\right)^2\right)^2.$
$if\ rand < FE_i$
$p = \mathrm{round}\,(1 + \mathrm{rand} * (D-1));$
$x_{i,p} = x_p^{min} + rand * \left(x_p^{max} - x_p^{min}\right);$
$f\left(X_i\right) = f\left(X_i^{new}\right);$
end

2.2.5. Interactions between the whirlpools

Similar to the effects of a whirlpool on the surrounding objects, they also interact with and displace each other. This effect has been modeled somehow same as the effects of whirlpools on the objects, where every whirlpool tends to pull other whirlpools and apply centripetal force on them and plunge them into their wells (i.e. unify the position of the considered whirlpool with its own position). In order to model this effect, at first the nearest whirlpool is calculated based on its objective function and the minimum amount of Eq. (7). Then, Eqs. (8) and (9) are used for updating whirlpool's position.

$$\Delta_t = f\left(Wh_t\right) * \left|Wh_t - sum\left(Wh_j\right)\right| \tag{7}$$

$$\Delta Wh_j = rand\,(1, D) * \left|\cos\left(\delta_j^{new}\right) + \sin\left(\delta_j^{new}\right)\right| * \left(Wh_f - Wh_j\right) \tag{8}$$

$$Wh_j^{new} = Wh_f - \Delta Wh_j \tag{9}$$

where δ_j is the value of the jth whirlpool hole's angle.

The pseudo-codes 4 and 5 summarize this phenomenon:

Pseudo-code 4:

for t = 1 : $N_{Wh} - \{j\}$
$\Delta_t = f\left(Wh_t\right) * \left|Wh_t - sum\left(Wh_j\right)\right|$
end
$Wh_f = Wh$ with minimum value of Δ_t

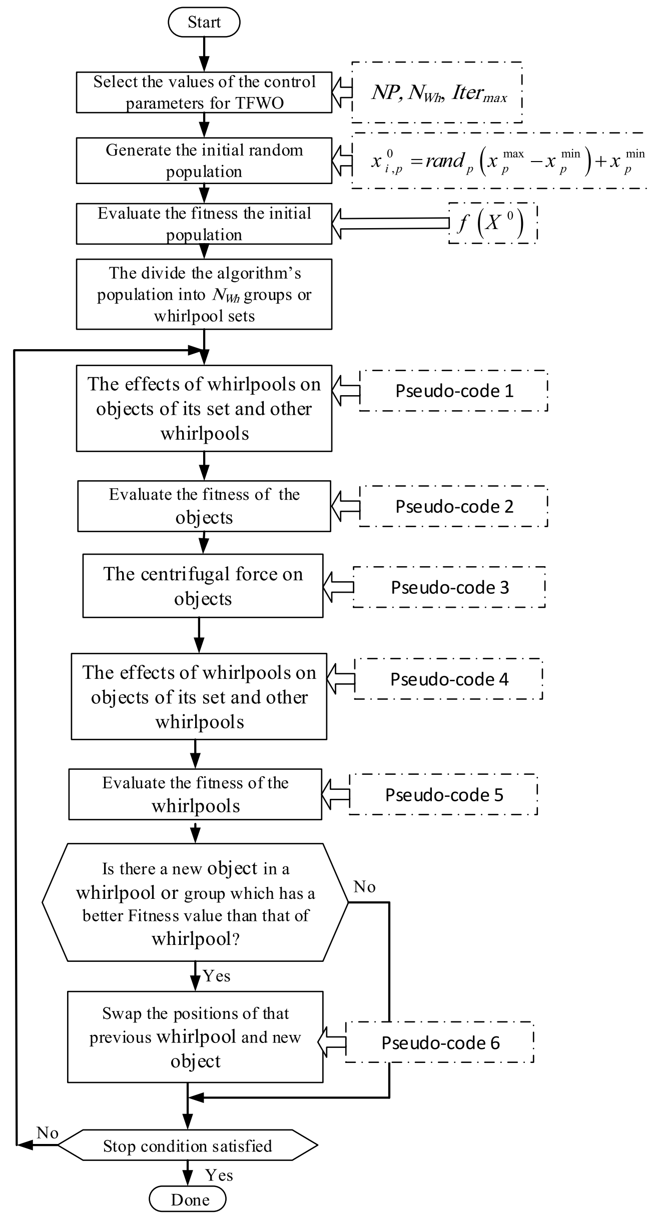

Fig. 4. Flowchart of the proposed **TFWO** optimization algorithm.

$Wh_j^{new} = Wh_f - \Delta Wh_j;$
$\Delta Wh_j = rand\,(1, D) * \left|\cos\left(\delta_j^{new}\right) + \sin\left(\delta_j^{new}\right)\right| * \left(Wh_f - Wh_j\right);$
$\delta_j^{new} = \delta_j + rand_1 * rand_2 * \pi.$

Pseudo-code 5:

$Wh_j^{new} = \min\left(\max\left(Wh_j^{new}, X^{min}\right), X^{max}\right);$
$if\ f\left(Wh_j^{new}\right) <= f\left(Wh_j\right)$
$Wh_j = Wh_j^{new};$
$f\left(Wh_j\right) = f\left(Wh_j^{new}\right);$
end

Finally, if the strongest member among the new members of the whirlpool's set has more strength (i.e. its value of objective function is less) than its corresponding whirlpool, then it is chosen as the new whirlpool for the next iteration. The pseudo-code 6 summarizes this phenomenon.

Pseudo-code 6:

$if\ f\left(X_{best}\right) <= f\left(Wh_j\right)$
$Wh_j \leftrightarrow X_{best}$
$f\left(Wh_j\right) \leftrightarrow f\left(X_{best}\right)$
end

Flowchart of the TFWO optimization method is shown in Fig. 4.

3. Simulation results

3.1. A competitive study

In the first section of this study, we compared the performance and effectiveness of the proposed TFWO algorithm with that of five powerful and efficient optimization algorithms listed below:

(1) SPSO (standard particle swarm optimization) (Zambrano-Bigiarini et al., 2013) and setting of control parameters as in developed model by Mahamed G.H. Omran and Maurice Clerc (http://www.particleswarm.info);
(2) ABC (artificial bee colony) (Karaboga and Basturk, 2007) and setting of control parameters as in (Karaboga and Basturk, 2007) (http://mf.erciyes.edu.tr/abc/);
(3) BBO (biogeography-based optimization) (Simon, 2008) and setting of control parameters as in (Simon, 2008) (http://academic.csuohio.edu/simond/bbo/);
(4) GWO (grey-wolf optimization) (Mirjalili et al., 2014) and setting of control parameters as in (Mirjalili et al., 2014).
(5) DE (selected model DE/rand/2) and setting of control parameters as in Mallipeddi et al. (2011),

The applied various benchmark test functions for global optimization are expressed in Table 1. The optimal objective function value for all benchmark test functions is $f_{\min} = 0$.

The optimization algorithms are implemented by MATLAB 2010 version 7.11(b) and the simulations are performed on a Pentium IV E5200, with 2 GB RAM. For each triple of benchmark test functions, optimization algorithms, and problem dimensions (*D*), 30 runs were carried out. The population size (*Np*) and the maximum number of iterations ($Iter_{max}$) is set to *Np* = 72 with $Iter_{max}$ = 3000 for D = 30 and *Np* = 210 with $Iter_{max}$ = 10000 for D = 100. The means (Mean) and standard deviations (Std) of the best solutions obtained for all the test functions with different dimensions were represented in Tables 2 and 3. ***Nb*** is the number of times the algorithm reaches the best value or rank 1 among all algorithms, ***Nw*** is the number of times the algorithm reaches the worst value or rank bottom-most among all algorithms, ***Mr*** is the average value the algorithm ranks for all test functions, and " + ", " − ", and " = " denote that the performance of the corresponding algorithm is better than, worse than, and similar to that of the proposed TFWO algorithm, respectively. At last, the final numbers of ***Nb/Nw/Mr*** and + / − / = are included as the overall final results of the algorithms.

The simulation results show that TFWO algorithm has much more optimization power for various benchmark test functions and for different dimensions. The TFWO is very competitive compared to SPSO, ABC, BBO, GWO and DE/rand/2 optimization algorithms for all various benchmark test functions selected here, especially for the real-parameter unimodal test functions such as f_{10}, f_{11}, f_{12} and f_{14} and real-parameter multi-modal test functions such as f_{17}, f_{18} and $f_{21}$for D = 30, and also the real-parameter unimodal test functions such as f_{10}, f_{11}, f_{12} and f_{13} and real-parameter multimodal test functions such as f_{15}, f_{16}, f_{17}, f_{18}, f_{19} and f_{21} for D = 100. However, for test functions f_{15}, f_{19} and f_{20} for D = 30, and f_{14} and f_{20} for D = 100, the proposed TFWO algorithm has not a better performance compared to other algorithms.

According to Tables 2 and 3, TFWO algorithm outperforms SPSO, ABC, BBO, GWO and DE/rand/2 optimization algorithms in seventeen, seventeen, eighteen, eighteen, and fifteen test functions for D = 30 and in twenty, nineteen, nineteen, eighteen, and seventeen test functions for D = 100. According to Tables 2 and 3, although the proposed TFWO algorithm can find the global optimal solutions for more functions, and also, the reported results in Tables 2 and 3 prove the superiority of the proposed TFWO algorithm over other successful SPSO, ABC, BBO, GWO and DE/rand/2 optimization algorithms in solving real-parameter uni-modal and multi-modal test functions and even to more traditional benchmark test functions such as f_2, f_3, f_4, f_6, f_7 and f_8 for both D = 30 and D = 100.

Convergence performance of all optimization algorithms for some selected test functions are shown in Fig. 5 (a–l) for D = 30 and Fig. 6 (a–d) for D = 100, respectively. It can be seen that the proposed TFWO algorithm has a suitable convergence compared to other algorithms for many of test functions.

3.2. Experimental setting of TFWO

In Table 4, a comparison was performed to set the number of whirlpools to optimize the real-parameter functions with D = 30 and similar conditions to the preceding section's simulation. It can be easily observed that the number of whirlpools in the range 3 to 6 is the proper choice for most of optimization functions. Moreover, this table shows that according to the type of optimization problem, it is possible to achieve a more appropriate optimal value by proper selection of the number of whirlpools.

4. Two the practical real-world examples:

4.1. TFWO algorithm for the different Economic Load Dispatch (ELD) problems

Operation and planning of power systems are of great importance (Nikoobakht et al., 2019, 2020). One of the problems regarding these subjects is economic load dispatch (ELD) problem. The main objective of ELD problem is to minimize the total operating costs of electrical power generation subject to various system and operational constraints. Summary of some optimization algorithms use for solving different ELD problems in the recent literature are such as Random Drift PSO (RDPSO) (Sun et al., 2013), Hybrid Hierarchical Evolution (HHE) (Pandit et al., 2014), distributed Auction-based Algorithm (AA) (Binetti et al., 2013), Chaotic Self-Adaptive Differential HS (CSADHS) (Rajagopalan et al., 2015), θ-Improved Cuckoo Optimization (θ-ICO) (Azizipanah-Abarghooee et al., 2014), Krill Herd Algorithm (KHA) (Mandal et al., 2014), Rank Cuckoo Search Algorithm (RCSA) (Nguyen and Vo, 2015), Crisscross Optimization Algorithm (COA) (Meng et al., 2015), immune algorithm with power redistribution (Aragón et al., 2015), hybrid cross-entropy method (Subathra et al., 2014), hybrid PSO and GSA (PSOGSA) (Radosavljević, 2016), a new improved DE (Gustavo et al. 2016), a Modified Symbiotic Organisms Search Algorithm (MSOSA) (Secui, 2016), Chaotic Bat Algorithm (CBA) (Adarsh et al., 2016), θ-Modified BA (θ-MBA) (Kavousi-Fard and Khosravi, 2016), Grey Wolf Optimization (GWO) (Pradhan et al., 2016), Exchange Market Algorithm (EMA) (Ghorbani and Babaei, 2016), Social Spider Algorithm (SSA) (Yu and Vok, 2016), Modified SSA (MSSA) (Elsayed et al., 2016), a new hybrid algorithm by DE and PSO for various types of multi-area economic dispatch problems (Ghasemi et al., 2016b), Hybrid GWO (HGWO) (Jayabarathi et al., 2016), Lightning Flash Algorithm (LFA) (Kheshti et al., 2017), PPSO (Gholamghasemi et al., 2019) and a hybrid of GRASP and DE algorithms (Neto et al., 2017).

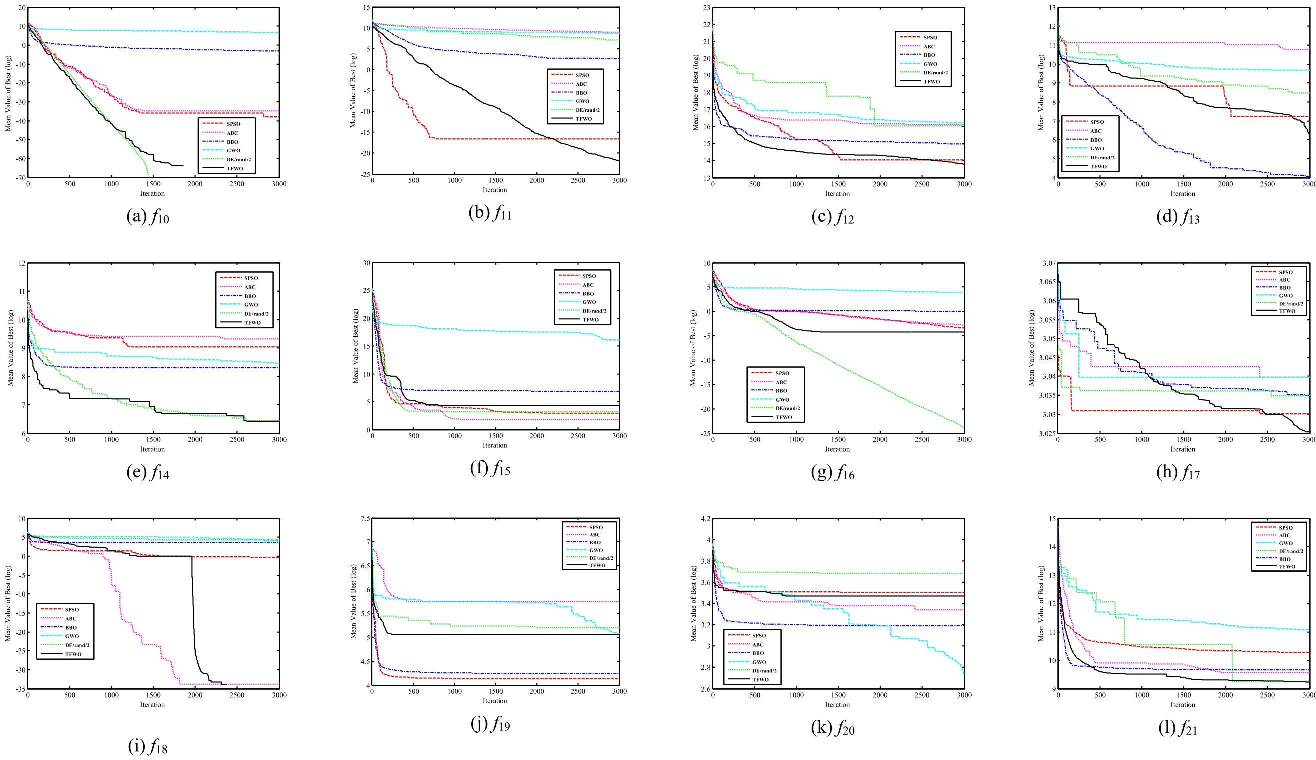

Fig. 5. Performance characteristics of algorithms for the real-parameter test functions with $D = 30$.

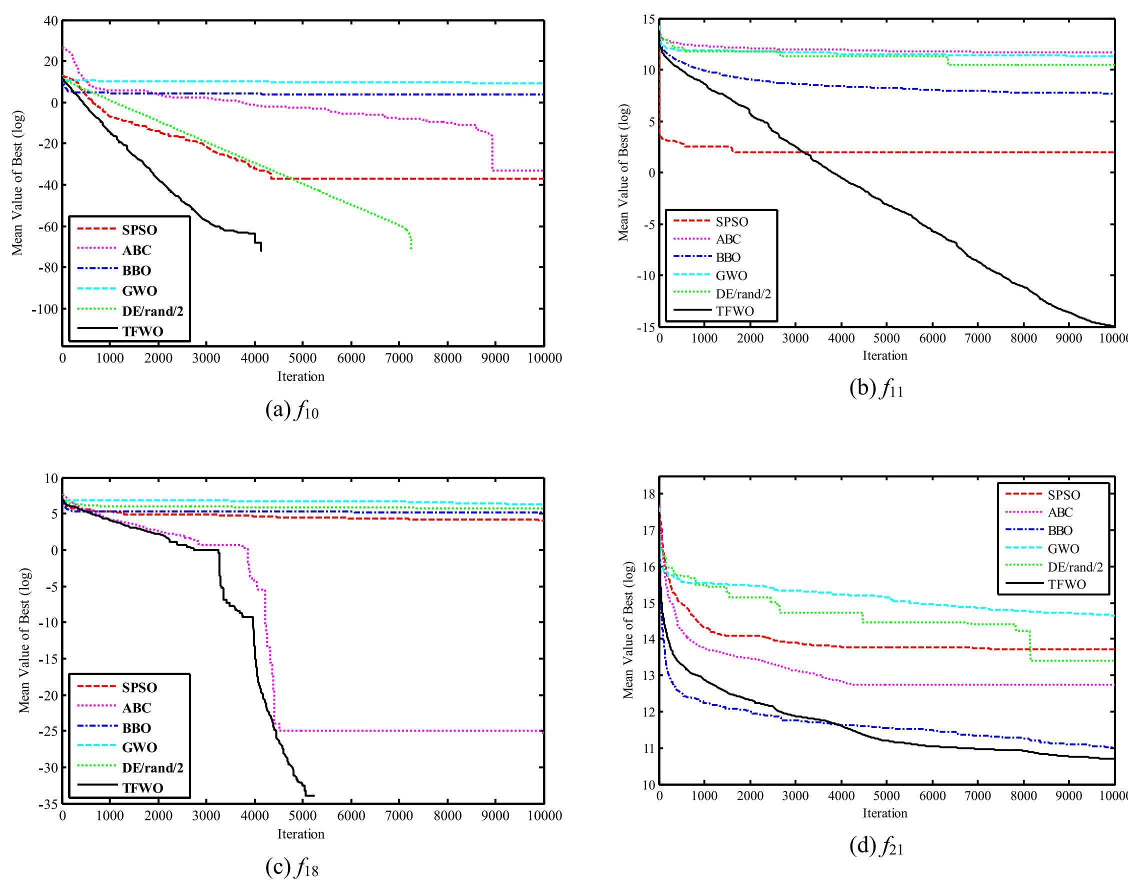

Fig. 6. Performance characteristics of algorithms for some of the real-parameter test functions with $D = 100$.

Table 1
Summary of the selected test functions for global optimization with $f_{\min} = 0$ (Chen et al., 2017; Suganthan et al., 2005).

Function name	Test function	Search range
Traditional benchmark test functions:		
Sphere	$f_1 = \sum_{i=1}^{D} x_i^2$.	$[-100, 100]^D$
Rosenbrock	$f_2 = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$.	$[-2.048, 2.048]^D$
Rastrigin	$f_3 = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$.	$[-5.12, 5.12]^D$
Noncontinuous Rastrigin	$f_4 = \sum_{i=1}^{D} (y_i^2 - 10\cos(2\pi y_i) + 10),\ y_i = \begin{cases} x_i, & \|x_i\| < 1/2, \\ round\,(2x_i), & \|x_i\| \geq 1/2, \end{cases}$.	$[-5.12, 5.12]^D$
Griewank	$f_5 = \frac{1}{4000} \sum_{i=1}^{D} \left(x_i - 100\right)^2 - \prod_{i=1}^{D} \cos\left(\frac{x_i - 100}{\sqrt{i}}\right) + 1$.	$[-600, 600]^D$
Penalized_1	$f_6 = \frac{\pi}{30} \left\{ 10\sin^2\left(\pi y_1\right) + \sum_{i=1}^{D-1} \left(y_i - 1\right)^2 . \left[1 + 10\sin^2\left(\pi y_i + 1\right)\right] + \left(y_D - 1\right)^2 \right\} + \sum_{i=1}^{D} u\left(x_i, \alpha, k, m\right)$. $u\left(x_i, \alpha, k, m\right) = \begin{cases} k\left(x_i - \alpha\right)^m, & x_i > \alpha, \\ 0, & -\alpha \leq x_i \leq \alpha, \\ k\left(-x_i - \alpha\right)^m, & x_i < -\alpha. \end{cases}$ Where $y_i = 1 + \left(x_i + 1\right)/4$, $a = 5$, $k = 100$ and $m = 4$.	$[-50, 50]^D$
Ackley	$f_7 = -20\exp(-0.2\sqrt{D^{-1}\sum_{i=1}^{D} x_i^2}) - \exp\left(D^{-1}\sum_{i-1}^{D} \cos\left(2\pi x_i\right)\right) + 20 + e$.	$[-32, 32]^D$
Weierstras	$f_8 = \sum_{i=1}^{D} \left(\sum_{k=0}^{k\max} \left[a^k \cos\left(2\pi b^k \left(x_i + 0.5\right)\right)\right]\right) - D\sum_{k=0}^{k\max} \left[a^k \cos\left(2\pi b^k\right)\right]$, $a = 0.5$ $b = 3$ k max $= 20$.	$[-0.5, 0.5]^D$
Schwefel_1_2	$f_9 = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j\right)^2$.	$[-100, 100]^D$
Real-parameter unimodal test functions:		
Shifted Sphere	$f_{10} = \sum_{i=1}^{D} z_i^2$, the o=[o_1, o_2, …, o_D] (the shifted global optimum) for all functions. And the $z = x - o$ for the all shifted functions and the $z = (x - o) * M$ for the all shifted Rotated functions.	$[-100, 100]^D$
Shifted Schwefel's Problem 1.2	$f_{11} = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} z_j\right)^2$.	$[-100, 100]^D$
Shifted Rotated High Conditioned Elliptic	$f_{12} = \sum_{i=1}^{D} \left(10^6\right)^{\frac{i-1}{D-1}} z_i^2$, the $z = (x - o) * M$ and the M: orthogonal matrix.	$[-100, 100]^D$
Shifted Schwefel's Problem 1.2 with Noise in Fitness	$f_{13} = \left(\sum_{i=1}^{D} \left(\sum_{j=1}^{i} z_j\right)^2\right) * (1 + 0.4\,\|N(0,1)\|)$.	$[-100, 100]^D$
Schwefel's Problem 2.6 with Global Optimum on Bounds	$f_{14} = \max\left\{\|A_i x - B_i\|\right\}$, the A is a D*D matrix, A_i is the ith row of A. And the $B_i = A_i$ *o. o is a D*1 vector, o_i are random number in the range [−100,100].	$[-100, 100]^D$
Real-parameter multimodal test functions:		
Shifted Rosenbrock's	$f_{15} = \sum_{i=1}^{D-1} (100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2)$, $z = x - o + 1$.	$[-100, 100]^D$
Shifted Rotated Griewank's without Bounds	$f_{16} = \frac{1}{4000} \sum_{i=1}^{D} \left(z_i\right)^2 - \prod_{i=1}^{D} \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1$, the $z = (x - o) * M$ and the $M = M' (1 + 0.3\,\|N(0,1)\|)$ and also M': linear transformation matrix, condition number = 3 orthogonal matrix.	$[-600, 600]^D$
Shifted Rotated Ackley's with Global Optimum on Bounds	$f_{17} = -20\exp(-0.2\sqrt{D^{-1}\sum_{i=1}^{D} z_i^2}) - \exp\left(D^{-1}\sum_{i-1}^{D} \cos\left(2\pi z_i\right)\right) + 20 + e$, the $z = (x - o) * M$.	$[-32, 32]^D$
Shifted Rastrigin's	$f_{18} = \sum_{i=1}^{D} (z_i^2 - 10\cos(2\pi z_i) + 10)$, the $z = x - o$.	$[-5, 5]^D$
Shifted Rotated Rastrigin's	$f_{19} = \sum_{i=1}^{D} (z_i^2 - 10\cos(2\pi z_i) + 10)$, the $z = (x - o) * M$.	$[-5, 5]^D$
Shifted Rotated Weierstrass	$f_{20} = \sum_{i=1}^{D} \left(\sum_{k=0}^{k\max} \left[a^k \cos\left(2\pi b^k \left(z_i + 0.5\right)\right)\right]\right) - D\sum_{k=0}^{k\max} \left[a^k \cos\left(2\pi b^k\right)\right]$, $a = 0.5$ $b = 3$ k max $= 20$.	$[-0.5, 0.5]^D$
Schwefel's Problem 2.1	$f_{21} = \sum_{i=1}^{D} \left(A_i - B_i(x)\right)^2$, $A_i = \sum_{j=1}^{D} \left(a_{ij}\sin\alpha_j + b_{ij}\cos\alpha_j\right)$, $B_i(x) = \sum_{j=1}^{D} \left(a_{ij}\sin x_j + b_{ij}\cos x_j\right)$, . a_{ij} , b_{ij} are integer random numbers in the range [−100,100], α are random numbers in the range [$-\pi$, π].	$[-\pi, \pi]^D$

4.1.1. Basic ELD problem

The objective function of ELD problem can be formulated in the following form (Aragón et al., 2015):

$$\text{Min } F_{\text{ELD1}} = \sum_{i=1}^{NG} F_{Ci}(P_{Gi}) = \sum_{i=1}^{NG} \left(a_i + b_i P_{Gi} + c_i P_{Gi}^2\right) \tag{10}$$

Subject to:

• ***Active power balance***: The total active power generated by all generators must meet the system load demand P_D and total transmission losses P_L. This constraint can be written as:

$$\sum_{i=1}^{NG} P_{Gi} = P_D + P_{Loss} \tag{11}$$

where transmission losses P_{Loss} can be calculated using loss coefficients B as follows (Aragón et al., 2015).

$$P_{Loss} = \sum_{i=1}^{NG} \sum_{j=1}^{NG} P_{Gi} B_{ij} P_{Gj} + \sum_{i=1}^{NG} B_{0i} + B_{00} \tag{12}$$

• ***Generation limits***:

$$P_{Gi,\min} \leq P_{Gi} \leq P_{Gi,\max} \tag{13}$$

• ***Ramp Rate Limits:*** The output of all online generators can only change with an acceptable ramp rate limit, i.e.:

$$P_{Gi} - P_{Gi}^0 \leq UR_{Gi} \quad \text{and} \quad P_{Gi}^0 - P_{Gi} \leq DR_{Gi} \tag{14}$$

where P_{Gi}^0 is the previous active power output of the ith generator (or is the old solution in algorithm) and DR_{Gi} and UR_{Gi} are down-ramp limit and up-ramp rate limit of the ith unit, respectively.

Table 2
Optimization results of algorithms over 30 independent runs on 23 test functions for $D = 30$ and $Iter_{\max} = 3000$.

f	SPSO Mean ± Std Rank, Winner	ABC Mean ± Std Rank, Winner	BBO Mean ± Std Rank, Winner	GWO Mean ± Std Rank, Winner	DE/rand/2 Mean ± Std Rank, Winner	TFWO Mean ± Std Rank
Traditional benchmark test functions:						
f_1	3.14e−011 ± 4.14e−012 5, –	6.48e−016 ± 8.79e−017 4, –	5.53e−002 ± 2.74e−002 6, –	**1.64e−241 ± 0.00e+000** **1,+**	4.57e−063 ± 5.52e−063 2, +	6.19e−040 ± 1.17e−039 3
f_2	**1.17e+000 ± 2.95e−001** **1,+**	7.67e+000 ± 5.44e+000 2, +	2.60e+001 ± 1.01e+000 5, –	2.62e+001 ± 8.15e−001 6, –	2.43e+001 ± 1.38e−001 4, –	2.21e+001 ± 5.75e−001 3
f_3	7.62e−002 ± 4.68e−003 4, –	1.42e−015 ± 3.18e−015 3, –	2.79e+001 ± 6.25e+000 5, –	7.92e−016 ± 2.45e−015 2, –	3.45e+001 ± 2.78e+000 6, –	**0.00e+000 ± 0.00e+000** **1**
f_4	2.63e+000 ± 4.10e+000 6, –	3.56e−016 ± 7.94e−016 2, –	1.76e+000 ± 5.71e−001 5, –	1.84e−015 ± 9.17e−016 3, –	1.71e+001 ± 1.96e+000 4, –	**0.00e+000 ± 0.00e+000** **1**
f_5	5.43e−003 ± 1.91e−002 3, +	1.79e−012 ± 6.50e−012 2, +	7.28e−002 ± 4.36e−002 5, –	3.81e+000 ± 2.98e+000 6, –	**0.00e+000 ± 0.00e+000** **1, +**	3.84e−002 ± 2.96e−002 4
f_6	1.10e−006 ± 6.47e−007 3, –	6.47e−005 ± 3.12e−006 5, –	6.12e−005 ± 1.67e−005 4, –	1.77e−002 ± 6.53e−002 6, –	2.57e−032 ± 0.00e+000 2, –	**2.44e−032 ± 2.45e−032** **1**
f_7	9.75e−005 ± 2.86e−005 5, –	5.24e−014 ± 1.05e−014 4, –	5.88e−002 ± 2.00e−002 6, –	1.25e−014 ± 2.53e−014 3, –	6.51e−015 ± 3.45e−015 2, –	**5.74e−015 ± 1.83e−015** **1**
f_8	8.46e−006 ± 3.00e−006 4, –	3.61e−015 ± 3.80e−015 2, –	4.40e−001 ± 3.75e−002 5, –	4.85e−015 ± 3.67e−015 3, –	**0.00e+000 ± 0.00e+000** **1,=**	**0.00e+000 ± 0.00e+000** **1**
f_9	6.00e−008 ± 1.68e−007 3, –	5.14e+003 ± 7.58e+002 6, –	1.78e+001 ± 1.48e+000 4, –	**2.97e−078 ± 7.15e−078** **1,+**	7.67e+002 ± 3.08e+002 5, –	1.57e−010 ± 3.00e−010 2
Nb/Nw/Mr	1/1/3.778	0/1/3.333	0/3/5.0	2/3/3.444	2/1/3.0	**5/0/1.889**
+/ − / =	**2/7/0**	**2/7/0**	**0/9/0**	**2/7/0**	**2/6/1**	–
Real-parameter unimodal (such as f_{10}, f_{11}, f_{12}, f_{13} and f_{14}) and multimodal (such as f_{15}, f_{16}, f_{17}, f_{18}, f_{19}, f_{20} and f_{21}) test functions:						
f_{10}	3.27e−017 ± 3.69e−017 2, –	8.27e−016 ± 9.80e−017 3, –	4.69e−002 ± 5.56e−003 4, –	8.43e+002 ± 6.64e+002 5, –	**0.00e+000 ± 0.00e+000** **1,=**	**0.00e+000 ± 0.00e+000** **1**
f_{11}	6.27e−008 ± 7.31e−008 2, –	7.66e+003 ± 3.42e+002 6, –	1.41e+001 ± 2.60e+000 3, –	6.25e+003 ± 2.90e+003 5, –	1.20e+003 ± 3.79e+002 4, –	**3.60e−010 ± 4.85e−010** **1**
f_{12}	1.26e+006 ± 2.94e+006 2, –	1.00e+007 ± 4.73e+006 5, –	3.17e+006 ± 1.14e+006 3, –	1.10e+007 ± 7.83e+006 6, –	9.13e+006 ± 2.42e+006 4, –	**9.93e+005 ± 4.79e+005** **1**
f_{13}	1.39e+003 ± 1.02e+003 3, –	4.72e+004 ± 5.32e+003 6, –	**5.74e+001 ± 2.00e+001** **1, +**	1.59e+004 ± 6.60e+003 5, –	4.83e+003 ± 2.25e+002 4, –	5.59e+002 ± 3.86e+002 2
f_{14}	8.38e+003 ± 1.32e+003 5, –	1.12e+004 ± 1.45e+003 6, –	4.05e+003 ± 2.98e+002 3, –	4.72e+003 ± 9.49e+002 4, –	6.25e+002 ± 3.09e+002 2, –	**6.14e+002 ± 1.95e+002** **1**
f_{15}	1.86e+001 ± 3.27e+002 2,+	**6.07e+000 ± 6.18e+000** **1,+**	9.62e+002 ± 1.94e+003 5, –	9.85e+006 ± 7.04e+006 6, –	2.42e+001 ± 4.66e−001 3, +	7.67e+001 ± 6.24e+001 4
f_{16}	3.26e−002 ± 6.93e−002 3, –	6.29e−002 ± 2.01e−002 4, –	9.91e−001 ± 7.64e−002 5, –	5.17e+001 ± 5.60e+001 6, –	**4.86e−011 ± 4.29e−011** **1,+**	1.42e−002 ± 1.04e−002 2
f_{17}	2.07e+001 ± 5.45e−002 2, –	2.09e+001 ± 3.69e−002 4, –	2.08e+001 ± 1.04e−001 3, –	2.09e+001 ± 3.69e−002 4, –	2.08e+001 ± 7.12e−002 3, –	**2.06e+001 ± 2.15e−001** **1**
f_{18}	7.67e−001 ± 2.59e−002 3, –	2.2e−015 ± 3.39e−015 2, –	3.81e+001 ± 4.75e+000 4, –	7.71e+001 ± 6.35e+000 6, –	6.02e+001 ± 1.14e+001 5, –	**0.00e+000 ± 0.00e+000** **1**
f_{19}	**6.24e+001 ± 3.00e+001** **1,+**	3.15e+002 ± 8.20e+001 6, –	6.97e+001 ± 2.14e+001 2, +	1.60e+002 ± 1.14e+002 4,	1.82e+002 ± 6.32e+000 5, –	1.58e+002 ± 3.58e+001 3
f_{20}	3.33e+001 ± 2.80e+001 5, –	2.82e+001 ± 1.23e+001 3, +	2.43e+001 ± 1.50e+000 2,+	**1.53e+001 ± 2.46e+000** **1,+**	3.99e+001 ± 1.79e+000 6, –	3.21e+001 ± 4.27e+000 4
f_{21}	2.90e+004 ± 9.49e+0043 5, –	1.42e+004 ± 4.85e+003 3, –	1.57e+004 ± 6.30e+003 4, –	6.45e+004 ± 3.50e+004 6, –	1.05e+004 ± 7.44e+003 2, –	**1.03e+004 ± 3.41e+003** **1**
Nb/Nw/Mr	1/**0**/2.917	1/5/4.083	1/**0**/3.250	1/6/4.834	2/1/3.333	**7/0/1.833**
+/ − / =	**2/10/0**	**2/10/0**	**3/9/0**	**1/11/0**	**2/9/1**	–
All						
Nb/Nw/Mr	2/1/3.286	1/6/3.762	1/3/4.0	3/9/4.238	4/2/3.19	**12/0/1.857**
+/ − / =	**2/17/0**	**4/17/0**	**3/18/0**	**3/18/0**	**4/15/2**	–

Eqs. (13) and (14) can be combined as follows:

$$\max\left\{P_{Gi,\min}, P_{Gi}^{0} - DR_{Gi}\right\} \leq P_{Gi} \leq \min\left\{P_{Gi,\max}, P_{Gi}^{0} + UR_{Gi}\right\} \tag{15}$$

4.1.2. Constraints related to valve-point effects

In practical ELD problems some generators may have Valve-Point Effects (VPE) which makes the problems non-smooth and non-convex. The cost function of generators with VPE can be formulated as follows (Ghasemi et al., 2016a):

$$\begin{aligned}\text{Min } F_{\text{ELD2}} = \sum_{i=1}^{NG} F_i(P_{Gi}) = \sum_{i=1}^{NG} a_i + b_i P_{Gi} + c_i P_{Gi}^2 \\ + \left|e_i \times \sin(f_i \times (P_{Gi,\min} - P_{Gi}))\right|\end{aligned} \tag{16}$$

4.1.3. Considering multi-fuel generators in ELD problem

The fuel cost function of each generator with Multi-Fuel (MF) must be presented with a piecewise function representing the effect of different fuel types (Ghasemi et al., 2016a). The cost function considering multi-fuel of ith unit is as follows (Ghasemi et al., 2016a):

$$F_i\left(P_{Gi}\right) = \begin{cases} a_{i1} + b_{i1}P_{Gi} + c_{i1}P_{Gi}^2, & fuel\ 1,\ P_{Gi,\min} \leq P_{Gi} \leq P_{Gi1} \\ a_{i2} + b_{i2}P_{Gi} + c_{i2}P_{Gi}^2, & fuel\ 2,\ P_{Gi1} \leq P_{Gi} \leq P_{Gi2} \\ \ldots \\ a_{ij} + b_{ij}P_{Gi} + c_{ij}P_{Gi}^2, & fuel\ j,\ \ P_{Gij-1} \leq P_{Gi} \leq P_{Gi,\max} \end{cases} \tag{17}$$

The fuel cost function of ith thermal unit with VPE and MF can be written as:

$$F_i(P_{Gi}) = \begin{cases} a_{i1} + b_{i1}P_{Gi} + c_{i1}P_{Gi}^2 + \left|e_{i1} \times \sin(f_{i1} \times (P_{Gi1,\min} - P_{Gi}))\right|, \\ fuel1,\ P_{Gi,\min} \leq P_{Gi} \leq P_{Gi1} \\ a_{i2} + b_{i2}P_{Gi} + c_{i2}P_{Gi}^2 + \left|e_{i2} \times \sin(f_{i2} \times (P_{Gi2,\min} - P_{Gi}))\right|, \\ fuel2,\ P_{Gi1} \leq P_{Gi} \leq P_{Gi2} \\ \ldots \\ a_{ij} + b_{ij}P_{Gi} + c_{ij}P_{Gi}^2 + \left|e_{ij} \times \sin(f_{ij} \times (P_{Gij,\min} - P_{Gi}))\right|, \\ fuel\ j,\ P_{Gij-1} \leq P_{Gi} \leq P_{Gi,\max} \end{cases} \tag{18}$$

4.1.4. Considering prohibited operating zones in ELD problem

Some thermal generators have Prohibited Operating Zones (POZ), on which the unit must not work (Ghasemi et al., 2016a). The POZ

Table 3
Optimization results of algorithms over 30 independent runs on 23 test functions for $D = 100$ and $Iter_{\max} = 10{,}000$.

f	SPSO Mean ± Std Rank, Winner	ABC Mean ± Std Rank, Winner	BBO Mean ± Std Rank, Winner	GWO Mean ± Std Rank, Winner	DE/rand/2 Mean ± Std Rank, Winner	TFWO Mean ± Std Rank
Traditional benchmark test functions:						
f_1	2.47e−017 ± 1.05e−017 4, –	4.09e−015 ± 3.10e−016 5, –	4.91e+001 ± 8.26e−001 6, –	**0.00e+000 ± 0.00e+000** 1, +	3.89e−039 ± 9.53e−040 3, –	1.08e−056 ± 2.11e−056 2
f_2	6.92e+001 ± 8.20e+000 3, –	8.19e+001 ± 7.62e+000 4, –	9.83e+001 ± 8.57e+000 6, –	9.20e+001 ± 4.89e−001 5, –	6.82e+001 ± 5.78e+000 2, –	**3.79e+001 ± 1.15e+001** **1**
f_3	5.19e−009 ± 3.25e−010 4, –	3.87e−013 ± 3.64e−013 3, –	8.07e+001 ± 2.99e+000 5, –	3.14e−015 ± 5.33e−015 2, –	4.30e+002 ± 3.69e+001 6, –	**2.56e−015 ± 8.11e−015** **1**
f_4	3.74e−003 ± 1.64e−003 4, –	2.03e−014 ± 1.55e−014 3, –	7.10e+001 ± 1.48e+001 5, –	8.88e−016 ± 9. 74e-016 2, –	2.15e+002 ± 7.40e+000 6, –	**0.00e+000 ± 0.00e+000** **1**
f_5	1.00e−003 ± 4.98e−003 3, +	8.26e−014 ± 1.20e−014 2, +	1.41e−001 ± 7.36e−002 5, –	5.30e+001 ± 6.61e+000 6, –	**0.00e+000 ± 0.00e+000** **1**, +	1.96e−002 ± 3.05e−002 4
f_6	9.50e−010 ± 2.47e−011 2, –	6.11e−009 ± 2.53e−010 3, –	1.36e−001 ± 5.10e−002 **4**, –	2.96e−001 ± 3.54e−002 **5**, –	**1.57e−032 ± 0.00e+000** **1**, =	**1.57e−032 ± 0.00e+000** **1**
f_7	3.72e−011 ± 2.00e−011 5, –	3.23e−013 ± 5.28e−014 4, –	1.58e+000 ± 7.39e−001 6, –	1.82e−014 ± 4.90e−015 3, –	1.63e−014 ± 4.10e−015 2, –	**1.60e−014 ± 3.57e−015** **1**
f_8	4.85e+000 ± 1.17e+000 5, –	5.68e−013 ± 9.37e−014 3, –	2.95e+000 ± 3.67e−001 4, –	1.43e−014 ± 1.64e−014 2, –	**0.00e+000 ± 0.00e+000** **1**, =	**0.00e+000 ± 0.00e+000** **1**
f_9	2.86e+000 ± 1.31e−001 3, –	8.14e+004 ± 9.86e+003 6, –	7.25e+001 ± 6.21e+000 4, –	**1.82e−115 ± 9.34e−115** **1**, +	3.15e+004 ± 4.67e+002 5, –	6.22e−007 ± 1.97e−007 2
Nb/Nw/Mr	0/1/3.667	0/1/3.667	0/3/5	2/2/3	3/2/3	**6/0/1.556**
+/ − / =	**1/8/0**	**1/8/0**	**0/9/0**	**2/7/0**	**1/6/2**	–
Real-parameter unimodal (such as f_{10}, f_{11}, f_{12}, f_{13} and f_{14}) and multimodal (such as f_{15}, f_{16}, f_{17}, f_{18}, f_{19}, f_{20} and f_{21}) test functions:						
f_{10}	8.14e−017 ± 9.92e−018 2, –	3.41e−015 ± 4.11e−016 3, –	4.73e+001 ± 2.25e+001 4, –	1.20e+004 ± 1.83e+003 5, –	**0.00e+000 ± 0.00e+000** **1**, =	**0.00e+000 ± 0.00e+000** **1**
f_{11}	7.32e+000 ± 1.41e−001 2, –	1.23e+005 ± 8.67e+003 6, –	2.20e+003 ± 1.83e+003 3, –	8.39e+004 ± 1.14e+004 5, –	3.57e+004 ± 1.70e+003 4, –	**3.19e−007 ± 9.14e−008** **1**
f_{12}	5.38e+006 ± 5.91e+005 2, –	8.14e+007 ± 1.89e+007 5, –	2.54e+007 ± 9.25e+006 4, –	2.38e+008 ± 2.52e+008 6, –	2.22e+007 ± 1.65e+006 3, –	**2.58e+006 ± 4.80e+005** **1**
f_{13}	8.72e+004 ± 1.64e+004 3, –	5.03e+005 ± 2.17e+004 6, –	6.66e+004 ± 1.41e+004 2, –	1.58e+005 ± 5.06e+004 4, –	4.31e+005 ± 8.23e+003 5, –	**5.49e+004 ± 6.45e+003** **1**
f_{14}	3.00e+004 ± 1.27e+003 5, –	5.51e+004 ± 2.86e+003 6, –	**7.13e+003 ± 3.85e+003** **1**, +	2.83e+004 ± 3.90e+004 4, –	7.24e+003 ± 1.68e+003 2, +	2.76e+004 ± 5.01e+003 3
f_{15}	1.15e+000 ± 6.06e−001 3, –	1.52e−002 ± 7.93e−003 2, –	1.84e+004 ± 7.96e+003 5, –	3.38e+009 ± 2.00e+009 6, –	8.89e+001 ± 2.40e−001 4, –	**2.09e−003 ± 2.63e−003** **1**
f_{16}	1.60e−002 ± 1.00e−002 2, –	1.28e−001 ± 1.73e−002 4, –	3.82e+000 ± 1.00e+000 5, –	5.14e+002 ± 1.74e+002 6, –	1.17e−001 ± 9.95e−003 3, –	**5.51e−003 ± 6.00e−003** **1**
f_{17}	2.10e+001 ± 1.70e−002 2, –	2.13e+001 ± 3.09e−002 4, –	2.13e+001 ± 9.60e−002 4, –	2.17e+001 ± 8.00e−003 5, –	2.12e+001 ± 2.00e−002 3, –	**2.03e+001 ± 1.08e−001** **1**
f_{18}	6.08e+001 ± 3.14e+000 3, –	7.98e−012 ± 9.00e−012 2, –	1.84e+002 ± 7.39e+001 4, –	5.38e+002 ± 4.76e+001 6, –	2.95e+002 ± 6.97e+000 5, –	**0.00e+000 ± 0.00e+000** **1**
f_{19}	4.94e+002 ± 3.16e+002 4, –	2.41e+003 ± 9.98e+001 6, –	4.85e+002 ± 2.23e+002 3, –	6.02e+002 ± 7.33e+001 5, –	4.77e+002 ± 1.12e+001 2, –	**4.26e+002 ± 1.15e+002** **1**
f_{20}	1.33e+002 ± 7.34e+000 4, –	1.31e+002 ± 1.13e+000 3, =	9.36e+001 ± 3.54e+001 2, +	**8.69e+001 ± 1.10e+001** **1**, +	1.36e+002 ± 5.25e+000 5, –	1.31e+002 ± 1.62e+001 3
f_{21}	9.15e+005 ± 8.49e+004 4, –	3.45e+005 ± 4.52e+004 3, –	6.17e+004 ± 3.24e+004 2, –	2.32e+006 ± 8.26e+003 6, –	6.69e+005 ± 1.05e+005 5, –	**4.40e+004 ± 1.09e+004** **1**
Nb/Nw/Mr	0/0/3.00	0/4/4.166	1/0/3.250	1/7/4.417	1/1/3.50	**10/0/1.333**
+/ − / =	**0/12/0**	**0/11/1**	**2/10/0**	**1/11/0**	**1/11/0**	–
All						
Nb/Nw/Mr	0/1/3.286	0/5/3.953	1/3/4.0	3/9/3.809	4/3/3.286	16/0/1.429
+/ − / =	**1/20/0**	**1/19/1**	**2/19/0**	**3/18/0**	**2/17/2**	–

constraints can be expressed as follows:

$$P_{Gi} \in \begin{cases} P_{Gi,\min} \le P_{Gi} \le P^{l}_{Gi1} \\ P^{u}_{Gik-1} \le P_{Gi} \le P^{l}_{Gik};\ k = 2, \ldots, n_i;\ \forall i \in \Omega \\ P^{u}_{Gin_i} \le P_{Gi} \le P_{Gi,\max} \end{cases} \tag{19}$$

In this section, the proposed TFWO algorithm has been applied to different types of convex and non-convex/non-smooth ELD problems of three benchmark power systems.

Case 1 ELD: The 10-unit benchmark power system with MF and VPE constraints

The input data for 10-unit benchmark power system was extracted from Sayah and Hamouda (2013) and the total power (load) demand is considered 2700 MW, with $Iter_{max} = 200$ and $Nwh = 3$, and the population size of $N = 40$ for TFWO algorithm.

Case 2 ELD: The 15-unit benchmark power system with transmission power losses considering both ramp rate limits and POZ

The total load demand is $P_D = 2630$ MW and other data of the system are given in Kavousi-Fard and Khosravi (2016), with $Iter_{max} = 200$ and $Nwh = 3$, and the population size of $N = 40$ for TFWO algorithm.

Case 3 ELD: The 80-unit benchmark power system with VPE constraint

The 80-unit large-scale benchmark power system is created by expanding a 40-unit power system (Niknam et al., 2011). The data of generators and the constraints of this system were extracted from Niknam et al. (2011). The real power (load) demand of the 80-unit benchmark power systems is assumed to be 21,000 MW, with $Iter_{max} = 1500$ and $Nwh = 3$, and the population size of $N = 80$ for TFWO algorithm.

Case 4 ELD: The 110-unit benchmark power system with ramp rate limits

This is a large-scale test system which is composed of 110 thermal units with quadratic fuel cost function characteristic which is a classical ELD problem. The total power demand $P_D = 15{,}000$ MW and the data for the thermal units are given in (Ghasemi et al., 2018). The real power (load) demand of the 110-unit benchmark power systems is assumed to be 15 000 MW, with $Iter_{max} = 2000$ and $Nwh = 3$, and the population size of $N = 80$ for TFWO algorithm.

The statistical results of the optimization algorithms including the minimum, the maximum, the mean and the standard deviation of

Table 4
Comparison of the different *Nwh* of **TFWO** over 30 independent runs on real-parameter test functions of for D=30 and $Iter_{max}$ = 3000.

f	$Nwh = 1$ Mean ± Std Winner	$Nwh = 3$ Mean ± Std Winner	$Nwh = 9$ Mean ± Std Winner	Initial ($Nwh = 6$) Mean ± Std
f_{10}	2.93e−028 ± 5.90e−028 –	**0.00e+000 ± 0.00e+000** =	0.00e+000 ± 0.00e+000 =	0.00e+000 ± 0.00e+000
f_{11}	2.33e−001 ± 4.64e−001 –	**9.84e−013 ± 1.37e−012** +	1.54e−008 ± 1.41e−008 –	3.60e−010 ± 4.85e−010
f_{12}	1.78e+006 ± 7.99e+005 –	**5.28e+005 ± 2.32e+005** +	1.46e+006 ± 7.94e+005 –	9.93e+005 ± 4.79e+005
f_{13}	9.06e+003 ± 1.85e+003 –	**1.86e+003 ± 2.67e+003** –	8.16e+002 ± 6.45e+002 –	5.59e+002 ± 3.86e+002
f_{14}	8.55e+003 ± 2.94e+003 –	**5.57e+003 ± 2.34e+003** –	7.53e+003 ± 1.78e+003 –	6.14e+002 ± 1.95e+002
f_{15}	7.33e+001 ± 4.43e+001 +	**2.41e+001 ± 3.69e+001** +	7.76e+001 ± 8.62e+001 –	7.67e+001 ± 6.24e+001
f_{16}	4.38e−002 ± 3.86e−002 –	**3.03e−002 ± 3.71e−002** –	1.60e−002 ± 1.10e−002 –	1.42e−002 ± 1.04e−002
f_{17}	2.04e+001 ± 1.89e−001 +	**2.05e+001 ± 3.9e−001** +	2.07e+001 ± 1.89e−001 –	2.06e+001 ± 2.15e−001
f_{18}	0.00e+000 ± 0.00e+000 =	**0.00e+000 ± 0.00e+000** =	1.29e−005 ± 3.16e−005 –	0.00e+000 ± 0.00e+000
f_{19}	4.21e+002 ± 6.72e+001 –	**2.22e+002 ± 7.30e+001** –	2.16e+002 ± 5.14e+001 –	1.58e+002 ± 3.58e+001
f_{20}	3.33e+001 ± 2.16e+000 –	**2.99e+001 ± 1.43e+000** +	3.06e+001 ± 3.19e+000 +	3.21e+001 ± 4.27e+000
f_{21}	7.30e+003 ± 6.66e+003 +	**4.59e+003 ± 3.30e+003** +	7.99e+003 ± 7.84e+003 +	1.03e+004 ± 3.41e+003
+/ − / =	3/8/1	**6/4/2**	2/9/1	–

Table 5
Comparison of results obtained by the algorithms for Case 1 ELD (in $/h).

Algorithm	Min. Cost	Max. Cost	Mean Cost	Std. Dev.
AA (Binetti et al., 2013)	623.9500	–	–	–
ORCSA (Nguyen and Vo, 2015)	623.8608	623.9353	623.8963	–
DEPSO (Sayah and Hamouda, 2013)	623.8300	624.0800	623.9000	–
SPPO (Singh et al., 2016)	623.8279	–	–	–
IPSO (Sun et al., 2013)	623.8730	–	–	–
TFWO	**623.8278**	**623.8417**	**623.8500**	**0.0098**

Table 6
Best statistical results of the algorithms for Case 2 ELD (in $/h).

Algorithm	Min. Cost	Mean cost	Max. Cost	Std. Dev.
IA_EDP (Aragón et al., 2015)	32,698.2018	32,750.2176	32,823.7790	29.2989
EMA (Ghorbani and Babaei, 2016)	32,704.4503	32,704.4504	32,704.4506	–
HGWO (Jayabarathi et al., 2016)	32,679.0000	32,685.0000	–	–
θ-MBA (Kavousi-Fard and Khosravi, 2016)	32,680.5956	32,687.3305	32,693.2640	–
TFWO	**32,577.3687**	**32,579.0940**	**32,581.2305**	**2.65e−003**

Table 7
Comparison of best results obtained by **TFWO** and the other algorithms for Case 3 ELD (in $/h).

Algorithm	Min. Cost	Mean Cost	Max. Cost	Std. Dev.
SSA-I (Elsayed et al., 2016)	243,173.5900	243,306.1050	243,490.8080	72.6630
MSSA (Elsayed et al., 2016)	242,909.2500	243,037.2500	243,229.9470	53.7620
KHA-III (Mandal et al., 2014)	242,836.7114	242,844.5534	242,853.6792	–
KHA-IV (Mandal et al., 2014)	242,825.2089	242,826.9347	242,828.1350	–
NPSO (Niknam et al., 2011)	242,844.1172	–	–	–
GWO (Pradhan et al., 2016)	242,825.4799	242,829.8192	242,837.1303	0.0930
CE-SQP (Subathra et al., 2014)	242,883.0400	243,945.2500	–	–
TFWO	**242,825.0587**	**242,827.0095**	**242,828.1074**	**1.0978**

the fuel cost objective function (Min. Cost, Max Cost, Mean Cost and Std. Dev., respectively) among the solutions obtained by 30 runs of

Table 8
Comparison of best results obtained by **TFWO** and the other algorithms for Case 4 ELD (in $/h).

Methods	Min. cost ($/h)	Mean cost ($/h)	Max. cost ($/h)	Std.
OIWO (Barisal and Prusty, 2015)	197,989.14	197,989.41	197,989.93	–
RQEA (Babu et al., 2008)	197,988.1793	197,988.1835	197,988.2006	–
SAB (Babu et al., 2002)	206,921.9057	207,764.7398	208,197.0059	–
SAF (Babu et al., 2002)	207,380.5164	207,813.3717	208,012.6248	–
BBO (Bhattacharjee et al., 2014)	198,241.1660	198,413.4500	199,102.5900	–
DE/BBO (Bhattacharjee et al., 2014)	198,231.0600	198,326.6600	198,828.5700	–
ORCCRO (Bhattacharjee et al., 2014)	198,016.2900	198,016.3200	198,016.8900	–
TFWO	197,988.1790	197988.1823	197988.1904	0.0068

Table 9
Best statistical results of the proposed **TFWO** for Case 1 ELD.

Units	TFWO	Fuel types
P_1 (MW)	218.5944	2
P_2 (MW)	211.7117	1
P_3 (MW)	280.6574	1
P_4 (MW)	239.6394	3
P_5 (MW)	279.9404	1
P_6 (MW)	239.6394	3
P_7 (MW)	287.7346	1
P_8 (MW)	239.6394	3
P_9 (MW)	426.6023	3
P_{10} (MW)	275.8410	1
Total cost ($/h)	623.8278	

Table 10
Best statistical solutions of **TFWO** for Case 2 ELD.

Units	TFWO
P_1 (MW)	455.0086
P_2 (MW)	419.9580
P_3 (MW)	130.0130
P_4 (MW)	130.0130
P_5 (MW)	268.8937
P_6 (MW)	460.0238
P_7 (MW)	430.0430
P_8 (MW)	59.9940
P_9 (MW)	24.9975
P_{10} (MW)	62.9618
P_{11} (MW)	79.9920
P_{12} (MW)	79.9920
P_{13} (MW)	24.9975
P_{14} (MW)	14.9985
P_{15} (MW)	14.9985
P_L (MW)	26.8850
Total cost ($/h)	32,577.3687

TFWO algorithm for 10-unit benchmark power system are compared to those from other previously proposed algorithms such as ORCSA (Nguyen and Vo, 2015), SPPO (Singh et al., 2016), DEPSO (Sayah and Hamouda, 2013), AA (Binetti et al., 2013) and IPSO (Sun et al., 2013) for case 1, θ-MBA (Kavousi-Fard and Khosravi, 2016), IA_EDP (Aragón et al., 2015), EMA (Ghorbani and Babaei, 2016) and HGWO (Jayabarathi et al., 2016) for case 2, MSSA (Elsayed et al., 2016), NPSO (Niknam et al., 2011), GWO (Pradhan et al., 2016), KHA-III (Mandal et al., 2014), KHA-IV (Mandal et al., 2014) and CE-SQP (Subathra et al., 2014) for case 3, the SAB and SAF from reference (Babu et al., 2002), RQEA (Babu et al., 2008), the ORCCRO, BBO and DE/BBO from reference (Bhattacharjee et al., 2014), OIWO (Barisal and Prusty, 2015) for case 4, respectively, and are summarized in Tables 5 to 8. The experimental results obtained by TFWO algorithm clearly show robustness, consistency and stability of TFWO algorithm in comparison with other reported methods regarding all four statistical indexes. Furthermore, the best solutions found by the proposed TFWO algorithm for the benchmark power systems are shown in Tables 9 to 12.

4.2. *Reliability–Redundancy Allocation Optimization (RRAO) for the Overspeed protection system of a gas turbine*

The RRAO for the Overspeed protection system of a gas turbine (Ghavidel et al., 2018) is selected as another real-world engineering problem to investigate the effectiveness of the proposed TFWO algorithm. The formulation and the data of this problem were extracted from Ghavidel et al. (2018). Since this is an additional test case, its formulation and data are not repeated here in order to avoid over-length paper.

Table 13 shows that the best solution for the Overspeed protection test system related to the proposed TFWO and the solutions reported by IA (Chen, 2006), EGHS (Zou et al., 2011), PSSO (Huang, 2015), AR-ICA (Afonso, Mariani, and dos Santos Coelho 2013), NAFSA (He et al., 2015) and GA-PSO (Sheikhalishahi et al., 2013). The results show that the TFWO algorithm is suitable and reliable for the maximization optimization problems.

5. Conclusion

This paper proposes the **Turbulent Flow of Water-based Optimization (TFWO)** algorithm for the first time. This algorithm is a new efficient optimization algorithm that is inspired by a random behavior of nature that occurs in rivers, seas, and oceans for solving a complex problem. In this paper, a novel efficient inspired algorithm based on Whirlpool theory has been proposed to discover the global solutions of different optimization problems, named TFWO. In order to discover essential features of TFWO, its results were compared with those of

Table 11
Best solutions obtained by **TFWO** for Case 3 ELD with cost 242,825.0587 $/h.

Units	Units power output (MW)									
1–10	110.7998	110.7999	97.4000	179.7331	87.8003	140.0000	259.5996	284.5997	284.5998	130.0000
11–20	94.0000	94.0000	214.7597	394.2792	394.2793	394.2793	489.2794	489.2794	511.2793	511.2794
21–30	523.2794	523.2793	523.2796	523.2796	523.2794	523.2795	10.0001	10.0001	10.0001	87.8001
31–40	190.0000	190.0000	190.0000	164.7998	194.0002	194.7929	110.0000	110.0000	110.0000	511.2794
41–50	110.8002	110.8003	97.3999	179.7331	87.8000	140.0000	259.5996	284.5996	284.5998	130.0001
51–60	94.0000	94.0000	214.7598	394.2793	394.2793	394.2793	489.2794	489.2794	511.2794	511.2794
61–70	523.2794	523.2796	523.2794	523.2794	523.2792	523.2794	110.0000	110.0000	110.0000	87.7999
71–80	190.0000	190.0000	190.0000	164.8000	200.0000	199.9998	110.0000	110.0000	110.0000	511.2793

Table 12
Best statistical solutions of **TFWO** for Case 4 ELD.

Units	Out-power	Units	Out-power	Units	Out-power	Units	Out-power
P_{G1} (MW)	2.4000	P_{G31} (MW)	10.0000	P_{G61} (MW)	45.0000	P_{G91} (MW)	57.1013
P_{G2} (MW)	2.4000	P_{G32} (MW)	20.0000	P_{G62} (MW)	45.0000	P_{G92} (MW)	100.0000
P_{G3} (MW)	2.4000	P_{G33} (MW)	80.0000	P_{G63} (MW)	185.0000	P_{G93} (MW)	440.0000
P_{G4} (MW)	2.4000	P_{G34} (MW)	250.0000	P_{G64} (MW)	185.0000	P_{G94} (MW)	500.0000
P_{G5} (MW)	2.4000	P_{G35} (MW)	360.0000	P_{G65} (MW)	185.0000	P_{G95} (MW)	600.0000
P_{G6} (MW)	4.0000	P_{G36} (MW)	400.0000	P_{G66} (MW)	185.0000	P_{G96} (MW)	471.4642
P_{G7} (MW)	4.0000	P_{G37} (MW)	40.0000	P_{G67} (MW)	70.0000	P_{G97} (MW)	3.6000
P_{G8} (MW)	4.0000	P_{G38} (MW)	70.0000	P_{G68} (MW)	70.0000	P_{G98} (MW)	3.6000
P_{G9} (MW)	4.0000	P_{G39} (MW)	100.0000	P_{G69} (MW)	70.0000	P_{G99} (MW)	4.4000
P_{G10} (MW)	64.3624	P_{G40} (MW)	120.0000	P_{G70} (MW)	360.0000	P_{G100} (MW)	4.4000
P_{G11} (MW)	62.1143	P_{G41} (MW)	157.0560	P_{G71} (MW)	400.0000	P_{G101} (MW)	10.0000
P_{G12} (MW)	36.2518	P_{G42} (MW)	220.0000	P_{G72} (MW)	400.0000	P_{G102} (MW)	10.0000
P_{G13} (MW)	56.5598	P_{G43} (MW)	440.0000	P_{G73} (MW)	105.3022	P_{G103} (MW)	20.0000
P_{G14} (MW)	25.0000	P_{G44} (MW)	560.0000	P_{G74} (MW)	191.5635	P_{G104} (MW)	20.0000
P_{G15} (MW)	25.0000	P_{G45} (MW)	660.0000	P_{G75} (MW)	90.0000	P_{G105} (MW)	40.0000
P_{G16} (MW)	25.0000	P_{G46} (MW)	616.2808	P_{G76} (MW)	50.0000	P_{G106} (MW)	40.0000
P_{G17} (MW)	155.0000	P_{G47} (MW)	5.4000	P_{G77} (MW)	160.0000	P_{G107} (MW)	50.0000
P_{G18} (MW)	155.0000	P_{G48} (MW)	5.4000	P_{G78} (MW)	295.6214	P_{G108} (MW)	30.0000
P_{G19} (MW)	155.0000	P_{G49} (MW)	8.4000	P_{G79} (MW)	175.1072	P_{G109} (MW)	40.0000
P_{G20} (MW)	155.0000	P_{G50} (MW)	8.4000	P_{G80} (MW)	97.9570	P_{G110} (MW)	20.0000
P_{G21} (MW)	68.9000	P_{G51} (MW)	8.4000	P_{G81} (MW)	10.0000		
P_{G22} (MW)	68.9000	P_{G52} (MW)	12.0000	P_{G82} (MW)	12.0000	$\sum P_i - P_D - P_L$	0.0000
P_{G23} (MW)	68.9000	P_{G53} (MW)	12.0000	P_{G83} (MW)	20.0000		
P_{G24} (MW)	350.0000	P_{G54} (MW)	12.0000	P_{G84} (MW)	200.0000		
P_{G25} (MW)	400.0000	P_{G55} (MW)	12.0000	P_{G85} (MW)	325.0000		
P_{G26} (MW)	400.0000	P_{G56} (MW)	25.2000	P_{G86} (MW)	440.0000	Total cost ($/h)	197,988.1790
P_{G27} (MW)	500.0000	P_{G57} (MW)	25.2000	P_{G87} (MW)	15.0663		
P_{G28} (MW)	500.0000	P_{G58} (MW)	35.0000	P_{G88} (MW)	24.3375		
P_{G29} (MW)	200.0000	P_{G59} (MW)	35.0000	P_{G89} (MW)	82.3855	Mean time (s)	46.22
P_{G30} (MW)	100.0000	P_{G60} (MW)	45.0000	P_{G90} (MW)	89.3686		

Table 13
Comparison of best results obtained by TFWO with some of previous reported results for the overspeed protection system.

Design variables	IA (Chen, 2006)	EGHS (Zou et al., 2011)	PSSO (Huang, 2015)	AR-ICA (Afonso et al., 2013)	NAFSA (He et al., 2015)	GA-PSO (Sheikhalishahi et al., 2013)	TFWO
r_1	0.903800	0.900925066	0.90166461	0.90148988	0.90160779120	0.901628	0.901614621
r_2	0.874992	0.851636929	0.88817296	0.85003526	0.84993077684	0.888230	0.849921181
r_3	0.919898	0.948079849	0.94821033	0.94812952	0.94814603278	0.948121	0.948141545
r_4	0.890609	0.887654500	0.84987084	0.88823833	0.88821809379	0.849921	0.88822282
n_1	5	5	5	5	5	5	5
n_2	5	6	5	5	6	5	6
n_3	5	4	4	4	4	4	4
n_4	5	5	6	5	6	6	5
Best ($f_5(r, n)$)	0.999942	0.99995463	0.99995467	0.999954673	0.99995467467	0.99995467	**0.9999546746767 80**
Slack (g_1)	50	55	–	55	55	55	55
Slack (g_2)	0.002152	0.00000105	–	0.00213782	4.5195e–07	0.000006	2.3180e–009
Slack (g_3)	28.803701	24.80188272	–	24.8018827	24.802	15.363463	24.80188
MPI (%)	21.853	9.847e–02	1.03e–02	3.699e–03	1.496e–05	1.03e–02	–
Mean	–	0.99993588	0.9999416669	0.99993804	0.99995075542	0.99995467	**0.9999546746767 75**
Worst	–	0.99985315	0.99986938	0.99982276	–	0.99995467	**0.9999546746767 62**
Std	–	2.2e–05	1.61e–5	0.00002204	4.43e–06	1.0e–16	**3.38e–24**

other proposed inspired algorithms as well as reported obtained results in the previous literature algorithm for different problems optimization. The simulation results show that the performance of TFWO is efficient and effective in finding the global optimum solution of many different kinds of purposefully arranged 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.

CRediT authorship contribution statement

Mojtaba Ghasemi: Conceptualization, Methodology, Software. **Iraj Faraji Davoudkhani:** Writing - original draft, Writing - review & editing. **Ebrahim Akbari:** Methodology, Writing - review & editing, Project administration. **Abolfazl Rahimnejad:** Visualization, Resources. **Sahand Ghavidel:** Resources, Validation. **Li Li:** Supervision.

References

Abbass, H.A., 2001. MBO: Marriage in honey bees optimization-A haplometrosis polygynous swarming approach. In: Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546), Vol. 1. IEEE, pp. 207–214.

Adarsh, B.R., Raghunathan, T., Jayabarathi, T., Yang, X.S., 2016. Economic dispatch using chaotic bat algorithm. Energy 96, 666–675.

Afonso, L.D., Mariani, V.C., dos Santos Coelho, L., 2013. Modified imperialist competitive algorithm based on attraction and repulsion concepts for reliability-redundancy optimization. Expert Syst. Appl. 40 (9), 3794–3802.

Ali, M.Z., Alkhatib, K., Tashtoush, Y., 2013. Cultural algorithms: Emerging social structures for the solution of complex optimization problems. Int. J. Artif. Intell. 11 (A13), 20–42.

Aragón, V.S., Esquivel, S.C., Coello, C.C., 2015. An immune algorithm with power redistribution for solving economic dispatch problems. Inform. Sci. 295, 609–632.

Atashpaz-Gargari, E., Lucas, C., 2007. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: 2007 IEEE Congress on Evolutionary Computation. Ieee, pp. 4661–4667.

Azizipanah-Abarghooee, R., Niknam, T., Zare, M., Gharibzadeh, M., 2014. Multi-objective short-term scheduling of thermoelectric power systems using a novel multi-objective θ-improved cuckoo optimisation algorithm. IET Gener. Transm. Distrib. 8 (5), 873–894.

Babu, G.S., Das, D.B., Patvardhan, C., 2002. Simulated annealing variants for the solution of economic load dispatch. J.-Inst. Eng. India El Electr. Eng. Div. 22, 2–229.

Babu, G.S., Das, D.B., Patvardhan, C., 2008. Real-parameter quantum evolutionary algorithm for economic load dispatch. IET Gener. Transm. Distrib. 2 (1), 22–31.

Barisal, A.K., Prusty, R.C., 2015. Large scale economic dispatch of power systems using oppositional invasive weed optimization. Appl. Soft Comput. 29, 122–137.

Bayraktar, Z., Komurcu, M., Bossard, J.A., Werner, D.H., 2013. The wind driven optimization technique and its application in electromagnetics. IEEE Trans. Antennas and Propagation 61 (5), 2745–2757.

Bhattacharjee, K., Bhattacharya, A., nee Dey, S.H., 2014. Oppositional real coded chemical reaction optimization for different economic dispatch problems. Int. J. Electr. Power Energy Syst. 55, 378–391.

Binetti, G., Davoudi, A., Naso, D., Turchiano, B., Lewis, F.L., 2013. A distributed auction-based algorithm for the nonconvex economic dispatch problem. IEEE Trans. Ind. Inf. 10 (2), 1124–1132.

Chen, T.C., 2006. IAs based approach for reliability redundancy allocation problems. Appl. Math. Comput. 182 (2), 1556–1567.

Chen, D., Zou, F., Lu, R., Wang, P., 2017. Learning backtracking search optimisation algorithm and its application. Inform. Sci. 376, 71–94.

Chu, S.C., Tsai, P.W., 2007. Computational intelligence based on the behavior of cats. Int. J. Innovative Comput. Inf. Control 3 (1), 163–173.

Civicioglu, P., 2013. Backtracking search optimization algorithm for numerical optimization problems. Appl. Math. Comput. 219 (15), 8121–8144.

Cuevas, E., Echavarría, A., Ramírez-Ortegón, M.A., 2014. An optimization algorithm inspired by the states of matter that improves the balance between exploration and exploitation. Appl. Intell. 40 (2), 256–272.

Dai, C., Zhu, Y., Chen, W., 2006. Seeker optimization algorithm. In: International Conference on Computational and Information Science. Springer, Berlin, Heidelberg, pp. 167–176.

Eberhart, R., Kennedy, J., 1995. A new optimizer using particle swarm theory. In: MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science. Ieee, pp. 39–43.

Eita, M.A., Fahmy, M.M., 2010. Group counseling optimization: a novel approach. In: Research and Development in Intelligent Systems XXVI. Springer, London, pp. 195–208.

Elsayed, W.T., Hegazy, Y.G., Bendary, F.M., El-Bages, M.S., 2016. Modified social spider algorithm for solving the economic dispatch problem. Eng. Sci. Technol. Int. J. 19 (4), 1672–1681.

Erol, O.K., Eksin, I., 2006. A new optimization method: big bang–big crunch. Adv. Eng. Softw. 37 (2), 106–111.

Ghasemi, M., Aghaei, J., Akbari, E., Ghavidel, S., Li, L., 2016b. A differential evolution particle swarm optimizer for various types of multi-area economic dispatch problems. Energy 107, 182–195.

Ghasemi, M., Akbari, E., Rahimnejad, A., Razavi, S.E., Ghavidel, S., Li, L., 2019. Phasor particle swarm optimization: a simple and efficient variant of PSO. Soft Comput. 23 (19), 9701–9718.

Ghasemi, M., Akbari, E., Zand, M., Hadipour, M., Ghavidel, S., Li, L., 2020. An efficient modified HPSO-TVAC-based dynamic economic dispatch of generating units. Electr. Power Compon. Syst. 47 (19–20), 1826–1840.

Ghasemi, M., Ghavidel, S., Aghaei, J., Akbari, E., Li, L., 2018. CFA optimizer: A new and powerful algorithm inspired by Franklin's and Coulomb's laws theory for solving the economic load dispatch problems. Int. Trans. Electr. Energy Syst. 28 (5), e2536.

Ghasemi, M., Taghizadeh, M., Ghavidel, S., Abbasian, A., 2016a. Colonial competitive differential evolution: an experimental study for optimal economic load dispatch. Appl. Soft Comput. 40, 342–363.

Ghavidel, S., Azizivahed, A., Li, L., 2018. A hybrid jaya algorithm for reliability–redundancy allocation problems. Eng. Optim. 50 (4), 698–715.

Gholamghasemi, M., Akbari, E., Asadpoor, M.B., Ghasemi, M., 2019. A new solution to the non-convex economic load dispatch problems using phasor particle swarm optimization. Appl. Soft Comput. 79, 111–124.

Ghorbani, N., Babaei, E., 2016. Exchange market algorithm for economic load dispatch. Int. J. Electr. Power Energy Syst. 75, 19–27.

Gong, W., Cai, Z., Liang, D., 2014. Adaptive ranking mutation operator based differential evolution for constrained optimization. IEEE Trans. Cybern. 45 (4), 716–727.

He, Q., Hu, X., Ren, H., Zhang, H., 2015. A novel artificial fish swarm algorithm for solving large-scale reliability–redundancy application problem. ISA Trans. 59, 105–113.

He, S., Wu, Q.H., Saunders, J.R., 2009. Group search optimizer: an optimization algorithm inspired by animal searching behavior. IEEE Trans. Evol. Comput. 13 (5), 973–990.

Hemmati, R., Rahideh, A., 2017. Optimal design of slotless tubular linear brushless PM machines using metaheuristic optimization techniques. J. Intell. Fuzzy Systems 32 (1), 351–362.

Hemmati, R., Wu, F., El-Refaie, A., 2019. Survey of insulation systems in electrical machines. In: 2019 IEEE International Electric Machines & Drives Conference (IEMDC). IEEE, pp. 2069–2076.

Hosseini, H.S., 2007. Problem solving by intelligent water drops. In: 2007 IEEE Congress on Evolutionary Computation. IEEE, pp. 3226–3231.

Huang, C.L., 2015. A particle-based simplified swarm optimization algorithm for reliability redundancy allocation problems. Reliab. Eng. Syst. Saf. 142, 221–230.

Jayabarathi, T., Raghunathan, T., Adarsh, B.R., Suganthan, P.N., 2016. Economic dispatch using hybrid grey wolf optimizer. Energy 111, 630–641.

Karaboga, D., Basturk, B., 2007. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J. Global Optim. 39 (3), 459–471.

Kavousi-Fard, A., Khosravi, A., 2016. An intelligent θ-modified bat algorithm to solve the non-convex economic dispatch problem considering practical constraints. Int. J. Electr. Power Energy Syst. 82, 189–196.

Kheshti, M., Kang, X., Bie, Z., Jiao, Z., Wang, X., 2017. An effective lightning flash algorithm solution to large scale non-convex economic dispatch with valve-point and multiple fuel options on generation units. Energy 129, 1–15.

Krishnanand, K.N., Ghose, D., 2009. Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. Swarm Intell. 3 (2), 87–124.

Kuo, R.J., Zulvia, F.E., 2015. The gradient evolution algorithm: A new metaheuristic. Inform. Sci. 316, 246–265.

Lam, A.Y., Li, V.O., 2009. Chemical-reaction-inspired metaheuristic for optimization. IEEE Trans. Evol. Comput. 14 (3), 381–399.

Lee, K.S., Geem, Z.W., 2005. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput. Methods Appl. Mech. Engrg. 194 (36–38), 3902–3933.

Mallipeddi, R., Suganthan, P.N., Pan, Q.K., Tasgetiren, M.F., 2011. Differential evolution algorithm with ensemble of parameters and mutation strategies. Appl. Soft Comput. 11 (2), 1679–1696.

Mandal, B., Roy, P.K., Mandal, S., 2014. Economic load dispatch using krill herd algorithm. Int. J. Electr. Power Energy Syst. 57, 1–10.

Mehrabian, A.R., Lucas, C., 2006. A novel numerical optimization algorithm inspired from weed colonization. Ecol. Inform. 1 (4), 355–366.

Meng, A., Hu, H., Yin, H., Peng, X., Guo, Z., 2015. Crisscross optimization algorithm for large-scale dynamic economic dispatch problem with valve-point effects. Energy 93, 2175–2190.

Mirjalili, S., Mirjalili, S.M., Lewis, A., 2014. Grey wolf optimizer. Adv. Eng. Softw. (69), 46–61.

Mitchell, M., 1999. An Introduction to Genetic Algorithms. MIT press.

Moosavian, N., Roodsari, B.K., 2013. Soccer league competition algorithm, a new method for solving systems of nonlinear equations. Int. J. Intell. Sci. 4 (01), 7.

Nelder, J.A., Mead, R., 1965. A simplex method for function minimization. Comput. J. 7 (4), 308–313.

Neto, J.X.V., Reynoso-Meza, G., Ruppel, T.H., Mariani, V.C., dos Santos Coelho, L., 2017. Solving non-smooth economic dispatch by a new combination of continuous GRASP algorithm and differential evolution. Int. J. Electr. Power Energy Syst. 84, 13–24.

Nguyen, T.T., Vo, D.N., 2015. The application of one rank cuckoo search algorithm for solving economic load dispatch problems. Appl. Soft Comput. 37, 763–773.

Niknam, T., Mojarrad, H.D., Meymand, H.Z., 2011. Non-smooth economic dispatch computation by fuzzy and self adaptive particle swarm optimization. Appl. Soft Comput. 11 (2), 2805–2817.
Nikoobakht, A., Aghaei, J., Fallahzadeh-Abarghouei, H., Hemmati, R., 2019. Flexible co-scheduling of integrated electrical and gas energy networks under continuous and discrete uncertainties. Energy 182, 201–210.
Nikoobakht, A., Aghaei, J., Massrur, H., Hemmati, R., 2020. Decentralized hybrid robust/stochastic expansion planning in coordinated transmission and active distribution networks for hosting large-scale wind energy. IET Gener. Transm. Distrib. 14 (5), 797–807.
Oftadeh, R., Mahjoob, M.J., Shariatpanahi, M., 2010. A novel meta-heuristic optimization algorithm inspired by group hunting of animals: Hunting search. Comput. Math. Appl. 60 (7), 2087–2098.
Omidvar, M.N., Li, X., Mei, Y., Yao, X., 2013. Cooperative co-evolution with differential grouping for large scale optimization. IEEE Trans. Evol. Comput. 18 (3), 378–393.
Pandit, M., Srivastava, L., Sharma, M., Dubey, H.M., Panigrahi, B.K., 2014. Large-scale multi-zone optimal power dispatch using hybrid hierarchical evolution technique. J. Eng. 2014 (3), 71–80.
Pradhan, M., Roy, P.K., Pal, T., 2016. Grey wolf optimization applied to economic load dispatch problems. Int. J. Electr. Power Energy Syst. 83, 325–334.
Radosavljević, J., 2016. A solution to the combined economic and emission dispatch using hybrid PSOGSA algorithm. Appl. Artif. Intell. 30 (5), 445–474.
Rajabioun, R., 2011. Cuckoo optimization algorithm. Appl. Soft Comput. 11 (8), 5508–5518.
Rajagopalan, A., Sengoden, V., Govindasamy, R., 2015. Solving economic load dispatch problems using chaotic self-adaptive differential harmony search algorithm. Int. Trans. Electr. Energy Syst. 25 (5), 845–858.
Rao, R., 2016. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Int. J. Ind. Eng. Comput. 7 (1), 19–34.
Rao, R.V., Savsani, V.J., Vakharia, D.P., 2011. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput. Aided Des. 43 (3), 303–315.
Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S., 2009. GSA: a gravitational search algorithm. Inf. Sci. 179 (13), 2232–2248.
Sadollah, A., Bahreininejad, A., Eskandar, H., Hamdi, M., 2013. Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. Appl. Soft Comput. 13 (5), 2592–2612.
Sayah, S., Hamouda, A., 2013. A hybrid differential evolution algorithm based on particle swarm optimization for nonconvex economic dispatch problems. Appl. Soft Comput. 13 (4), 1608–1619.
Secui, D.C., 2016. A modified symbiotic organisms search algorithm for large scale economic dispatch problem with valve-point effects. Energy 113, 366–384.
Seif, Z., Ahmadi, M.B., 2015. An opposition-based algorithm for function optimization. Eng. Appl. Artif. Intell. 37, 293–306.
Sheikhalishahi, M., Ebrahimipour, V., Shiri, H., Zaman, H., Jeihoonian, M., 2013. A hybrid GA–PSO approach for reliability optimization in redundancy allocation problem. Int. J. Adv. Manuf. Technol. 68 (1–4), 317–338.
Simon, D., 2008. Biogeography-based optimization. IEEE Trans. Evol. Comput. 12 (6), 702–713.
Singh, N.J., Dhillon, J.S., Kothari, D.P., 2016. Synergic predator–prey optimization for economic thermal power dispatch problem. Appl. Soft Comput. 43, 298–311.
Socha, K., Dorigo, M., 2008. Ant colony optimization for continuous domains. European J. Oper. Res. 185 (3), 1155–1173.
Storn, R., Price, K., 1997. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. 11 (4), 341–359.
Subathra, M.S.P., Selvan, S.E., Victoire, T.A.A., Christinal, A.H., Amato, U., 2014. A hybrid with cross-entropy method and sequential quadratic programming to solve economic load dispatch problem. IEEE Syst. J. 9 (3), 1031–1044.
Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.P., Auger, A., Tiwari, S., 2005. Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization. KanGAL report, 2005005, p. 2005.
Sun, J., Palade, V., Wu, X.J., Fang, W., Wang, Z., 2013. Solving the power economic dispatch problem with generator constraints by random drift particle swarm optimization. IEEE Trans. Ind. Inf. 10 (1), 222–232.
Sun, G., Zhao, R., Lan, Y., 2016. Joint operations algorithm for large-scale global optimization. Appl. Soft Comput. 38, 1025–1039.
Tizhoosh, H.R., 2005. Opposition-based learning: a new scheme for machine intelligence. In: International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), Vol. 1. IEEE, pp. 695–701.
Winfield, D.H., 1969. Function and Functional Optimization by Interpolation in Data Tables (Doctoral dissertation). Harvard University.
Wolpert, D.H., Macready, W.G., 1997. No free lunch theorems for optimization. IEEE Trans. Evol. Comput. 1 (1), 67–82.
Wu, G., Qiu, D., Yu, Y., Pedrycz, W., Ma, M., Li, H., 2014. Superior solution guided particle swarm optimization combined with local search techniques. Expert Syst. Appl. 41 (16), 7536–7548.
Yang, X.S., 2010. Firefly algorithm, stochastic test functions and design optimisation. arXiv preprint arXiv:1003.1409.
Zambrano-Bigiarini, M., Clerc, M., Rojas, R., 2013. Standard particle swarm optimisation 2011 at cec-2013: A baseline for future pso improvements. In: 2013 IEEE Congress on Evolutionary Computation. IEEE, pp. 2337–2344.
Zheng, Y.J., 2015. Water wave optimization: a new nature-inspired metaheuristic. Comput. Oper. Res. 55, 1–11.
Zou, D., Gao, L., Li, S., Wu, J., 2011. An effective global harmony search algorithm for reliability problems. Expert Syst. Appl. 38 (4), 4642–4648.