Journal of the Chinese Institute of Engineers

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tcie20

Improved Whale Optimization Algorithm applied to design PID plus second-order derivative controller for automatic voltage regulator system

Diab Mokeddem & Seyedali Mirjalili

To cite this article: Diab Mokeddem & Seyedali Mirjalili (2020): Improved Whale Optimization Algorithm applied to design PID plus second-order derivative controller for automatic voltage regulator system, Journal of the Chinese Institute of Engineers, DOI: 10.1080/02533839.2020.1771205

To link to this article: https://doi.org/10.1080/02533839.2020.1771205

Published online: 07 Jul 2020.

JOURNAL OF THE CHINESE INSTITUTE OF ENGINEERS
https://doi.org/10.1080/02533839.2020.1771205

Improved Whale Optimization Algorithm applied to design PID plus second-order derivative controller for automatic voltage regulator system

Diab Mokeddem[a] and Seyedali Mirjalili[b]

[a]Department of Electrical Engineering Faculty of Technology, University of Ferhat Abbas Setif-1, Setif, Algeria; [b]Centre for Artificial Intelligence Research and Optimisation, Torrens University Australia, Brisbane, Australia

ABSTRACT
The paper proposes an Improved Whale Optimization Algorithm (IWOA). Its performance is validated by solving 23 benchmark functions. Comparing the results of IWOA with well-known meta-heuristic algorithms shows its efficiency. Three non-parametric statistical tests, namely, Friedman, Friedman aligned and Quade tests are used to confirm the proposed algorithm's superiority. IWOA is employed to design the parameters of a controller, namely, PID plus second-order derivative (PIDD^2) for an automatic voltage regulator system (AVR). In fact, the proposed technique benefits from an evolutionary operator crossover to promote the diversity of solutions while maintaining a reasonable local search behavior. The results are compared with the results of similar algorithms including Particle Swarm Optimization, Genetic Algorithm, Teaching Learning-Based Optimization, Differential Evolution, Cuckoo Search algorithm and Artificial Bee Colony, demonstrating the advantages and the efficiency of the IWOA-PIDD^2 controller. Robustness analysis of the optimal design obtained is conducted by varying the time constants of the AVR system component. The results proved that the proposed technique reliably outperforms most of the current techniques.

ARTICLE HISTORY
Received 20 July 2018
Accepted 12 May 2020

KEYWORDS
Whale optimization algorithm; metaheuristic technique; arithmetic crossover; PIDD^2 controller; automatic voltage regulator

1. Introduction

In power system networks, the Automatic Voltage Regulator (AVR) is of great interest as it maintains the level of the terminal voltage within the excitation system and improves power quality. As the loads of power are changing continuously, the voltage levels vary too and make it hard to maintain equilibrium between active and reactive powers without control. As such, we create a closed loop with the AVR system and the controller to keep the voltage magnitude within the standard values. Hence, a good controller is necessary for the acceptable performance of the AVR. To control the AVR system, many structures are available, among them, the Proportional Integral Derivative (PID), which is the most often preferred controller, known for its design structure simplicity, reliability and performance robustness over a wide range of operating conditions. A new controller named PID plus second-order derivative (PIDD^2) for AVR system has been introduced by Sahib (2015). The three parameters of proportional, integral and derivative gains determine the design of the PID controller (Kiam, Chong, and Li 2005). Optimal values of the controller gains are obtained using population-based algorithms or optimization techniques. Zeng et al. (2015a) proposed the design of multivariable PID controllers using real-coded population-based extremal optimization. In addition, Zeng et al. (2019) present an adaptive optimized PID neural network for the optimal control issue of multivariable nonlinear control systems. Furthermore, Zeng et al. (2015b) designed a novel fractional-order PID controller using multi-objective extremal optimization (MOEO) with three objective functions. Gaing (2004) proposed Particle Swarm Optimization (PSO) algorithm to design a PID controller for AVR system and compared it with Genetic Algorithm (GA) performance. A PSO-based fuzzy logic controller has been designed by Ghoshal (2004). Also, Panda, Sahu, and Mohanty (2012) tuned PID controller for AVR system using PSO. Artificial Bee Colony (ABC) has been implemented by Gozde and Taplamacioglu (2011) for comparative performance analyses for AVR system. Sahu, Panda, and Rout (2013) optimized parallel 2-DOF PID controller by Differential Evolution (DE). PIDD^2 controller gains are adjusted by Raju, Saikia, and Sinha (2016) using Ant Lion Optimizer algorithm. Al Gizi et al. (2015) presented a combined GA, Sugeno fuzzy logic and radial basis function neural network approaches for AVR system. Teaching Learning-Based Optimization (TLBO) is used by Chatterjee and Mukherjee (2016) to tune classical PID controller for an AVR system. Bingul and Karahan (2018) proposed a novel performance measure to find optimal designs of PID controller using Cuckoo Search algorithm (CS) for AVR system. These novel swarm intelligence methods are population-inspired algorithms where individuals cooperate to create an intelligent behavior (Lee and Shih 2012; Mirjalili 2015).

The GA is a search approach inspired by natural reproduction. It is very effective in solving constraint and unconstraint optimization problems. Unfortunately, GA is known as time-consuming because of slow convergence, as it has a very high response time. Similar to GA, DE is a meta-heuristic capable of searching for global optimum using specialized mutation, crossover and selection operators. The disadvantage of DE algorithm consists mainly in premature convergence.

CONTACT Diab Mokeddem mokeddem_d@yahoo.fr Department of Electrical Engineering Faculty of Technology, University of Ferhat Abbas Setif-1, Setif, Algeria

Another intelligent PSO algorithm works on the principle of collective behavior of animals such as swarms of birds and schools of fish. It is applied in many areas of science and engineering due to fast speed convergence and simple concept; it relies just on position update and speed equations. Each particle gravitates toward the local best position ever visited by all particles in the search space and the whole swarm shares the local best positions of the particles as the global best position.

However, PSO algorithm has a large number of parameters to be tuned such as swarm size, position clamping, velocity clamping and topology of neighborhoods, making the results much reliant on the selected values and causing premature convergence to local optima. The same drawback is confronted with ABC algorithm; which imitates honey bee comportment; and suffers also from slow convergence during the searching phase because of the need to adjust too many parameters, such as the number of different bee groups and limit values.

The above algorithms are very dependent on the definition of many parameters, which has led to interest in algorithms that require no adjusting of parameters, such as TLBO. This technique introduces the influence of a teacher on the grades and results of learners in two major steps: (i) learning from teacher and (ii) learning from other learners. The TLBO's main disadvantage is that it converges to local optimal solution instead of the near global optimal one due to learners' forgetting aspect. Similarly, CS algorithm's main drawbacks are slow convergence and easy stagnation to local optimum. Hence, the need for a new efficient algorithm requiring no adjusting of parameters, with fast convergence to global solutions, which avoids falling into local optimums. These characteristics are achieved by Whale Optimization Algorithm (WOA).

WOA is a recent algorithm that mimics the intelligent hunting behavior of the humpback whales proposed by Mirjalili and Lewis. As opposed to other swarm intelligence algorithms that require adjustment of at least three parameters as in PSO (Eberhart and Shi 2001), GA (Eberhart and Shi 1998; Holland 1992), Artificial Fish Swarm Algorithm (Dorigo, Birattari, and Stutzle 2006) and five parameters to adjust in Ant Colony Optimization Algorithm (Tsai and Lin 2011), WOA has only two controlling parameters including maximum generation number and population size. Further, WOA has good properties of easy implementation and high flexibility. WOA is effective for solving constraint and unconstraint objective optimization problems and is very well suited for simple and composite test functions. It has the characteristic of effectively finding optimal solutions thanks to the capability of convergence in many search spaces (local and global search).

The hunting behavior of humpback whale is called bubble-net feeding method, which is the main inspiration of the WOA algorithm. While encircling prey during hunting, whales move in a spiral shape (9-shaped path) by creating distinctive bubbles from about 12 m down and swim up to the surface to start hunting.

This paper intends to introduce an Improved Whale Optimization Algorithm (IWOA) to a PIDD^2 for AVR system. The original version of this algorithm is proposed to solve a wide range of problems. However, it needs modification to solve specific problems as suggested by No Free Lunch theorem in optimization. This theory states that there is no optimization algorithm to solve all optimization problems. Therefore, there is room for improvement to solve a specific set of problems.

To do so, this work integrates a crossover operator to promote randomness and diversity of solutions in each iteration, while maintaining a reasonable local search in an attempt not to degrade the accuracy of the final solution. The performance of IWOA-PIDD^2 is compared with some PID controllers optimized by other meta-heuristics, including PSO, GA, DEA, TLBO, CS and ABC. In addition, robustness analyses of the IWOA-PIDD^2 controlled AVR system under parametric variation are done. The rest of the paper is organized as follows. Section 2 gives preliminaries and essential operators in the WOA algorithm. In Section 3, we describe the IWOA with an illustrative example. Section 4 provides computational experiments of benchmark functions with result analyses. The concept of AVR system and the design of PID controller with double derivative are elucidated in Section 5. Section 6 concludes the work and suggests future directions.

2. Mechanism of Whale Optimization Algorithm (WOA)

The WOA algorithm is a new nature-inspired algorithm developed by Mirjalili and Lewis in 2016. It mimics the hunting , social foraging behavior of humpback whales, their favorite food is small fish and krill herds close to the surface. The humpback whales foraging behavior is called the bubble-net feeding method, in which they create distinctive bubbles from about 12 m down in spiral shape around the prey, encircling prey and then swim up to the surface to start hunting them. The hunting behavior of the whale optimizer is described by three main steps: encircling prey, bubble-net attacking and search for prey.

2.1. *Phase of encircling*

Once the prey location is defined, WOA starts encircling them. The prey here is the best optimal solution found so far (Mirjalili and Lewis 2016). After finding the best candidate solution in each iteration, the other solutions are relocated in the neighborhood of the best search agent following the equations:

$$\overrightarrow{\boldsymbol{X}}(t+1) = \overrightarrow{\boldsymbol{X}}^*(t) - A \cdot \overrightarrow{\boldsymbol{D}}, \tag{1}$$

$$\overrightarrow{\boldsymbol{D}} = \left| C \cdot \overrightarrow{\boldsymbol{X}}^*(t) - \overrightarrow{\boldsymbol{X}}(t) \right|, \tag{2}$$

$$A = 2 \cdot a \cdot r - a, \tag{3}$$

$$C = 2 \cdot r, \tag{4}$$

where $\overrightarrow{\boldsymbol{X}}^*$, $\overrightarrow{\boldsymbol{X}}$ denotes the position vector of the best solution and the position vector of a solution, t shows the current iteration. The values of a are decreased over the course of iterations from 2 to 0 and r is a random value between 0 and 1.

2.2. *Phase of bubble-net attacking (Exploitation)*

Mainly obtained by shrinking encircling mechanism, mathematically defined by Equation (3) where the value of A is decreasing. As such, the value of A is set to be a random value in the interval [$-a$, a] through iterations. Between original position and position of the current best agent, we find the new position of A.

The WOA algorithm requires humpback whales to move toward the prey using a spiral updating position equation is given (Mirjalili and Lewis 2016):

$$\overrightarrow{\boldsymbol{X}}(t+1) = \overrightarrow{\boldsymbol{X}}^{*}(t) + e^{bl} \cdot \cos(2\pi l) \cdot \overrightarrow{\boldsymbol{D}}, \tag{5}$$

where $\overrightarrow{\boldsymbol{D}}^{'} = \left|\overrightarrow{\boldsymbol{X}}^{*}(t) - \overrightarrow{\boldsymbol{X}}(t)\right|$ is the distance between the prey (best solution) and the ith whale, l is a random value uniformly distributed in the range of [−1,1] and b is a constant that changes the spiral shape.

The two foraging mechanisms, shrinking encircling and spiral updating position discussed above are mathematically modeled as follows:

$$\vec{\boldsymbol{X}}(t+1) = \begin{cases} \vec{\boldsymbol{X}}^{*}(t) - A.\vec{\boldsymbol{D}} & \text{if } p < 0.5 \\ \vec{\boldsymbol{X}}^{*}(t) + e^{bl}\cos(2\pi l).\vec{\boldsymbol{D}} & \text{if } p > 0.5 \end{cases}, \tag{6}$$

where p is a random number in [0, 1] switching between these two mechanisms Equation (6) with an equal probability (50%) to update whale position.

2.3. *Phase of searching (Exploration)*

In WOA, solutions face abrupt changes when A takes the values >1 or <−1 in order to force the search agent to move far away from the target whale. It is used to make the transition between exploration and exploitation phase easier by decreasing A.

When the value of $|A| \geq 1$, the WOA algorithm performs a global search and emphasizes exploration. For $|A| < 1$ the algorithm performs a local search emphasizing exploitation.

The mathematical model of the exploration phase can be given as:

$$\overrightarrow{\boldsymbol{X}}(t+1) = \overrightarrow{\boldsymbol{X}}_{rand} - A \cdot \overrightarrow{\boldsymbol{D}''}, \tag{7}$$

$$\overrightarrow{\boldsymbol{D}}'' = \left|C \cdot \overrightarrow{\boldsymbol{X}}_{rand} - \overrightarrow{\boldsymbol{X}}(t)\right|, \tag{8}$$

where $\overrightarrow{\boldsymbol{X}}_{rand}$ is randomly selected from whales (agents) in the current iteration.

Since its appearance in 2016, WOA algorithm has been applied successfully to solve various optimization problems as for size optimization of skeletal structures and simulated annealing for feature selection by hybrid WOA (Mafarja and Mirjalili 2017). Although WOA algorithm performance is very efficient to solve many optimization problems it has faced difficulty to converge to global optima and had been trapped in near global optima in case of PID controller tuning for AVR system.

3. Improved Whale Optimization Algorithm (IWOA)

This section proposes an improved version of WOA to improve the accuracy of global solution and avoid near global solution stagnation using crossover operator.

3.1. *Arithmetic crossover*

GA is a powerful evolutionary stochastic search technique, which is inspired from biological processes in nature. It relies on the selection and the operators of crossover and mutation (Holland 1992).

In this work, the standard WOA algorithm is integrated with crossover operator from GA. The advantage of incorporating the crossover operator is to promote the exploratory behavior of search agents (whales). The crossover operator is involved in the production of new children (offsprings) by combining the genes of two parents (agents); to improve reproduction, producing better offspring.

In this study, we used an arithmetic crossover operator (Köksoy and Yalcinoz 2008) that creates new offspring from two parent chromosome using a linear combination and following the equations:

$$\begin{cases} Offspring1 = a * Parent1 + (1 - a) * Parent2 \\ Offspring2 = (1 - a) * Parent1 + a * Parent2 \end{cases}, \tag{9}$$

where a is the weight factor with a random value between 0 and 1 which controls individual dominance in reproduction. For each gene this operation is executed separately.

Consider two parents selected for crossover consisting of four floats genes for each:

$$\begin{matrix} Parent\ 1 : |0.2|1.4|0.2|7.5| \\ Parent\ 2 : |0.5|3.5|0.1|6.6| \end{matrix},$$

.

If $a = 0.6$, two offsprings are produced as follows:

$$\begin{matrix} Offspring\ 1 : |0.32|2.24|0.16|7.14| \\ Offspring\ 2 : |0.38|2.66|0.14|6.96| \end{matrix},$$

.

This method brings the merits of GA to the standard WOA algorithm (Mirjalili and Lewis 2016) for increasing the searchability and the diversity of the prey. To emphasize a global search, the crossover operator is applied after the end of each iteration.

By integrating arithmetic crossover operator into the WOA, the IWOA is summarized in Figure 1. The pseudocode of the IWOA algorithm begins with a random initialization of population. Then, it evaluates the fitness of each search agent to assign the best one and update other agent positions in the neighborhood of this best search agent, in the phase of encircling. This phase is followed by exploration and exploitation stages. Crossover is applied to emphasize global search by producing new offspring in the search space and re-calculate the fitness to update the current best solution. These procedures are repeated until the end of iterations.

```
Set the initial values of the population X_i (i = 1, 2,…, n), coefficient vectors A, C,
maximum number of iteration Max_iter and crossover probability Pc.
Set t = 0. {Counter initialization}.
for (i = 1 : i ≤ n) do
    Evaluate the fitness of each search agent
end for
Assign the best search agent X*
while (t < Max_iter)
    for (i = 1 : i ≤ n)do
        Update a, A, C, l, and p
        if (p < 0.5) then
          if (|A| < 1) then
            Update the position of the current search agent by the Equation (1)
          else if (|A| ≥ 1)
                Select a random search agent (X_rand)
                Update the position of the current search agent by the Equation (7)
          end if
        else if (p ≥ 0.5)
              Update the position of the current search by the Equation (5)
        end if
    end for
    Apply crossover by the Equation (9)
    Check if any search agent goes beyond the search space and amend it
    Calculate the fitness of each search agent
    Update X* if there is a better solution
    t = t+1
end while
Return X*
```

Figure 1. Pseudocode of the IWOA algorithm.

4. Computational experiments

4.1. *Benchmark functions*

The IWOA algorithm is tested on 23 well-known benchmark functions from literature (Yao, Liu, and Lin 1999) organized depending on variable dimensions into two groups: low- and high-dimension functions as listed in Table 1. Functions from F_1 to F_7 are unimodal test functions requiring only exploitation to converge to the unique global optimum. Multimodal functions from F_8 to F_{23} present many optima (local and global), emphasizing balance between exploitation and exploration to avoid local optima and reach the global one. Table 1 provides the function dimension *Dim*, the search range and the optimum f_{opt} of each test function.

For results verification, the IWOA algorithm is compared with a number of the best algorithms in the literature, such as genetic algorithm (GA) (Gaing 2004), particle swarm optimization (PSO) (Eberhart and Shi 2001), gravitational search algorithm (GSA) (Rashedi, Nezamabadi-pour, and Saryazdi 2009), differential evolution (DE) (Storn and Price 1997), Fast Evolutionary Programing (FEP) (Yao, Liu, and Lin 1999) and whale optimization algorithm (WOA) (Mirjalili and Lewis 2016). To judge their performance, algorithms are compared fairly under the same parameters: population size of 30, 500 maximum iterations and over 30 runs with independent population initializations each time. Table 2 reports the experimental results of average and standard deviation values for each algorithm for all test functions listed in Table 1. The algorithms convergence curves for each benchmark function are presented in Figure 2.

4.2. *Result analysis*

From the results in Table 2 and convergence plot in Figure 2, it is obvious that IWOA algorithm outperforms the other algorithms for the majority of test functions. For the unimodal functions from F_1 to F_7; IWOA algorithm obtains optimal values better than the results of WOA, PSO and GA with the smallest standard deviation and average values, it shows competitive results compared with WOA in F_5 and although outperformed by PSO in F_6 it is ranked the second. IWOA provides good exploitation hence its fast convergence to the optimal solution.

In case of multimodal functions under higher dimension as (F_8, F_9, F_{10}, F_{11}, F_{12}, F_{13}) statistical results from Table 2 illustrate that IWOA reached the optimum solution with fast convergence for functions F_8, F_9, F_{10}, F_{11} followed closely by WOA. Although PSO shows better convergence for F_{12} and F_{13} its performance is very poor for the other four functions. Meanwhile, IWOA is ranked the second for functions F_{12} and

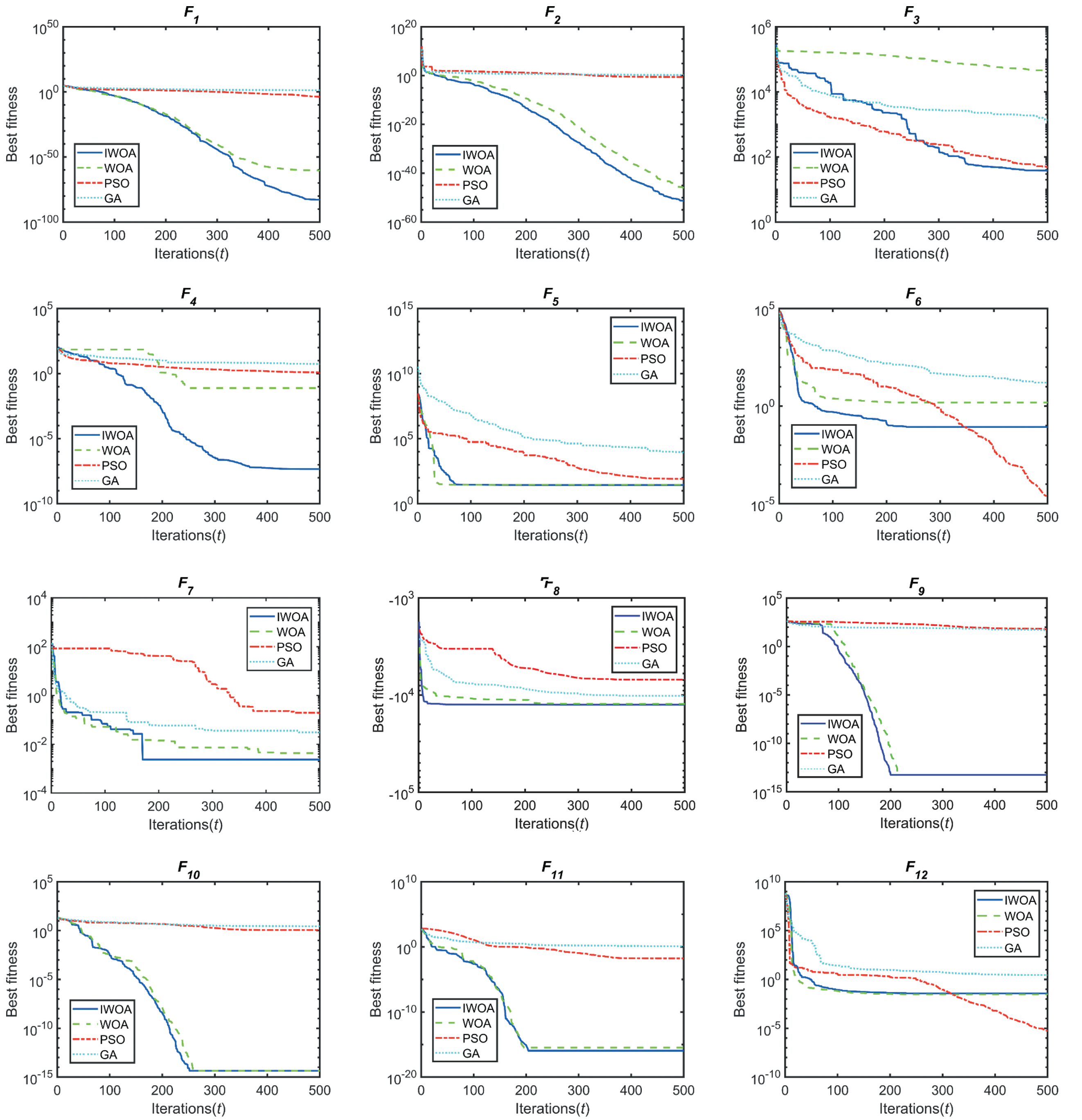

Figure 2. Best fitness curves of four algorithms for the defined benchmark functions.

F_{13} with approximate values for average and standard deviation compared with PSO.

For lower dimensions multimodal functions from F_{14} to F_{23}, a rapid convergence to the optimal solution is observed from the first iterations in all functions cases for IWOA and WOA. There was a close win for IWOA. For the curve F_{17}, all the algorithms show a similar behavior of convergence. The characteristic of exploration associated with whale optimization algorithm is beneficial for WOA and IWOA over GA, PSO, DE, GSA and FEP to find the global optimal solution.

Moreover, the superiority of the proposed IWOA algorithm over the competing methods is proved by means of non-parametric statistical tests: Friedman, Friedman aligned and Quade tests (Zeng et al. 2019; Li et al. 2016). In these tests, we compared the performance of the algorithms IWOA, FEP, WOA, GSA, PSO and GA, based on average error rate $\left(\left|f_{min} - f_{opt}\right|\right)$ over 23 benchmark test functions. Table 3 reports average Rank statistics and related p-value obtained by Friedman, Friedman aligned and Quade tests for IWOA and other comparative algorithms. The results show IWOA has the best average rank for all tests. Thus, the performance of IWOA is better than other algorithms. Analyzing this table we note that the algorithms IWOA, FEP, WOA, GSA, PSO and GA are statistically different with a significance level of $\alpha = 0.05$, as the p-values obtained by the tests: Friedman (7.6123×10^{-4}), Friedman aligned (2.4163×10^{-3}) and Quade (1.7989×10^{-5}) are smaller than α.

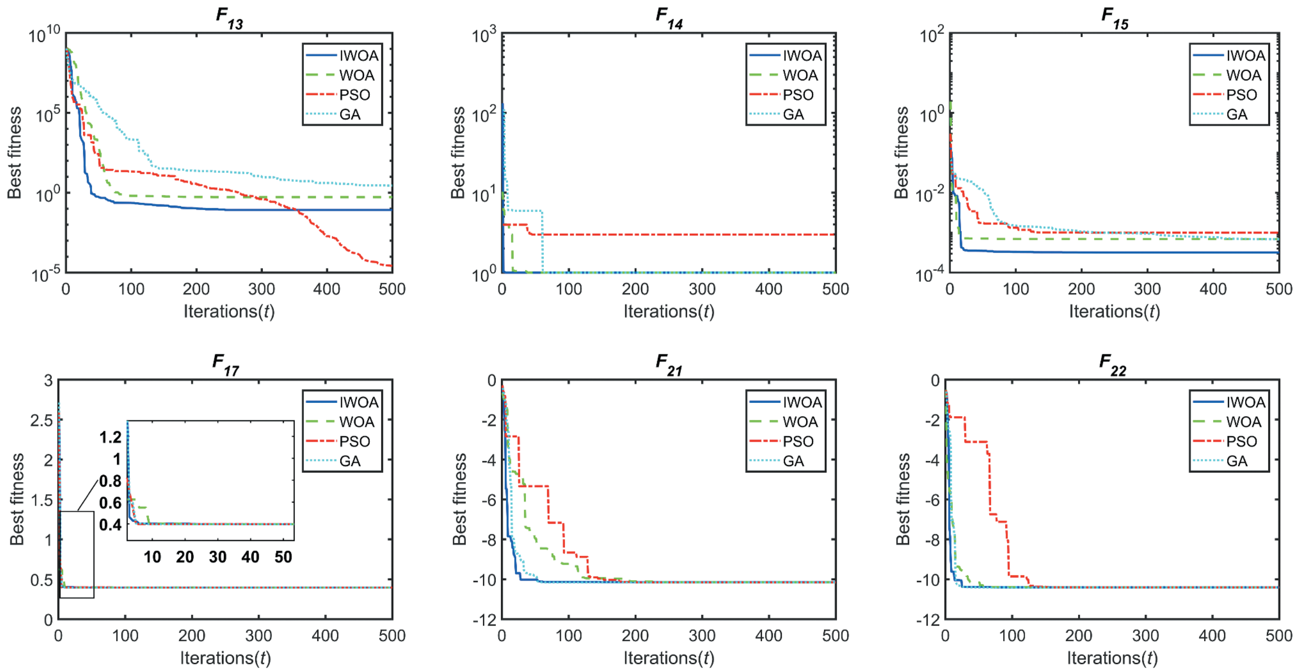

Figure 2. (Continued).

The IWOA is very advantageous thanks to fast convergence rate and good accuracy in computing the global optimum value and outperforms many meta-heuristic algorithms. In the next section, the performance of IWOA algorithm is tested on engineering control problem.

5. Application of IWOA on AVR system

5.1. Description of the AVR system

An AVR is used for synchronous generator to keep the terminal voltage at a specified level. The main components in AVR system are an amplifier, an exciter, a generator and a sensor (Wong and Li 2014; Panda, Sahu, and Mohanty 2012; Chatterjee and Mukherjee 2016). Figure 3 illustrates the closed loop of the AVR system with PIDD^2 controller. Dynamic performance of AVR system is analyzed by defining the transfer functions of its components as follows:

The Amplifier is modeled by the following transfer function:

$$G_a(s) = \frac{K_a}{1+\tau_a s}, \tag{11}$$

where K_a is the gain of the amplifier system having values in the range between [10, 40]; τ_a is the amplifier time constant with values in the range between [0.02 s, 0.1 s].

The Exciter is formulated as follows:

$$G_e(s) = \frac{K_e}{1+\tau_e s}, \tag{12}$$

where K_e and τ_e are, respectively, the gain and time constant of the exciter system. The values of K_e and τ_e are in the ranges between [1, 10] and [0.4 s, 1.0 s], respectively.

The Generator transfer function is given by:

$$G_g(s) = \frac{K_g}{1+\tau_g s}, \tag{13}$$

where K_g is the gain of generator having values in the range between [0.7, 1] and τ_g is the generator time constant with values in the range between [1 s, 2 s].

The Sensor can be defined by a simple first-order transfer function as it is illustrated in the following equation: τ_s

$$G_s(s) = \frac{K_s}{1+\tau_s s}, \tag{14}$$

with K_s is gain of sensor with values around 1.0 and τ_s is the sensor time constant varying from 0.001 to 0.06 s.

In this paper, the AVR system parameter values are considered as follows (Panda, Sahu, and Mohanty 2012): $K_a = 10.0, \tau_a = 0.1, K_e = 1.0, \tau_e = 0.4, K_g = 1.0, \tau_g = 1.0, K_s = 1.0, \tau_s = 0.01$.

The closed-loop transfer function is obtained by:

$$\frac{V_t(s)}{V_{ref}(s)} = \frac{0.1s + 10}{0.0004s^4 + 0.0454s^3 + 0.555s^2 + 1.51s + 11}. \tag{15}$$

The AVR system output response without any controller is shown in Figure 4. It is clear from the figure that the output response has many oscillations with peak amplitude of 1.5 V (overshoot of 65.4%), a rise time of 0.261 s, a steady-state at 0.909 and a settling time of 6.970 s. A PID controller is included to maintain the terminal voltage at the rage of 1.0 and enhance AVR's dynamic response.

5.2. PID controller with double derivative

In recent decades, PID controllers have been mostly adopted in industry to solve process control applications

Table 1. Definition of high- and low-dimensional benchmark functions.

Function	Dim	Bounds	f_{opt}
High-dimensional functions			
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	[−100,100]	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[−100,100]	0
$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j\right)^2$	30	[−100,100]	0
$F_4(x) = max_i\{\lvert x_i \rvert\}, 1 \leq i \leq n$	30	[−100,100]	0
$F_5(x) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$	30	[−100,100]	0
$F_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	[−100,100]	0
$F_7(x) = \sum_{i=1}^{n} ix_i^4 + r_{and}$	30	[−100,100]	0
$F_8(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	30	[−500,500]	−12569.5
$F_9(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	30	[−5.12,5.12]	0
$F_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} - \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + e$	30	[−32,32]	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[−600,600]	0
$F_{12}(x) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2\left[1 + 10\sin^2(\pi y_{i+1})\right] + (y_n - 1)^2\right\}$ $+ \sum_{i=1}^{n} u(x_i, 10, 100, 4) y_i = 1 + \frac{x+1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m x_i a \\ 0 - a x_i a \\ k(-x_i - a)^m x_i - a \end{cases}$	30	[−50,50]	0
$F_{13}(x) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n} (x_i - 1)^2\left[1 + \sin^2(3\pi x_i + 1)\right]\right.$ $\left. + (x_n - 1)^2\left[1 + \sin^2(2\pi x_n)\right]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	[−50,50]	0
Low-dimensional functions			
$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} \left(x_i - a_{ij}\right)^6}\right) - 1$	2	[−65,65]	1
$F_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_1 x_2}{b_i^2 + b_1 x_3 + x_4}\right]^2$	4	[−5,5]	0.0003
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	[−5,5]	−1.0316
$F_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	2	$x_1 \in [-5,0]$ $x_2 \in [10,15]$	0.398
$F_{18}(x) = \left[1 + (x_1 + x_2 + 1)^2\left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)\right]$ $\times \left[30 + (2x_1 - 3x_2)^2\left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)\right]$	2	[−2,2]	3
$F_{19}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	3	[1,3]	−3.86
$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	6	[0,1]	−3.32
$F_{21}(x) = \sum_{i=1}^{4} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	[0,10]	−10.1532
$F_{22}(x) = \sum_{i=1}^{7} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	[0,10]	−10.1532
$F_{23}(x) = \sum_{i=1}^{10} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	[0,10]	−10.1532

problems as they provide robust performance, simple design and easy implementation. Many modifications have occurred on PID controller's design, among them a Proportional – Integral – Derivative with Double Derivative ($PIDD^2$) controller. The addition of double derivate gain improves stability and speed response of the system. This latter is increased thanks to decrease in integral gain and increase in proportional gain. This is why there is interest in controlling AVR system by $PIDD^2$. This controller structure is formulated by the transfer function in Equation (16):

$$C_{PIDD^2} = K_p + \frac{K_i}{s} + K_d s + K_{dd} s^2, \qquad (16)$$

where K_p indicates proportional gain, K_i shows integral gain, K_d is derivative gain and K_{dd} is double derivative gains of ($PIDD^2$) controller. Double derivative controllers offer suitable dynamic responses and converge faster than single derivative controllers. For a good step response, the proposed controller parameters are tuned using the proposed algorithm, IWOA, providing minimization in the performance criteria's time domain. The performance criteria include integral of time multiplied absolute error overshoot M_p and settling time t_s is given by:

$$J = 0.8 \int_{0}^{t_{sim}} t\lvert e(t)\rvert dt + 0.1 t_s + 1.1 M_p, \qquad (17)$$

where t_{sim} is the simulation time.

Table 2. Performance results of benchmark functions.

	IWOA		WOA		PSO		GA	
F	*ave*	*std*	*ave*	*std*	*ave*	*std*	*ave*	*std*
F_1	8.764×10^{-74}	4.566×10^{-73}	1.439×10^{-62}	7.875×10^{-62}	1.067×10^{-4}	9.771×10^{-5}	20.0207	7.0759
F_2	5.363×10^{-50}	1.615×10^{-49}	1.047×10^{-45}	5.487×10^{-45}	0.0474	0.0931	1.8535	0.2692
F_3	84.5686	94.5020	3.985×10^{4}	1.122×10^{4}	88.5403	38.2349	1276.5722	372.3204
F_4	1.491×10^{-5}	3.231×10^{-5}	44.04881	25.1640	83.8055	26.3379	5.4551	1.2696
F_5	27.8014	0.5493	28.4461	0.3247	71.2343	29.3469	13243.9700	8335.6949
F_6	0.3850	0.1813	1.0388	0.3630	3.277×10^{-4}	8.992×10^{-4}	19.7975	7.0308
F_7	0.0033	0.0036	0.0056	0.0071	0.1618	0.0599	0.0326	0.0105
F_8	−9214.8	1604.1	−8920.4	1453.3	−5029.2848	1447.5463	−10208.709	568.8203
F_9	5.684×10^{-14}	0	1.894×10^{-15}	1.037×10^{-14}	56.6132	12.9359	49.7769	9.9089
F_{10}	4.322×10^{-15}	3.296×10^{-15}	5.033×10^{-15}	2.483×10^{-15}	0.2602	0.5195	2.8278	0.3655
F_{11}	0	0	0.0067	0.037	0.0102	0.0112	1.1822	0.0498
F_{12}	0.0625	0.0421	0.0616	0.0810	0.01383	0.0450	1.8768	1.2206
F_{13}	0.4312	0.2558	0.52868	0.3087	0.00490	0.0089	2.4337	1.0062
F_{14}	2.60118	2.9986	2.2797	2.2331	3.2016	2.9123	0.9980	2.332×10^{-16}
F_{15}	3.201×10^{-4}	6.822×10^{-4}	5.973×10^{-4}	2.463×10^{-4}	9.579×10^{-4}	3.186×10^{-4}	0.0061	0.0087
F_{16}	−1.0316	7.486×10^{-9}	−1.0316	1.030×10^{-9}	−1.0316	6.453×10^{-16}	−1.0316	6.453×10^{-16}
F_{17}	0.3978	1.702×10^{-6}	0.3979	5.029×10^{-6}	0.3978	0	0.3978	0
F_{18}	3.0000	1.828×10^{-4}	3.0000	5.326×10^{-5}	3.0000	1.414×10^{-15}	2.9999	2.332×10^{-15}
F_{19}	−3.8612	0.0021	−3.8614	0.0027	−3.8628	2.640×10^{-15}	−3.8627	2.696×10^{-15}
F_{20}	−3.3207	0.0794	−3.1671	0.2722	−3.2625	0.0605	−3.2863	0.0554
F_{21}	−8.9719	2.4585	−7.5223	2.9253	−7.6985	3.1463	−6.9949	3.6987
F_{22}	−8.0226	2.9570	−8.3803	2.9452	−8.6647	2.9806	−7.4903	3.6644
F_{23}	−7.9469	3.0241	−7.5665	3.1935	−9.0348	2.8326	−7.7227	3.5536

	DE		GSA		FEP	
F	*ave*	*std*	*ave*	*std*	*ave*	*std*
F_1	8.2×10^{-14}	5.9×10^{-14}	2.53×10^{-16}	9.67×10^{-17}	5.7×10^{-4}	1.3×10^{-4}
F_2	1.5×10^{-9}	9.9×10^{-10}	0.0556	0.1940	0.0081	7.7×10^{-4}
F_3	6.8×10^{-11}	7.4×10^{-11}	896.5347	318.9559	0.0160	0.0140
F_4	0	0	7.3548	1.7414	0.3000	0.5000
F_5	0	0	67.5430	62.2253	5.0600	5.8700
F_6	0	0	2.5×10^{-16}	1.74×10^{-16}	0	0
F_7	0.0046	0.0012	0.0894	0.04339	0.1415	0.3522
F_8	−11,080.1	574.70	−2821.07	493.0375	−12,554.5	52.600
F_9	69.2	38.80	25.9684	7.4700	0.0460	0.0120
F_{10}	9.7×10^{-8}	4.2×10^{-8}	0.0620	0.2362	0.0180	0.0021
F_{11}	0	0	27.7015	5.0403	0.0160	0.0220
F_{12}	7.9×10^{-15}	8×10^{-15}	1.7996	0.9511	9.2×10^{-6}	3.6×10^{-6}
F_{13}	5.1×10^{-14}	4.8×10^{-14}	8.8990	7.1262	1.6×10^{-4}	7.3×10^{-5}
F_{14}	0.9980	3.3×10^{-16}	5.8598	3.8312	1.2200	0.5600
F_{15}	4.5×10^{-14}	0.00033	0.0036	0.0016	0.0005	0.0003
F_{16}	−1.0316	3.1×10^{-13}	−1.0316	4.88×10^{-16}	−1.0300	4.9×10^{-7}
F_{17}	0.3978	9.9×10^{-9}	0.3978	0	0.3980	1.5×10^{-7}
F_{18}	3	2×10^{-15}	3	4.17×10^{-15}	3.0200	0.1100
F_{19}	N/A	N/A	−3.8627	2.29×10^{-15}	−3.8600	1.4×10^{-5}
F_{20}	N/A	N/A	−3.3177	2.29×10^{-15}	−3.2700	0.0590
F_{21}	−10.1532	25×10^{-7}	−5.9551	3.7370	−5.5200	1.5900
F_{22}	−10.4029	3.9×10^{-7}	−9.6844	2.0140	−5.5300	2.1200
F_{23}	−10.5364	1.9×10^{-7}	−10.5364	2.6×10^{-15}	−6.5700	3.1400

Table 3. Averages ranks, statistics and related *p*-value obtained by Friedman, Friedman aligned and Quade tests for IWOA and other comparative algorithms.

Order	Algorithm	Friedman test	Friedman aligned test	Quade test
1	IWOA	2.2609	52.0000	2.2028986
2	FEP	3.0870	61.3043	2.7934783
3	WOA	3.2391	65.0652	3.4184783
4	PSO	3.9130	70.7826	3.9456522
5	GA	3.9783	78.8043	4.1032609
6	GSA	4.5217	89.0434	4.5362319
p-value		7.6123×10^{-4}	2.4163×10^{-3}	1.7989×10^{-5}
Statistical value		21.1428	18.4653	6.68513

Note that the settling time is the time after which the response remains within a band of ±5% around the desired final value.

The gains of the controller are bounded as follows:

$$\begin{array}{c} K_p^{min} \le K_p \le K_p^{max} \\ K_i^{min} \le K_i \le K_i^{max} \\ K_d^{min} \le K_d \le K_d^{max} \\ K_{dd}^{min} \le K_{dd} \le K_{dd}^{max} \end{array}. \tag{18}$$

To increase the accuracy of the optimization process, the range of all gains are set to [0.001,4] in IWOA-PIDD^2. Such limitations will help IWOA to search more promising regions of the search space. Hence, we benefit from IWOA exploration and exploitation abilities, through investigating the search space globally, avoiding local optima and then refining the best solutions discovered to achieve the global optimum.

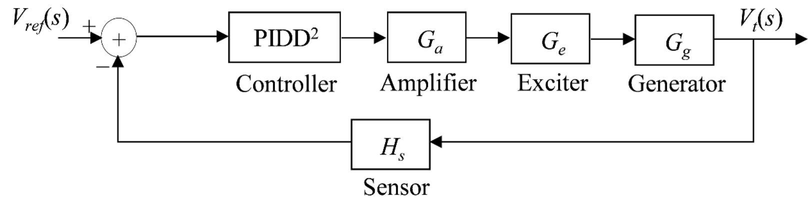

Figure 3. Control loop of the AVR system with PIDD^2 controller.

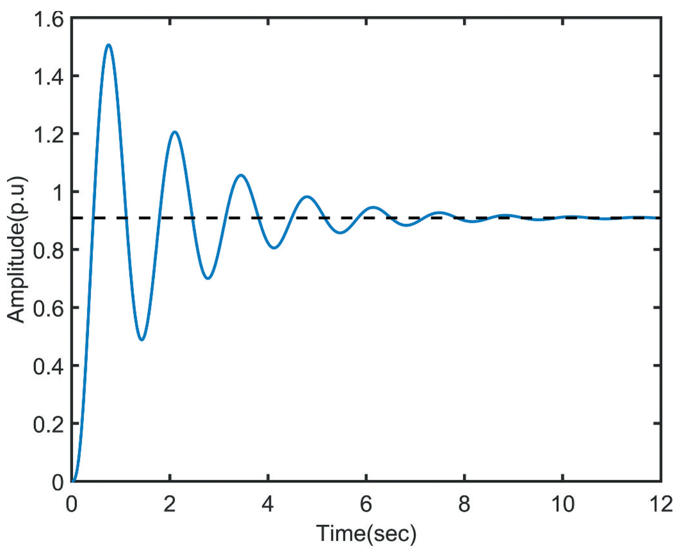

Figure 4. AVR system without PID controller closed-loop step response.

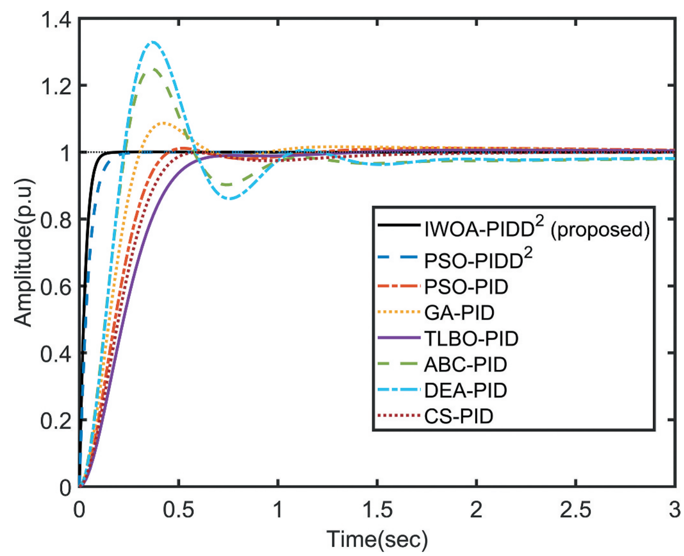

Figure 5. Comparison of the terminal voltage in AVR system with different algorithms.

5.3. *Results and discussion*

IWOA algorithm is used to optimize a PIDD^2 controller gains to achieve the desired response for the tuning of an AVR.

The simulation parameters of the IWOA algorithm are tabulated in Table 4.

Figure 5 shows the time responses of the proposed PIDD^2 controller compared to other well-known controllers such as PSO-PIDD^2 (Sahib 2015), PSO-PID (Gaing 2004), GA-PID (Gaing 2004), TLBO-PID (Chatterjee and Mukherjee 2016), ABC-PID (Gozde and Taplamacioglu 2011), DEA-PID (Sahu, Panda, and Rout 2013) and CS-PID (Bingul and Karahan 2018) for the same AVR system. From Figure 5, we can conclude that the proposed IWOA-PIDD^2 controller has better step response in comparison with other optimized PID controllers. Table 5 indicates the performance indices with the controller parameters.

Table 4. IWOA searching parameters.

Parameters	Values
Population size	30
Crossover function	Arithmetic crossover
Number of iteration	40

Inspecting the results in Table 5, it is evident that the proposed IWOA algorithm with PIDD^2 based controller provides the best dynamic responses compared to other meta-heuristic algorithms PSO, GA, TLBO, ABC, DEA and CS in terms of rise time t_r, settling time t_s and peak time t_p and obtains the third best value of overshoot M_p. The best response performance indices values are $M_p = 0.0653$, $t_r = 0.0584$, $t_s = 0.0982$ and $t_p = 0.3392$. From the numerical analysis, we find that IWOA-PIDD^2 controller reduces the values of time-domain performance indices. The corresponding control signals of the different PID controllers are shown in Figure 6.

Figure 7 shows the typical performance criteria (J) convergence curve using IWOA-PIDD^2 controller. We notice that after

Table 5. Best solution using various controllers.

Type of controller	K_p	K_i	K_d	K_{dd}	$M_p\%$	t_r $0.1 \rightarrow 0.9$	t_s $\pm 5\%$	t_p
IWOA-PIDD^2	3.9348	2.5753	1.3985	0.10453	0.0653	0.0584	0.0982	0.3392
PSO- PIDD^2	2.7784	1.8521	0.9997	0.07394	0.0023	0.0930	0.1635	0.4295
PSO-PID	0.6568	0.5393	0.2458	-	1.1652	0.2722	0.4111	1.9200
GA-PID	0.8861	0.7984	0.3158	-	8.6532	0.2041	0.6058	0.4222
TLBO-PID	0.5302	0.4001	0.1787	-	0.6369	0.3796	0.6245	2.0731
ABC-PID	1.6524	0.4083	0.3654	-	24.8559	0.1577	3.0939	0.3753
DEA-PID	1.9499	0.4430	0.3427	-	32.7495	0.1540	2.6502	0.3597
CS-PID	0.6198	0.4165	0.2126	-	0.0204	0.3277	1.1674	3.7471

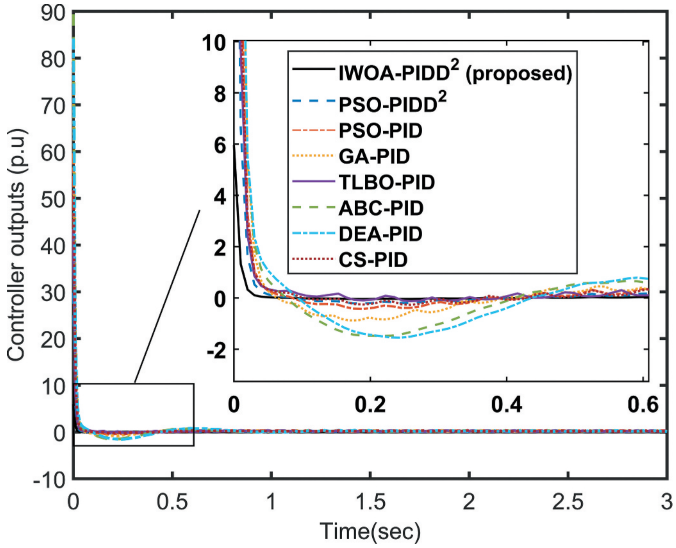

Figure 6. Control signals of different PID controllers.

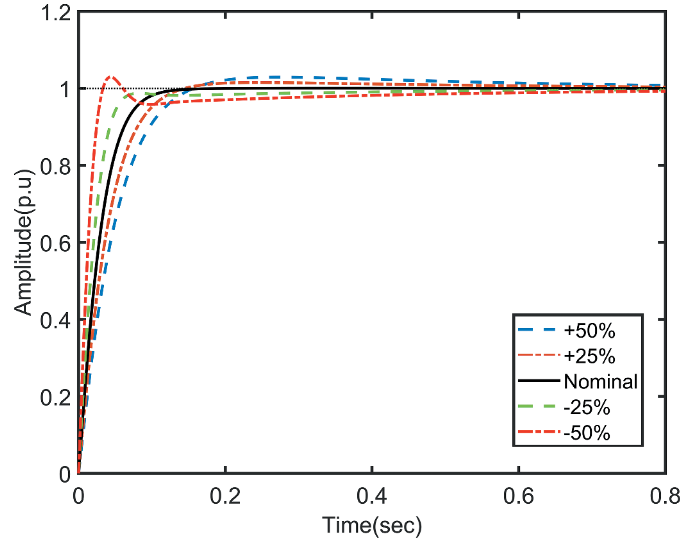

Figure 9. Step response curves with the range of −50% to +50% for τ_e.

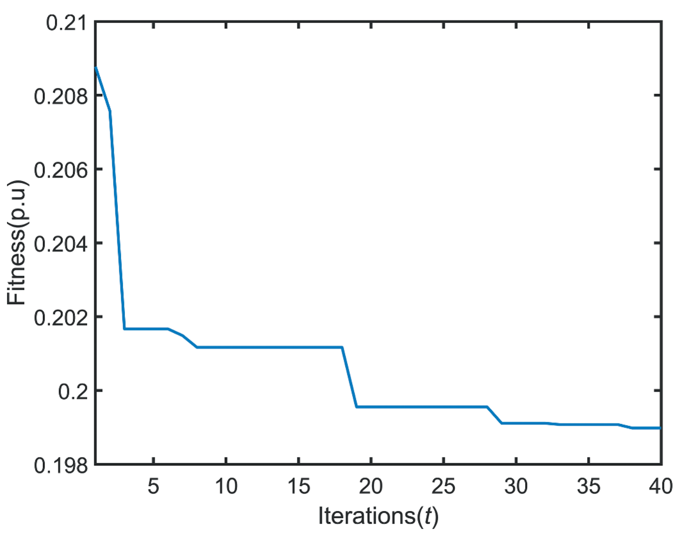

Figure 7. Performance criteria (*J*) convergence curve.

Table 6. Performance analysis of IWOA algorithm-based PIDD^2 controller for AVR system.

Parameter	Change ratio (%)	Peak value(p.u)	t_s	t_r	t_p
τ_a	+50%	1.0556	0.3267	0.0762	0.1838
	+25%	1.0286	0.2376	0.0677	0.1654
	−25%	0.9999	0.1997	0.0459	0.7533
	−50%	0.9998	0.2646	0.0316	0.8931
τ_e	+50%	1.0293	0.4710	0.0909	0.2724
	+25%	1.0153	0.1190	0.0758	0.2405
	−25%	0.9993	0.0693	0.0395	1.6565
	−50%	1.0289	0.3582	0.0258	0.0412
τ	+50%	1.0155	0.1555	0.0969	0.3414
	+25%	1.0082	0.1295	0.0780	0.3080
	−25%	0.9970	0.0753	0.0468	0.6439
	−50%	1.0552	0.1161	0.0241	0.0419
τ_s	+50%	1.0207	0.1044	0.0481	0.0984
	+25%	1.0034	0.0802	0.0523	0.1203
	−25%	1.0006	0.1163	0.0655	0.3584
	−50%	1.0006	0.1299	0.0723	0.3698

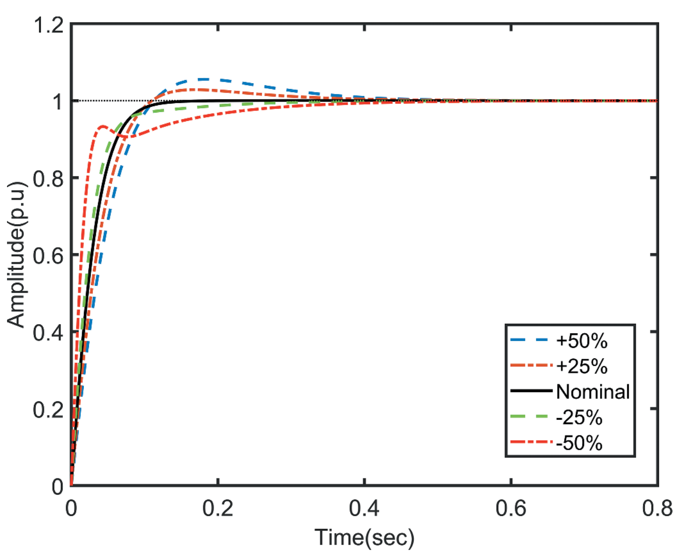

Figure 8. Step response curves with the range of −50% to +50% for τ_a.

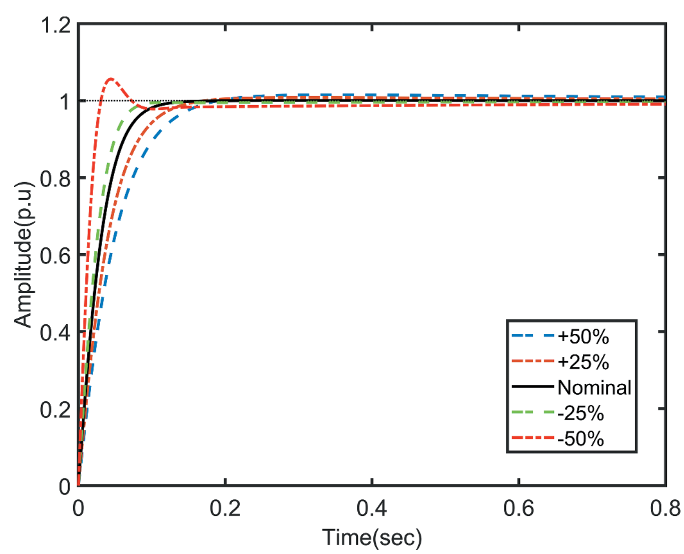

Figure 10. Step response curves with the range of −50% to +50% for τ_g.

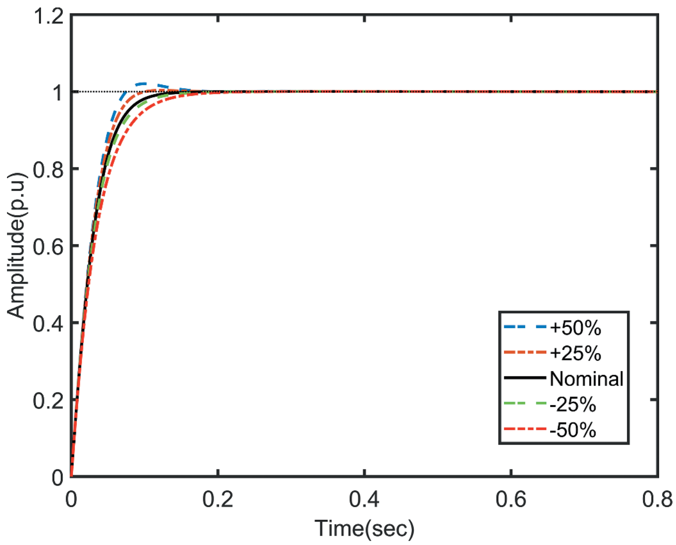

Figure 11. Step response curves ranging from −50% to +50% for τ_s.

40 iterations the objective function (J) converges at a value of 0.19898.

5.4. *Robustness analysis*

The proposed controller IWOA-PIDD^2 robustness is investigated on AVR system components under varying time constants of (amplifier, exciter, generator and sensor) separately, from $-50\ \%\mathrm{to}+50\ \%$ in steps of 25%. Figures 8–11 demonstrate the step responses of the AVR systems for the four changing time constants; it can be observed from the nominal response that parameters are within a small range as depicted in Table 6.

Table 7 lists the total deviation range and the maximum deviation percentage values for AVR system.

The average deviation percentage of performance index values are: maximum overshoot $M_p = 3.9\%$, settling time $t_s = 175.6\%,$, rise time $t_r = 44\%$ and peak time $t_p = 162.4\%$. It is clear from simulation results that the boundaries of total deviation are within the specified range, proving efficiency and robustness of the proposed PIDD^2 controller tuned by IWOA algorithm for AVR system.

The IWOA algorithm is superior over other algorithms in some characteristics such as fast convergence speed to reach the global optima while avoiding premature convergence to the local one. These merits are gained from the integration of the crossover operator to the standard WOA algorithm to provide diversification in the population by generating new solutions in the search region. In addition, the mechanisms of exploitation and exploration in WOA promote it over meta-heuristic techniques. IWOA algorithm is practical, very efficient and suitable for satisfying control engineering requirements related to minimizing computational time, overshoot and oscillations.

6. Conclusion

This work has proposed a new improved algorithm, IWOA, by incorporating the evolutionary operator crossover into the standard WOA to perform more effectively. The performance is benchmarked on 23 benchmark functions from the literature. Computational results proved that IWOA has good convergence speed to reach the global optima compared with some of the best meta-heuristic algorithms. Superiority of the proposed algorithm is demonstrated using three non-parametric statistical tests, namely, Friedman, Friedman aligned and Quade tests. The IWOA algorithm was applied successfully to search the optimal gain parameters of a PIDD^2, to control properly an AVR system. The time response characteristic parameters, such as maximum overshoot (M_p), settling time (t_s), rise time (t_r) and peak time (t_p) of the AVR system have been evaluated with IWOA-PIDD^2 and other existing controllers such as PSO-PIDD^2, PSO-PID, GA-PID, TLBO-PID, ABC-PID, DEA-PID, and CS-PID. Results indicate that the proposed controller improved, significantly, dynamic performance, namely, speed convergence and it is very suitable for the AVR system.

Moreover, robustness analysis reveals that the proposed IWOA-PIDD^2 controller has characteristics of high stability for AVR system and performs well even though changing the time constants of the system components.

For future work, the IWOA algorithm will be applied to optimize complex systems and be extended to design controllers in the class of nonlinear uncertain systems.

Nomenclature

A	random value in the interval [$-a$, a]
a	variable decreases from 2 to 0
b	constant
G_a	transfer function of the amplifier
G_e	transfer function of the exciter
G_g	transfer function of the generator
G_s	transfer function of the sensor
J	performance criteria

Table 7. Total and maximum deviation percentage for the AVR system.

	Total deviation range/max deviation percentage (%)			
Model parameters	Peak value	t_s	t_r	t_p
τ_a	0.0558/5.4%	0.1270/232.6%	0.0446/30.4%	0.7277/163.2%
τ_e	0.0300/2 .8%	0.4017/379.6%	0.0651/55.6%	1.6153/388.3%
τ_g	0.0582/5.4%	0.0802/58.3%	0.0728/66.0%	0.6020/89.2%
τ_s	0.0201/2.0%	0.0497/32.2%	0.0242/23.8%	0.2714/9.0%
Avreage	0.0410/3.9%	0.1646/175.6%	0.0516/44%	0.8041/162.4%

K_a	gain constant of the amplifier
K_e	gain constant of the exciter
K_g	gain constant of the generator
K_s	gdain constant of the sensor
K_d	derivative gain of $PIDD^2$ controller
K_{dd}	double derivative gains of $PIDD^2$ controller
K_i	integral gain of $PIDD^2$ controller
K_p	proportional gain of $PIDD^2$ controller
l	random value uniformly distributed in the range of [−1,1]
M_p	maximum overshoot
p	random number between 0 and 1
r	random value between 0 and 1
t	current iteration
t_p	peak time
t_r	rise time
t_s	settling time
t_{sim}	simulation time
V_{ref}	reference voltage
V_t	terminal voltage
$\vec{\boldsymbol{X}}$	position vector of a solution
$\vec{\boldsymbol{X}}^*$	position vector of the best solution
$\vec{\boldsymbol{X}}_{rand}$	whale selected randomly
τ_a	time constant of the amplifier
τ_e	time constant of the exciter
τ_s	time constant of the sensor
τ_g	time constant of the generator

Disclosure statement

No potential conflict of interest was reported by the authors.

References

Al Gizi, A. J. H., M. W. Mustafa, N. A. Al-geelani, and M. A. Alsaedi. 2015. "Sugeno Fuzzy PID Tuning, by Genetic-Neutral for AVR in Electrical Power Generation." *Applied Soft Computing* 28: 226–236. doi:10.1016/j.asoc.2014.10.046.

Bingul, Z., and O. Karahan. 2018. "A Novel Performance Criterion Approach to Optimum Design of PID Controller Using Cuckoo Search Algorithm for AVR System." *Journal of the Franklin Institute* 355 (13): 5534–5559. doi:10.1016/j.jfranklin.2018.05.056.

Chatterjee, S., and V. Mukherjee. 2016. "PID Controller for Automatic Voltage Regulator Using Teaching–Learning Based Optimization Technique." *International Journal of Electrical Power Energy Systems* 77: 418–429. doi:10.1016/j.ijepes.2015.11.010.

Dorigo, M., M. Birattari, and T. Stutzle. 2006. "Ant Colony Optimization." *IEEE Computational Intelligence Magazine* 1 (4): 28–39. doi:10.1109/MCI.2006.329691.

Eberhart, R. C., and Y. Shi. 1998. "Comparison between Genetic Algorithms and Particle Swarm Optimization." *Evolutionary Programming VII* 1447: 611–616. doi:10.1007/BFb0040812.

Eberhart, R. C., and Y. Shi. 2001. "Particle Swarm Optimization: Developments, Applications and Resources." *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat.N.01TH8546),* Seoul, South Korea, 27-30 May 2001:81–86. New York: IEEE.

Gaing, Z.-L. 2004. "A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System." *IEEE Transactions on Energy Conversion* 19 (2): 384–391. doi:10.1109/TEC.2003.821821.

Ghoshal, S. P. 2004. "Optimizations of PID Gains by Particle Swarm Optimizations in Fuzzy Based Automatic Generation Control." *Electric Power Systems Research* 72 (3): 203–212. doi:10.1016/j.epsr.2004.04.004.

Gozde, H., and M. C. Taplamacioglu. 2011. "Comparative Performance Analysis of Artificial Bee Colony Algorithm for Automatic Voltage Regulator (AVR) System." *Journal of the Franklin Institute* 348 (8): 1927–1946. doi:10.1016/j.jfranklin.2011.05.012.

Holland, J. H. 1992. "Genetic Algorithms." *Scientific American* 267 (1): 66–72. doi:10.1038/scientificamerican0792-66.

Kiam, H. A., G. Chong, and Y. Li. 2005. "PID Control System Analysis, Design, and Technology." *IEEE Transactions on Control Systems Technology* 13 (4): 559–576. doi:10.1109/TCST.2005.847331.

Köksoy, O., and T. Yalcinoz. 2008. "Robust Design Using Pareto Type Optimization: A Genetic Algorithm with Arithmetic Crossover." *Computers and Industrial Engineering* 55 (1): 208–218. doi:10.1016/j.cie.2007.11.019.

Lee, S.-C., and C.-L. Shih. 2012. "Optimal Single Input PID-Type Fuzzy Logic Controller." *Journal of the Chinese Institute of Engineers* 35 (4): 413–420. doi:10.1080/02533839.2012.655902.

Li, L.-M., K.-D. Lu, G.-Q. Zeng, L. Wu, and M.-R. Chen. 2016. "A Novel Real-Coded Population-Based Extremal Optimization Algorithm with Polynomial Mutation: A Non-Parametric Statistical Study on Continuous Optimization Problems." *Neurocomputing* 174: 577–587. doi:10.1016/j.neucom.2015.09.075.

Mafarja, M. M., and S. Mirjalili. 2017. "Hybrid Whale Optimization Algorithm with Simulated Annealing for Feature Selection." *Neurocomputing* 260: 302–312. doi:10.1016/j.neucom.2017.04.053.

Mirjalili, S. 2015. "Moth-Flame Optimization Algorithm. A Novel Nature-Inspired Heuristic Paradigm." *Knowledge-Based Systems* 89: 228–249. doi:10.1016/j.knosys.2015.07.006.

Mirjalili, S., and A. Lewis. 2016. "The Whale Optimization Algorithm." *Advances in Engineering Software* 95: 51–67. doi:10.1016/j.advengsoft.2016.01.008.

Panda, S., B. K. Sahu, and P. K. Mohanty. 2012. "Design and Performance Analysis of PID Controller for an Automatic Voltage Regulator System Using Simplified Particle Swarm Optimization." *Journal of the Franklin Institute* 349 (8): 2609–2625. doi:10.1016/j.jfranklin.2012.06.008.

Raju, M., L. C. Saikia, and N. Sinha. 2016. "Automatic Generation Control of a Multi-Area System Using Ant Lion Optimizer Algorithm Based PID Plus Second Order Derivative Controller." *International Journal of Electrical Power Energy Systems* 80: 52–63. doi:10.1016/j.ijepes.2016.01.037.

Rashedi, E., H. Nezamabadi-pour, and S. Saryazdi. 2009. "GSA: A Gravitational Search Algorithm." *Information Sciences* 179 (13): 2232–2248. doi:10.1016/j.ins.2009.03.004.

Sahib, M. A. 2015. "A Novel Optimal PID Plus Second Order Derivative Controller for AVR System." *Engineering Science and Technology, an International Journal* 18 (2): 194–206. doi:10.1016/j.jestch.2014.11.006.

Sahu, R. K., S. Panda, and U. K. Rout. 2013. "DE Optimized Parallel 2-DOF PID Controller for Load Frequency Control of Power System with Governor Dead-Band Nonlinearity." *International Journal of Electrical Power Energy Systems* 49: 19–33. doi:10.1016/j.ijepes.2012.12.009.

Storn, R., and K. Price. 1997. "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces." *Journal of Global Optimization* 11: 341–359. doi:10.1023/A:1008202821328.

Tsai, H. C., and Y. H. Lin. 2011. "Modification of the Fish Swarm Algorithm with Particle Swarm Optimization Formulation and Communication Behavior." *Applied Soft Computing* 11 (8): 5367–5374. doi:10.1016/j.asoc.2011.05.022.

Wong, -C.-C., and S.-A. Li. 2014. "Switching-Type PD-PI Controller Design by HEA for AVR System." *Journal of the Chinese Institute of Engineers* 37 (5): 687–697. doi:10.1080/02533839.2013.815011.

Yao, X., Y. Liu, and G. Lin. 1999. "Evolutionary Programming Made Faster." *IEEE Transactions on Evolutionary Computation* 3 (2): 82–102. doi:10.1109/4235.771163.

Zeng, G. Q., J. Chen, M. R. Chen, Y. X. Dai, L. M. Li, K. D. Lu, and C. W. Zheng. 2015a. "Design of Multivariable PID Controllers Using Real-Coded Population-Based Extremal Optimization." *Neurocomputing* 151: 1343–1353. doi:10.1016/j.neucom.2014.10.060.

Zeng, G. Q., J. Chen, Y. X. Dai, L. M. Li, C. W. Zheng, and M. R. Chen. 2015b. "Design of Fractional Order PID Controller for Automatic Regulator Voltage System Based on Multi-Objective Extremal Optimization." *Neurocomputing* 160: 173–184. doi:10.1016/j.neucom.2015.02.051.

Zeng, G. Q., X. Q. Xie, M. R. Chen, and J. Weng. 2019. "Adaptive Population Extremal Optimization Based PID Neural Network for Multivariable Nonlinear Control Systems." *Swarm and Evolutionary Computation* 44: 320–334. doi:10.1016/j.swevo.2018.04.008.