Decision Analytics Journal 7 (2023) 100205

Contents lists available at ScienceDirect

Decision Analytics Journal

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

A self-competitive mutation strategy for Differential Evolution algorithms with applications to Proportional–Integral–Derivative controllers and Automatic Voltage Regulator systems

Mojtaba Ghasemi [a], Abolfazl Rahimnejad [b,*], Milad Gil [c], Ebrahim Akbari [d], S. Andrew Gadsden [b]

[a] *Department of Electronics and Electrical Engineering, Shiraz University of Technology, Shiraz, Iran*
[b] *Department of Mechanical Engineering, McMaster University, Hamilton, Canada*
[c] *Department of Electrical and Computer Engineering, Babol Noshirvani University of Technology, Babol, Iran*
[d] *Department of Mathematics, Faculty of Science, University of Hradec Králové, Czech Republic*

ARTICLE INFO

Keywords:
Optimization
Differential Evolution
Real parameter test functions
Self-competitive mutation
Proportional–Integral–Derivative
Automatic Voltage Regulator

ABSTRACT

The Differential Evolution (DE) algorithm is a powerful and simple optimizer for solving various optimization problems. Based on the literature, DE has shown suitable performance in exploring search spaces and locating global optimums. However, it is typically slow in extracting the problem solution. In this paper, the exploration ability of the DE algorithm is augmented with a competitive control parameter ω based on the value of the objective function of the mutating members. A new mutation strategy is introduced, subtracting weaker members from superior weaker ones. The proposed DE algorithm, which is referred to as the self-competitive DE, has been employed for solving real-world optimization problems. Several DE algorithms are enhanced with the proposed parameter ω, and the efficiencies of the resulting enhanced algorithms are tested. Furthermore, the optimal Proportional–Integral–Derivative (PID) controller tuning for an Automatic Voltage Regulator (AVR) system is used to investigate the effectiveness of the proposed strategy in solving real-world optimization problems. Simulation results demonstrate a good performance of the proposed parameter ω over several other well-known DE algorithms.

1. Introduction

There are some cases in engineering and optimization problems that cannot be solved using conventional analytic approaches because either the analytical solution is unavailable or very difficult to obtain, or complex functions and numerous parameters of the problem introduce a large number of solutions making it difficult to assess all of the possible solutions. Evolutionary and swarm intelligence algorithms with improved versions for example symbiotic organisms search (SOS) [1], sooty tern optimization algorithm (STOA) [2], salp swarm algorithm (SSA) [3], stochastic fractal search algorithm (SFS) [4], simulated annealing (SA) [5], circulatory system based optimization (CSBO) [6], are some random search methods inspired by natural biological evolution modeling. These algorithms benefit from superior features while working on possible solutions, which provide more close estimations of the optimal solution. This paper proposes a modification to the basic DE algorithm and its improved variants. This optimizer, which was first introduced in [7], is a class of evolutionary algorithms and has a very simple structure from conception and implementation points of view. In their next study [8], the authors of the paper compared the performance of DE with several popular optimization methods. The comparison proved the superiority of the DE algorithm to the other optimization methods. From then on, this algorithm has been the main topic of a large number of scientific articles in the literature on evolutionary algorithms. Today, DE is known as a very effective optimization method for continuous environments and is used in a wide range of engineering problems.

In recent years, improved and modified versions and different combinations and applications of the DE algorithm have been presented. Various mutation strategies with both exploitation and exploration abilities have been used by scholars for local and global search purposes [9–11]. A number of recently introduced mutation operators include Gaussian mutation [12], Gaussian PBX-α [13], DE/rand-to-best/pbest [13], and rotation-invariant mutation operator [14]. Mutation strategies also use the data associated with the best and worst adjacent members [15]. Two factors highly impact DE performance. The first one is the strategy adopted for producing new offspring. This is carried out using mutation and crossover operators. The second factor is the employed mechanism for controlling the algorithm parameters including crossover rate, scaling factor, and the number of the population [16]. The optimization process is repeated until the termination criterion of the algorithm is satisfied [17,18].

* Corresponding author.
E-mail addresses: a.rahimnejad@gmail.com, rahimnea@mcmaster.ca (A. Rahimnejad).

https://doi.org/10.1016/j.dajour.2023.100205
Received 25 November 2022; Received in revised form 6 March 2023; Accepted 20 March 2023
Available online 22 March 2023

In the following, the main topics of research on the DE algorithm are presented.

(1) During the last decades, several competent mutation strategies have been employed on DE. An improved mutation strategy called "DE/current-to-pbest" was introduced in [19] to improve the performance of the traditional DE. In another study, the cultural algorithm (CA) with diversity measure was utilized to improve the DE in terms of optimization [19]. A successful-parent selection method was given in [20], which adapts the parent selection during stagnation.

(2) In some studies, parameter control methods were provided for the DE algorithm. In [20,21], the weight of the mutation (F) and the probability of crossover (CR) parameters experience evolution during the population evolution. Composite DE (CoDE) [22], and self-adaptive DE (SaDE) [23] are among the DE algorithms with a set of different mutation operators. In SaDE, mutation operators and the related control parameters are self-adapted over time using experience while providing suitable solutions. However, in CoDE, new solutions are obtained by merging three original mutation operators and with new control parameter settings based on comparison with different strategies. Additionally, an adaptive population tuning method is presented in [24] to reallocate the computing resources more reasonably.

(3) Optimization of the conventional crossover strategy has also been discussed in the literature, e.g. a multimodal method in DE [25]. Also, eigenvectors of the covariance matrix are used in [26] to make the crossover rotationally invariant and so a better search ability is achieved.

(4) Some researchers have presented several other DE modifications, where more than one mutation operator is used to breed new solutions, such as EPSDE [27,28], a multi-population-based DE (MDE) [29], etc. To modify the performance of the DE, intelligent selection of mutation vectors has deeply been investigated in some studies [30].

Ref. [30] suggests a multi-population DE with a balanced ensemble of mutation strategies known as mDE-bES. Exploitation and exploration must be utilized in the evolution strategy at the same time to boost the robustness of the algorithm. Hence, single-mutation and multi-mutation operator strategies with combined search skills were introduced. The first type includes BDE [31], CCDE [32], ProDE [33], BoRDE [34], TLBSaDE [35], and TDE [36]. And, the second type includes PDE [37], DEPSO [38], a modified DE using a diversity maintenance strategy [39], DE algorithms based on nonparametric statistical tests [40], jDEsoo [41], and SPSRDEMMS [42]. Other novel ideas have also been used in the DE algorithms besides the above-mentioned studies. Ref. [43] suggested using the opposition-based DE (ODE), where opposition-based learning, a novel approach in machine intelligence, is employed to increase the convergence rate of DE. In [44] the attraction–repulsion concept in electromagnetism was utilized to modify the optimization power of the original DE with hybrid mutation. Furthermore, in [31], explorative and exploitive mutation operators are combined in a linear way to give a hybrid approach (BDE) which makes a balance between these two operators. A new hybrid of DE and particle swarm optimizer [45]. In CCDE (colonial competitive DE) [32], various types of DE algorithms are enhanced using the concepts of socio-political evolution. Neglecting half of the valuable data, when determining the mutation vectors, is one of the deficiencies of the mentioned methods. In the ProDE method, individuals in the surrounding area of the target individual are chosen to be involved in the mutation operation [33]. The authors of [46] proposed an adaptive DE algorithm with a Lagrange interpolation argument algorithm.

Also, a teaching-learning procedure was used by TLBSaDE, a new version of SaDE, to produce mutant individuals [35]. A novel multi-objective modified DE (MOMDE) was introduced in [39] using a new diversity maintenance strategy. By using a SaDE-type learning scheme, the researchers in [40] modified the EPSDE. A parallel DE algorithm with a generalized opposition-based learning strategy and self-adapting control parameters was presented in PDE [37]. A new self-adaptive mutation for DE was proposed in [47] and a new multi-population DE via an ensemble of different mutation strategies was proposed in [48]. In SPSRDEMMS [42], a single mutation strategy is selected from DE/best/1 and DE/rand/1 to produce the mutant individual as the population number is reduced through the iterations. An automated control parameter adaptation approach for CR and F has been studied in [49], which increases the robustness. Hence, this method can be employed instead of a method with fixed control parameters. In addition, a selection rule was presented for population size (NP) in [49]. A modified DE with a new role assignment technique was investigated in [50]. Novel parameter control methods of DE are provided in [51,52]. Several selection methods were also given in [53] for the best individual in DE/current-to-pbest/1, DE/rand-to-best/1, and DE/best/1. The authors of [54] present a similar rank-based parent selection scheme, and the authors of [54] proposed a new micro DE.

In this paper, a new mutation strategy, named self-competitive mutation strategy, is proposed for DE algorithms in a way that the movement is always towards the members with better fitness values; i.e. the mutation equations are reformatted so that always the weaker members are subtracted from the other one. This way it is ensured that the algorithm would always tend to search the regions with a higher chance of improving the solutions. This mutation strategy is then applied to several basic and modern DE algorithms, which have been widely used in recent years, without altering the other parameters and structure of the selected algorithms. After that, the basic and advanced versions of the proposed DE algorithms are applied to some standard test functions, the results of which show that the proposed mutation strategy enhance the performance of the algorithms in solving various range of functions. Friedman rank test and Wilcoxon signed-rank test are employed for assessing the performance of the proposed enhanced variants of the DE algorithm [55,56].

The purpose of this comprehensive comparative study is to show the effect of the proposed self-competitive mutation strategy on different variants of basic and modern DE algorithms and also, the optimal design of a proportional–integral–derivative (PID) controller for an automatic voltage regulator (AVR) system is used to investigate the effectiveness of the proposed strategy in solving real-world optimization problems. The optimal design of a PID controller for the AVR system is very complex, nonlinear, and non-convex. Thus, the present study combines the power of the best self-competitive DE (SCDE) algorithm to propose a novel robust algorithm for the optimal design of a PID controller for the AVR system.

Here are the main contributions of this paper:

1. Hybridizing basic and modern DE algorithms with the proposed self-competitive mutation strategy developed novel, efficient, and robust optimization algorithms named self-competitive DE (SCDE) algorithms.
2. This work addresses the very complex, nonlinear, and non-convex characteristics of real-world optimization problems by using the standard shifted rotated multimodal and hybrid composition functions from the popular and very widely used CEC 2005. The proposed SCDE algorithms have been employed for solving these standard real parameter test functions.
3. Two non-parametric statistical tests, i.e., Friedman rank test and Wilcoxon signed-rank test, are utilized for investigating the performance of different DE algorithms augmented with the proposed self-competitive strategy.
4. An optimal scheduling of a PID controller for the AVR system is determined by the best SCDE algorithm in comparison with its original version.

The rest of the paper is organized as follows. In Section 2, a brief review of the basic and advanced versions of the DE algorithm is presented. The proposed self-competitive versions are introduced and discussed in Section 3. In Section 4, a comparative study of the results obtained by the basic and advanced versions of DE and their self-competitive counterparts is presented. In Section 5, a comparative application of the proposed best SCDE algorithm in designing an optimal PID controller for the AVR system is presented. Finally, the paper is concluded.

2. Differential evolution and its variants

Metaheuristic algorithms are generally known as general-purpose optimization algorithms that are able to find near-optimal solutions for mathematical and real-world problems, while classical and analytic methods are not able to find the optimal solution at a reasonable computational time. One of these evolutionary algorithms that have been broadly employed in various fields is the differential evolution algorithm, which was first introduced by [7,8]. The differential evolution algorithm deals with a population of individuals that denote chromosomes in the genetic space and represent vector values as solutions in the problem space. This algorithm is among the algorithms that work with real variables, which is considered one of the advantages of this algorithm. DE was proposed to solve the main demerits of genetic algorithms (GAs), i.e. the absence of local search. The main distinction between GA and DE is in their mutation, crossover, and selection operators. DE uses a differential operator to generate new solutions, which have the ability to exchange information among members of the population. One of the advantages of DE is that it includes a memory that keeps the info of suitable solutions in the current individuals. Furthermore, in DE, all individuals have the same likelihood of being elected as one of the parents. The DE algorithm has high speed, simplicity, and robustness. It has three parameters: NP which is the population size, F which shows the weight of the mutation and CR which is the probability of crossover. The parameter F is usually selected between 0 to 2 and the parameter CR is selected between 0 and 1.

The components and steps of DE, in general, include:

1. Establishment of an initial population.
2. Mutation operator
3. Crossover operator
4. Selection operator
5. Termination criterion of the algorithm.

The following describes each of these steps for DE algorithms.

Step 1: Establishment of the initial population

Similar to other evolutionary algorithms the starting point of the algorithm ($Iter = 1$) is the establishment of an initial population. In this algorithm, the initial population is created randomly with a uniform distribution with a size of NP in the range of $X_{\min} = \left[x_{min,1}, x_{min,2}, \ldots, x_{min,D}\right]$ and $X_{\max} = \left[x_{max,1}, x_{max,2}, \ldots, x_{max,D}\right]$ considering the dimension D of the problem. The jth decision variable of ith chromosome (solution) can be initialized as:

$$x^{i}_{1,j} = rand_j\,(0,1) \times \left(x_{j,\max} - x_{j,\min}\right) + x_{j,\min} \tag{1}$$

In the above relation:
$rand_j\,(0,1)$ is a function that generates random numbers between 0 and 1 for the jth ($j = 1, 2, 3, \ldots, D$) dimension of the variable.

Step 2: Mutation Operator

In this step, for each target vector, e.g. the ith member X^{i}_{Iter} (i = 1, 2, .., NP), several members are randomly selected, e.g. $X^{r_1}_{Iter}$, $X^{r_2}_{Iter}$, $X^{r_3}_{Iter}$, $X^{r_4}_{Iter}$, and $X^{r_5}_{Iter}$, and the mutation vector (V^{i}_{Iter}) is generated based on the corresponding mutation equation. The following relationships are some of the most used equations for mutation in DE [32]:

"DE/rand/1":

$$V^{i}_{Iter} = X^{r_1}_{Iter} + F\left(X^{r_2}_{Iter} - X^{r_3}_{Iter}\right) \tag{2}$$

"DE/best/1":

$$V^{i}_{Iter} = X_{best} + F\left(X^{r_1}_{Iter} - X^{r_2}_{Iter}\right) \tag{3}$$

"DE/current-to-rand/1":

$$V^{i}_{Iter} = X^{i}_{Iter} + F\left(X_{best} - X^{i}_{Iter}\right) + F\left(X^{r_1}_{Iter} - X^{r_2}_{Iter}\right) \tag{4}$$

"DE/rand/2":

$$V^{i}_{Iter} = X^{r_1}_{Iter} + F\left(X^{r_2}_{Iter} - X^{r_3}_{Iter}\right) + F\left(X^{r_4}_{Iter} - X^{r_5}_{Iter}\right) \tag{5}$$

"DE/rand-to-best/1":

$$V^{i}_{Iter} = X^{r_1}_{Iter} + F\left(X_{best} - X^{r_1}_{Iter}\right) + F\left(X^{r_2}_{Iter} - X^{r_3}_{Iter}\right) \tag{6}$$

"DE/current-to-rand/1":

$$\begin{aligned} V^{i}_{Iter} &= X^{i}_{Iter} + rand(0,1) * \left(X^{i}_{Iter} - X^{r_1}_{Iter}\right) \\ &+ F * rand(0,1) * \left(X^{r_2}_{Iter} - X^{r_3}_{Iter}\right) \end{aligned} \tag{7}$$

where $\left(X^{r_1}_{Iter} - X^{r_2}_{Iter}\right)$, $\left(X^{r_2}_{Iter} - X^{r_3}_{Iter}\right)$ and $\left(X^{r_4}_{Iter} - X^{r_5}_{Iter}\right)$ are different vectors that mutate the base vector. X_{best} is the best individual vector with the best fitness value in the current population at iteration *Iter*.

Step 3: Crossover operator

In this step, a crossover is built between the mutation and target vectors (V^{i}_{Iter} and X^{i}_{Iter}, respectively), and the trial vector (U^{i}_{Iter}) is produced based on Eq. (8) such that each component of the mutation vector is transferred with a probability of CR to the trial vector, otherwise the corresponding in the original vector component is transferred to the trial vector.

$$u^{i}_{Iter,j} = \begin{cases} v^{i}_{Iter,j}, & if \quad \left(rand_{i,j}(0,1) \le CR\right) \\ x^{i}_{Iter,j}, & otherwise \end{cases} \tag{8}$$

Step 4: Selection

In this step, the values of the objective functions for the trial vectors are evaluated. In the minimization problem, if the trial vector has a value lower than the target vector, it is selected as one of the members of the next generation (X^{i}_{Iter+1}); otherwise, the target vector (X^{i}_{Iter}) will be transferred to the population in the next generation.

$$X^{i}_{Iter+1} = \begin{cases} U^{i}_{Iter}, & if \quad \left(f(U^{i}_{Iter}) \le f(X^{i}_{Iter})\right) \\ X^{i}_{Iter}, & otherwise. \end{cases} \tag{9}$$

This procedure continues until *NP* new members are generated for the next generation. Then the procedure is iterated to meet the termination criterion. The pseudo-code of this algorithm with DE/rand/1 mutation strategy is shown in Algorithm 1.

In the following, four variants of DE are introduced:

2.1. jDE

A novel adaptive DE, known as jDE, was introduced in [21] by imitating the classic DE/rand/1 method. The new method is able to adjust the population size and tune the control parameters F_i and CR_i of the individuals in the optimization process. The values of these two quantities for individuals are assumed 0.5 and 0.9, respectively, in the initialization step.

New values of these two parameters are reproduced based on uniform distributions in the ranges of [0.1, 1] and [0,1], respectively, using the jDE algorithm. Better values lead to individuals with a higher chance of survival.

2.2. EPSDE

EPSDE [28] performs two mutation strategies, namely DE/rand/1 and DE/current-to-pbest/1, at the same time, where the probability of generating offspring is adapted based on their success ratios in the previous fifty generations. The adaptation process is able to develop the most desired mutation strategy at the consequent learning steps. In this process, the participant heuristics (such as different versions of the differential evolution algorithm, simplex methods, and evolution strategies) are taken into account at the same time, and the probabilities of generating offspring by these heuristics are matched dynamically.

Algorithm 1:

Set values of the control parameters of DE/rand/1: *NP, $Iter_{max}$, CR*, and *F*.
Create the initial random population *NP* (*i*=1, 2, …, *NP*);
Evaluate the fitness of each individual in the population;
while the parameter *Iter* is lower than the maximum number of iterations, $Iter_{max}$, **do**
Increase the iteration number (*Iter*= *Iter*+1);
for $i = 1$ to *NP* **do**
Choose three random members, e.g. $r_1 \neq r_2 \neq r_3 \neq i$;
for $j = 1$ to *D* **do**
$v^{i}_{Iter,j} = x^{r_1}_{Iter,j} + F\left(x^{r_2}_{Iter,j} - x^{r_3}_{Iter,j}\right)$;
end for
for $j = 1$ to *D* **do**
$u^{i}_{Iter,j} = \begin{cases} v^{i}_{Iter,j}, & if \quad \left(rand_{i,j}(0,1) \leq CR\right) \\ x^{i}_{Iter,j}, & otherwise \end{cases}$;
end for
for $j = 1$ to *D* **do**
$X^{i}_{Iter+1} = \begin{cases} U^{i}_{Iter}, & if \quad \left(f(U^{i}_{Iter}) \leq f(X^{i}_{Iter})\right) \\ X^{i}_{Iter}, & otherwise. \end{cases}$;
if $f(X^{i}_{Iter+1}) < f(X_{best})$
$X_{best} = X^{i}_{Iter+1}$ and $f(X_{best}) = f(X^{i}_{Iter+1})$;
end if
end for
end for
end while

2.3. SaDE

In SaDE [23], a normal distribution is used to produce mutation factors separately at each generation. The proposed approach has proved a great potential at providing both local and global search by presenting suitable mutation vectors during the evolution step. The probabilities of the crossover operation are also randomly produced using a normal distribution. In contrast to F_i, the values of CR_i are only changed once per six consecutive generations. A local search is used for a number of suitable individuals after 200 generations to accelerate the convergence speed of the SaDE.

2.4. JADE

JADE [19] method is employed to enhance the optimization performance. It carries out the DE/current-to-pbest mutation strategy and updates control parameters adaptively to the desired values. According to the simulations, JADE provides more suitable performance compared to conventional or adaptive DE algorithms, canonical particle swarm optimization, and other evolutionary algorithms from the convergence speed viewpoint.

3. The proposed strategy, self-competitive differential evolution (SCDE) algorithms

This paper proposes a simple yet effective strategy called the self-competitive strategy which modifies the mutation operator of different variants of the DE algorithm. In order to implement the proposed strategy to any variant of DE, the following modification is made to its mutation operator:

If there is a subtraction of two randomly selected members to form the difference vector, it is multiplied by a control parameter. The parameter is selected so that it is guaranteed that always the weaker member is subtracted from the better one. Although the method is straightforward, based on the subsequent results and discussions, it is useful for some of the most widely used DE models.

For instance, in the DE/rand/1 model algorithm, which includes the subtraction of two randomly selected members to form the difference vector $\left(X^{r_2}_{Iter} - X^{r_3}_{Iter}\right)$, the implementation of the proposed strategy starts by defining a control parameter ω, given in Eq. (10), which can be one of three values of 1, −1, or 0. This parameter is selected such that multiplying it by $\left(X^{r_2}_{Iter} - X^{r_3}_{Iter}\right)$ and thus obtaining $\omega^{23}_{Iter} \times \left(X^{r_2}_{Iter} - X^{r_3}_{Iter}\right)$ leads to the subtraction of the member with a higher cost function from the other member.

$$\omega^{23}_{Iter} = \frac{f\left(X^{r_3}_{Iter}\right) - f\left(X^{r_2}_{Iter}\right)}{\left|f\left(X^{r_3}_{Iter}\right) - f\left(X^{r_2}_{Iter}\right)\right| + \varepsilon} = \begin{cases} 1 & if\ f\left(X^{r_2}_{Iter}\right) < f\left(X^{r_3}_{Iter}\right), \text{i.e. } if\ X^{r_2}_{Iter} \text{ is better than } X^{r_3}_{Iter} \\ -1 & if\ f\left(X^{r_2}_{Iter}\right) > f\left(X^{r_3}_{Iter}\right), \text{i.e. } if\ X^{r_3}_{Iter} \text{ is better than } X^{r_2}_{Iter} \\ 0 & if\ f\left(X^{r_2}_{Iter}\right) = f\left(X^{r_3}_{Iter}\right) \end{cases} \tag{10}$$

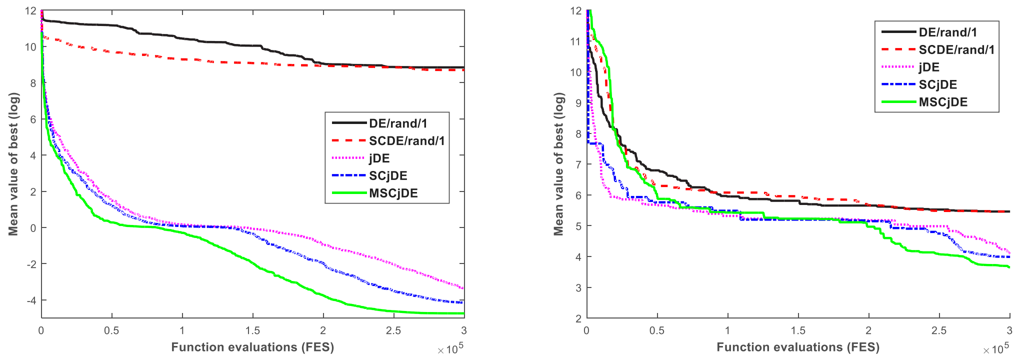


Fig. 1. The convergence characteristics of these five algorithms for Unimodal 4 and Multimodal 10 test functions.

Table 1
A typical example of the self-competitive strategy.

ω_{Iter}^{23}	ω_{Iter}^{45}	$V_{Iter}^{i}=X_{Iter}^{r_1}+\omega_{Iter}^{23}\times F\times\left(X_{Iter}^{r_2}-X_{Iter}^{r_3}\right)+\omega_{Iter}^{45}\times F\times\left(X_{Iter}^{r_4}-X_{Iter}^{r_5}\right)$
1	1	$X_{Iter}^{r_1}+F\times\left(X_{Iter}^{r_2}-X_{Iter}^{r_3}\right)+F\times\left(X_{Iter}^{r_4}-X_{Iter}^{r_5}\right)$
1	−1	$X_{Iter}^{r_1}+F\times\left(X_{Iter}^{r_2}-X_{Iter}^{r_3}\right)+F\times\left(X_{Iter}^{r_5}-X_{Iter}^{r_4}\right)$
1	0	$X_{Iter}^{r_1}+F\times\left(X_{Iter}^{r_2}-X_{Iter}^{r_3}\right)$
−1	1	$X_{Iter}^{r_1}+F\times\left(X_{Iter}^{r_3}-X_{Iter}^{r_2}\right)+F\times\left(X_{Iter}^{r_4}-X_{Iter}^{r_5}\right)$
−1	−1	$X_{Iter}^{r_1}+F\times\left(X_{Iter}^{r_3}-X_{Iter}^{r_2}\right)+F\times\left(X_{Iter}^{r_5}-X_{Iter}^{r_4}\right)$
−1	0	$X_{Iter}^{r_1}+F\times\left(X_{Iter}^{r_3}-X_{Iter}^{r_2}\right)$
0	1	$X_{Iter}^{r_1}+F\times\left(X_{Iter}^{r_4}-X_{Iter}^{r_5}\right)$
0	−1	$X_{Iter}^{r_1}+F\times\left(X_{Iter}^{r_5}-X_{Iter}^{r_4}\right)$
0	0	$X_{Iter}^{r_1}$

In the above equation, the value of ε is a small value that is used to avoid a zero denominator. In order to thoroughly describe the proposed strategy, the cases that can occur in improving the DE/rand/2 algorithm and forming its self-competitive version (SCDE/rand/2), are presented in Table 1. In this table ω_{Iter}^{23} and ω_{Iter}^{45} are the parameters which are multiplied by the difference vectors $\left(X_{Iter}^{r_2}-X_{Iter}^{r_3}\right)$ and $\left(X_{Iter}^{r_4}-X_{Iter}^{r_5}\right)$, respectively.

Similarly, the proposed self-competitive strategy can be used for improving other variants of DE algorithms, which are formulated in the next section.

4. Experimental results

To show the impact of the competitive control parameter ω on various DE algorithms, 25 widely used real parameter test functions of dimension 30 were employed from CEC 2005; whose data and details are provided in [57]. The number of function evaluations is considered 300,000 for all algorithms, the same as that of [57]. These test functions include the following: Unimodal Functions (F1 to F5), Basic Multimodal Functions (F6 to F12), Expanded Multimodal Functions for the (F13 and F14), and Hybrid Composition Functions (F15 to F25).

In Sections 4.1 to 4.4, the impact of the competitive control parameter ω on different DE algorithms is demonstrated. In each section, the Wilcoxon signed-rank test with a significance level of 0.05 is used for pairwise comparison of one of the proposed self-competitive DE algorithms with other relevant algorithms. Then in Section 4.5, the results of the Friedman test are presented, through which the best-proposed variant of DE in terms of average ranking is identified.

4.1. SCDE/rand/1 and SCjDE

In this part, the performance of the proposed method on the DE/rand/1 algorithm and the improved DE algorithm known as jDE, which is an adaptive and modified version of the DE/rand/1 algorithm, is analyzed. Codes of the jDE algorithm are extracted from (http://dces.essex.ac.uk/staff/qzhang/) with the same control parameters. The self-competitive versions of DE and jDE are called SCDE and SCjDE, respectively. The only change made to the DE and jDE to achieve the SCDE and SCjDE algorithms is using the control parameter coefficient ω_{Iter}^{23}, and their other conditions are identical to those of their original counterparts. The values of $F = 0.9$ and $CR = 0.9$ are set for algorithms DE/rand/1 and SCDE/rand/1. Also, $NP = 60$ is chosen for these two algorithms.

- Modified SCjDE (MSCjDE)

The proposed SCjDE algorithm can be further improved by using (11) as a mutation equation. In (11), the mutation vector $X_{Iter}^{r_1}$ is compared with the current vector X_{Iter}^{i}, and the better one is used in the mutation process. The modified SCjDE algorithm is called MSCjDE.

$$V_{Iter}^{i}=\begin{cases} X_{Iter}^{i}+\omega_{Iter}^{23}\times F\times\left(X_{Iter}^{r_2}-X_{Iter}^{r_3}\right) & if\ f\left(X_{Iter}^{i}\right)<f\left(X_{Iter}^{r_1}\right)\\ X_{Iter}^{r_1}+\omega_{Iter}^{23}\times F\times\left(X_{Iter}^{r_2}-X_{Iter}^{r_3}\right) & if\ f\left(X_{Iter}^{i}\right)\geq f\left(X_{Iter}^{r_1}\right)\end{cases} \tag{11}$$

The results of the algorithms given in this section are listed in Table 2. In this table and other Tables, subscripts $=/-/+$ show the comparison with the original algorithm in the same table: '=' means the equal result, '−' means the worse result, and '+' means the better result than the original algorithm. Furthermore, each algorithm was run 25 times independently for optimizing each function and the average and standard deviation values of the optimal objective function values of these runs were reported in the tables as Mean and Std. Dev. indices, respectively. Moreover, In order to compare the performance of all algorithms in optimizing each test function, the Rank index is used, which shows the rank of each algorithm in the list of sorted Mean indices for each test function. Additionally, *Nb* shows the number of times the considered algorithm obtains the best result among all algorithms and *Mr* is the average of the Rank indices of each algorithm in optimizing all test functions. As is seen from Table 2, algorithms with self-competitive mutations outperform their corresponding original counterparts. The most potent algorithm among the given five algorithms is the MSCjDE algorithm. It can be concluded from this table that the proposed control parameter was effective on DE/rand/1 and jDE algorithms. Moreover, the convergence characteristics of these five algorithms in one example run for the unimodal test function F4 and the multimodal test function F10 are illustrated in Fig. 1.

The results of the Wilcoxon signed-rank test for comparing MSCjDE with some versions of DE/rand/1 are presented in Table 3. In this article, SPR and SNR are assumed as the sum of the positive and negative ranks, respectively, MPR and MNR are considered as the mean of the positive and negative ranks, respectively, F(i) < F(j) shows the number of times the first algorithm outperforms the second one, and F(j) < F(i) demonstrate the opposite. It should be noted that, in the Wilcoxon test, positive ranks are associated with the cases in which

Table 2
The results of DE/rand/1 and SCDE/rand/1 (F = 0.9, CR = 0.9) and jDE, SCjDE, and MSCjDE for the 30-D real-parameter functions.

Function	DE/rand/1 Mean ± Std Dev Rank	SCDE/rand/1 Mean ± Std Dev Rank	jDE Mean ± Std Dev Rank	SCjDE Mean ± Std Dev Rank	MSCjDE Mean ± Std Dev Rank
F1	7.34E−01 ± 4.61E−01 7	1.42E−01 ± 1.07E−01 +, 6	0.00E+00 ± 0.00E+00 1	0.00E+00 ± 0.00E+00 =, 1	0.00E+00 ± 0.00E+00 =, 1
F2	2.21E+03 ± 1.13E+03 14	1.28E+03 ± 7.39E+02 +, 12	8.85E−07 ± 1.15E−06 10	2.60E−07 ± 5.24E−07 +, 9	1.06E−09 ± 3.02E−09 +, 7
F3	1.94E+07 ± 8.54E+06 13	1.10E+07 ± 3.94E+06 +, 12	1.90E+05 ± 1.00E+05 7	1.86E+05 ± 1.24E+05 +, 6	7.37E+04 ± 4.46E+04 +, 4
F4	6.89E+03 ± 3.19E+03 16	5.95E+03 ± 3.28E+03 +, 15	3.42E−02 ± 3.19E−01 8	1.54E−02 ± 3.38E−02 +, 7	8.74E−03 ± 1.61E−02 +, 5
F5	2.33E+03 ± 5.64E+02 16	1.61E+03 ± 6.77E+02 +, 14	4.00E+02 ± 3.35E+02 9	2.48E+02 ± 3.57E+02 +, 8	1.41E+01 ± 6.35E+01 +, 3
F6	6.67E+02 ± 5.42E+02 15	2.68E+02 ± 2.07E+02 +, 13	2.25E+01 ± 2.45E+01 11	9.57E+00 ± 1.69E+01 +, 8	6.54E−01 ± 1.13E+00 +, 2
F7	1.13E+00 ± 1.07E−01 15	9.89E−01 ± 5.00E−02 +, 14	1.03E−02 ± 9.58E−03 6	6.11E−03 ± 6.86E−03 +, 3	5.72E−03 ± 6.19E−03 +, 2
F8	2.09E+01 ± 5.48E−02 1	2.09E+01 ± 3.82E−02 =, 1	2.09E+01 ± 8.01E−02 1	2.09E+01 ± 4.32E−02 =, 1	2.09E+01 ± 4.74E−02 =, 1
F9	8.80E+01 ± 2.16E+01 11	8.47E+01 ± 3.47E+01 +, 10	0.00E+00 ± 0.00E+00 1	0.00E+00 ± 0.00E+00 =, 1	0.00E+00 ± 0.00E+00 =, 1
F10	2.35E+02 ± 1.65E+01 16	2.35E+02 ± 1.59E+01 =, 16	5.75E+01 ± 7.61E+00 7	5.39E+01 ± 1.01E+01 +, 6	3.85E+01 ± 6.22E+00 +, 5
F11	3.90E+01 ± 1.49E+01 12	3.90E+01 ± 1.27E+00 =, 12	2.75E+01 ± 1.83E+00 5	2.75E+01 ± 1.36E+00 =, 5	2.46E+01 ± 1.35E+00 +, 3
F12	4.77E+04 ± 4.22E+04 14	3.16E+04 ± 3.32E+04 +, 9	1.63E+04 ± 87.76E+03 7	1.03E+04 ± 8.26E+03 +, 4	9.14E+03 ± 3.77E+03 +, 3
F13	1.69E+01 ± 2.45E+00 11	1.61E+01 ± 2.10E+00 +, 9	1.71E+00 ± 9.40E−02 2	1.71E+00 ± 1.44E−01 =, 2	1.71E+00 ± 1.34E−01 =, 2
F14	1.34E+01 ± 1.81E−01 7	1.34E+01 ± 1.30E−01 =, 7	1.30E+01 ± 2.10E−01 5	1.30E+01 ± 2.19E−01 =, 5	1.27E+01 ± 2.56E−01 +, 3
F15	4.03E+02 ± 6.47E+01 16	3.68E+02 ± 9.34E+01 +, 12	3.75E+02 ± 9.59E+01 13	3.48E+02 ± 9.538E+01 +, 1	3.68E+02 ± 8.53E+01 +, 2
F16	2.77E+02 ± 3.93E+01 15	2.69E+02 ± 2.66E+01 +, 14	7.94E+01 ± 8.75E+01 2	7.94E+01 ± 2.10E+01 =, 2	7.42E+01 ± 3.37E+01 +, 1
F17	2.97E+02 ± 2.08E+01 18	1.88E+02 ± 2.39E+01 +, 9	1.35E+02 ± 2.33E+01 3	1.67E+02 ± 8.57E+01 −, 7	1.29E+02 ± 9.75E+01 +, 1
F18	9.07E+02 ± 2.81E−01 7	9.07E+02 ± 5.85E−01 =, 7	9.04E+02 ± 1.13E+01 3	9.04E+02 ± 8.40E−01 =, 3	9.04E+02 ± 3.09E−01 =, 3
F19	9.07E+02 ± 2.45E−01 6	9.07E+02 ± 5.58E−01 =, 6	9.04E+02 ± 1.20E+00 4	9.04E+02 ± 9.18E−01 =, 4	9.04E+02 ± 3.06E−01 =, 4
F20	9.07E+02 ± 2.95E−01 5	9.07E+02 ± 4.41E−01 =, 5	9.04E+02 ± 1.15E+00 3	9.04E+02 ± 1.17E+00 =, 3	9.04E+02 ± 1.02E+00 =, 3
F21	5.00E+02 ± 6.42E−02 1	5.00E+02 ± 3.17E−02 =, 1	5.00E+02 ± 8.84E−13 1	5.00E+02 ± 1.11E−13 =, 1	5.00E+02 ± 1.14E−13 =, 1
F22	9.18E+02 ± 1.22E+01 11	9.10E+02 ± 1.14E+01 +, 10	8.75E+02 ± 1.93E+01 6	8.75E+02 ± 1.87E+01 =, 6	8.65E+02 ± 1.95E+01 +, 4
F23	5.34E+02 ± 6.60E−04 1	5.34E+02 ± 7.50E−04 =, 1	5.34E+02 ± 2.90E−04 1	5.34E+02 ± 1.00E−04 =, 1	5.34E+02 ± 1.38E−04 =, 1
F24	2.00E+02 ± 1.56E−01 1	2.00E+02 ± 3.63E−02 =, 1	2.00E+02 ± 2.90E−14 1	2.00E+02 ± 2.90E−14 =, 1	2.00E+02 ± 1.16E−12 =, 1
F25	2.00E+02 ± 2.19E−01 1	2.00E+02 ± 5.31E−12 =, 1	2.00E+02 ± 2.90E−14 1	2.00E+02 ± 1.86E−12 =, 1	2.00E+02 ± 1.48E−12 =, 1
+/ − / =	–	14/0/11	–	9/1/15	14/0/11
Nb/Mr	5/13.889	5/12.056	7/6.556	8/5.333	9/3.556

Table 3
Wilcoxon's test results between MSCjDE and some versions of DE/rand/1.

i	j	MPR	MNR	SPR	SNR	F(i) < F(j)	F(j) < F(i)	p-value	0.95 Confidence interval	
MSCjDE	DE/rand/1	10.500000	NaN	210	0	20	0	9.515546e−05	−3.446496e+03	−34.999996
	SCDE/rand/1	10.000000	NaN	190	0	19	0	1.421330e−04	−2.997496e+03	−29.850079
	jDE	7.500000	NaN	105	0	14	0	1.097051e−03	−3.583000e+03	−3.002327
	SCjDE	7.307692	10.000000	95	10	13	1	8.373849e−03	−5.826000e+02	−1.450004

the first algorithm surpasses the second one. It is obvious from Table 3 that all of the p-values are lower than the significance level of 0.05 and all of the SPR values are higher than SNR values. Therefore, it is concluded that MSCjDE is significantly superior to other tested variants of DE/rand/1.

4.2. SCDE/current-to-pbest/1 and SCJADE

In this section, the effect of the proposed strategy on the DE/current-to-pbest/1 algorithm and the improved DE algorithm known as JADE [19], which is an adaptive and modified version of the DE/current-to-pbest/1 algorithm, is analyzed. Codes of the JADE algorithm are extracted from (http://dces.essex.ac.uk/staff/qzhang/) with the same control parameters. The self-competitive versions of DE/current-to-pbest/1 and JADE are called SCDE/current-to-pbest/1 and SCJADE, respectively. The only change made on the improved versions is using the control parameter coefficient ω^{12}_{Iter}, and their other conditions are identical to those of their original counterparts. The values of F = 0.5, CR = 0.5, and NP = 60 are chosen for DE/current-to-pbest/1 and SCDE/current-to-pbest/1 algorithms. The mutation equation for SCDE/current-to-pbest/1 and SCJADE algorithms is given in Eq. (12).

$$v^{i}_{j,Iter} = x^{i}_{j,Iter} + F\left(x_{best,j} - x^{i}_{j,Iter}\right) + \omega^{12}_{Iter} \times F\left(x^{r_1}_{j,Iter} - x^{r_2}_{j,Iter}\right) \tag{12}$$

The results of the algorithms given in this section are listed in Table 4. One can easily see from Table 4 that SCDE/current-to-pbest/1 outperforms its original algorithm for 16 functions and loses at only two functions. Also, the self-competitive strategy was able to improve the performance of the JADE algorithm and SCJADE algorithms have a weaker performance than JADE for just one test function. In addition, the convergence characteristics of these five algorithms in one example run for Unimodal test function F4 and Multimodal test function F10 are illustrated in Fig. 2.

The results of the Wilcoxon signed-rank test for comparing SCJADE with some versions of DE/current-to-pbest/1 are presented in Table 5. It is seen from Table 5 that the p-values for comparing SCJADE with DE/current-to-pbest/1 and SCDE/current-to-pbest/1 are lower than the significance level and their SPR values are higher than SNR values, which shows the significance and superiority of SCJADE over these two algorithms. When comparing SCJADE with JADE, the p-value is higher than 0.05. Nevertheless, since the SPR value is greater than the SNR value, it is concluded that SCJADE outperforms JADE, but not significantly.

4.3. SCDE/best/1, SCDE/current-to-rand/1 and SCDE/rand/2

In this section, the performance of the proposed improvement on the original DE, DE/best/1, DE/current-to-rand/1, and DE/rand/2 algorithms is tested. The values of F and CR for different algorithms are as follows: F = 0.7 and CR = 0.9 for DE/best/1 and SCDE/best/1 algorithms, F = 0.8 and CR = 0.2 for DE/current-to-rand/1 and SCDE/current-to-rand/1 algorithms, and F = 0.45 and CR = 0.7 for DE/rand/2 and SCDE/rand/2 algorithms. Furthermore, NP = 60 is chosen for all algorithms.

Based on the investigations, SCDE/current-to-rand/1 works better without considering the parameter ω^{i1}_{Iter}, as in (14).

"SCDE/best/1":

$$V^{i}_{Iter} = X_{best} + \omega^{12}_{Iter} \times F\left(X^{r_1}_{Iter} - X^{r_2}_{Iter}\right) \tag{13}$$

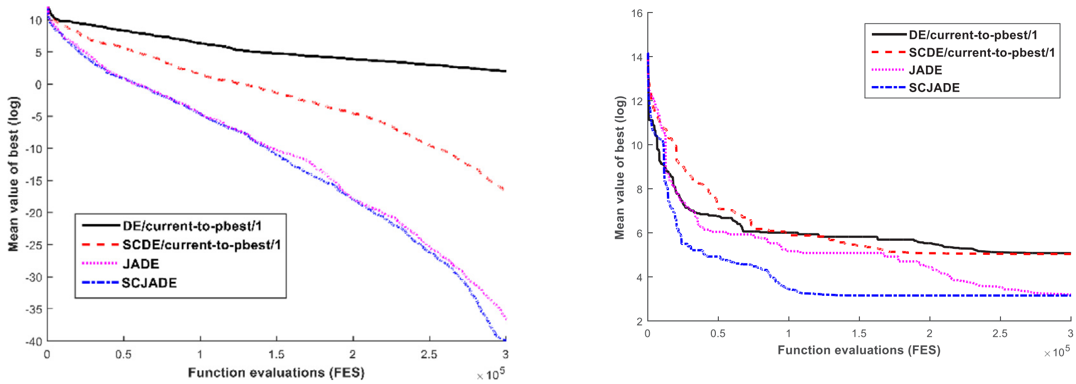


Fig. 2. The convergence characteristics of the algorithms for Unimodal test function F4 and Multimodal test function F10.

Table 4
The results of DE/current-to-pbest/1, SCDE/current-to-pbest/1, and JADE variants (F = 0.5, CR = 0.5) for the 30-D real-parameter functions.

Function	DE/current-to-pbest/1 Mean ± Std Dev Rank	SCDE/current-to-pbest/1 Mean ± Std Dev Rank	JADE Mean ± Std Dev Rank	SCJADE Mean ± Std Dev Rank
F1	2.38E+00 ± 4.76E+00 8	3.70E−22 ± 1.84E−21 +, 3	0.00E+00 ± 0.00E+00 1	0.00E+00 ± 0.00E+00 =, 1
F2	1.83E+02 ± 1.80E+02 11	2.29E−07 ± 9.24E−07 +, 8	1.00E−28 ± 1.05E−28 1	1.00E−28 ± 1.43E−28 =, 1
F3	1.45E+06 ± 6.97E+05 9	2.10E+06 ± 8.64E+05 −, 11	5.62E+03 ± 3.62E+03 1	5.62E+03 ± 4.20E+03 =, 1
F4	7.72E+00 ± 1.18E+01 11	5.61E−10 ± 2.25E−09 +, 3	1.03E−15 ± 3.41E−15 2	4.80E−18 ± 1.61E−17 +, 1
F5	2.19E+03 ± 4.77E+02 15	1.07E+03 ± 5.85E+02 +, 10	6.60E−08 ± 2.27E−07 2	8.63E−11 ± 2.22E−10 +, 1
F6	3.10E+05 ± 6.57E+05 17	5.32E+02 ± 9.22E+02 +, 14	1.03E+01 ± 2.91E+01 9	1.03E+01 ± 4.20E+01 =, 9
F7	1.79E+01 ± 1.21E+01 17	3.82E+00 ± 2.54E+00 +, 16	6.99E−03 ± 7.57E−03 2	1.80E−03 ± 3.69E−03 +, 1
F8	2.09E+01 ± 8.41E−02 1	2.09E+01 ± 3.00E−02 =, 1	2.09E+01 ± 1.75E−01 1	2.09E+01 ± 3.75E−02 =, 1
F9	1.96E+01 ± 1.10E+01 2	1.96E+01 ± 1.23E+01 =, 2	0.00E+00 ± 0.00E+00 1	0.00E+00 ± 0.00E+00 =, 1
F10	1.61E+02 ± 1.52E+01 11	1.55E+02 ± 1.16E+01 +, 10	2.47E+01 ± 5.78E+00 3	2.35E+01 ± 2.96E+00 +, 1
F11	3.60E+01 ± 1.91E+00 11	3.46E+01 ± 3.00E+00 +, 8	2.52E+01 ± 1.48E+00 4	2.52E+01 ± 1.55E+00 =, 4
F12	1.62E+04 ± 1.08E+04 6	1.57E+04 ± 1.37E+04 +, 5	5.28E+03 ± 3.90E+03 2	5.20E+03 ± 6.08E+03 +, 1
F13	9.44E+00 ± 8.80E−01 8	9.44E+00 ± 9.26E−01 =, 8	1.45E+00 ± 1.32E−01 1	1.45E+00 ± 1.17E−01 =, 1
F14	1.29E+01 ± 2.11E−01 4	1.29E+01 ± 2.72E−01 =, 4	1.23E+01 ± 2.69E−01 2	1.23E+01 ± 2.37E−01 =, 2
F15	3.55E+02 ± 1.17E+02 10	3.85E+02 ± 1.09E+02 −, 14	3.44E+02 ± 1.12E+02 7	3.44E+02 ± 9.61E+02 =, 7
F16	3.03E+02 ± 1.28E+02 17	2.44E+02 ± 1.08E+02 +, 11	8.96E+01 ± 9.92E+01 3	1.30E+02 ± 1.52E+02 −, 5
F17	2.82E+02 ± 1.04E+02 16	2.71E+02 ± 1.03E+02 +, 15	1.46E+02 ± 1.31E+02 5	1.36E+02 ± 1.53E+02 +, 4
F18	9.26E+02 ± 3.04E+01 11	9.10E+02 ± 3.37E+01 +, 9	9.04E+02 ± 7.46E−01 3	9.04E+02 ± 7.53E−01 =, 3
F19	9.15E+02 ± 4.87E+01 8	9.15E+02 ± 2.63E+01 =, 8	9.04E+02 ± 2.05E−01 4	9.04E+02 ± 7.33E−01 =, 4
F20	9.20E+02 ± 3.19E+01 9	9.20E+02 ± 7.20E+01 =, 9	9.04E+02 ± 3.34E−01 3	9.04E+02 ± 9.75E−01 =, 3
F21	5.99E+02 ± 1.95E+02 5	5.63E+02 ± 1.58E+02 +, 4	5.00E+02 ± 4.46E−14 1	5.00E+02 ± 1.74E−12 =, 1
F22	9.27E+02 ± 2.16E+01 13	9.03E+02 ± 1.56E+01 +, 9	8.70E+02 ± 1.83E+01 5	8.65E+02 ± 1.94E+01 +, 4
F23	7.28E+02 ± 2.07E+02 5	6.33E+02 ± 1.69E+02 +, 2	5.34E+02 ± 3.27E−13 1	5.34E+02 ± 3.63E−13 =, 1
F24	2.05E+02 ± 8.39E+00 2	2.05E+02 ± 6.00E+01 =, 2	2.00E+02 ± 2.90E−14 1	2.00E+02 ± 2.88E−14 =, 1
F25	2.09E+02 ± 2.40E+01 2	2.00E+02 ± 4.47E−01 +, 1	2.00E+02 ± 2.90E−14 1	2.00E+02 ± 2.90E−14 =, 1
+/ − / =	–	16/2/7	–	7/1/17
Nb/Mr	1/12.722	2/10.389	10/3.667	15/3.333

Table 5
Wilcoxon's test results between the SCJADE and some version of DE/current-to-pbest/1.

i	j	MPR	MNR	SPR	SNR	F(i) < F(j)	F(j) < F(i)	p-value	0.95 Confidence interval	
	DE/current-to-pbest/1	12.500000	NaN	300	0	24	0	1.940350e−05	−1102.999960	−16.49997
SCJADE	SCDE/current-to-pbest/1	12.000000	NaN	276	0	23	0	2.888847e−05	−279.850028	−14.49995
	JADE	4.142857	7.0000	29	7	7	1	1.414821e−01	−40.599929	20.19739

"SCDE/current-to-rand/1":

$$V^i_{Iter} = X^i_{Iter} + rand(0,1) \times \left(X^i_{Iter} - X^{r_1}_{Iter}\right) \\ +\omega^{23}_{Iter} \times rand(0,1) \times F\left(X^{r_2}_{Iter} - X^{r_3}_{Iter}\right) \tag{14}$$

"SCDE/rand/2":

$$V^i_{Iter} = X^{r_1}_{Iter} + \omega^{23}_{Iter} \times F\left(X^{r_2}_{Iter} - X^{r_3}_{Iter}\right) + \omega^{45}_{Iter} \times F\left(X^{r_4}_{Iter} - X^{r_5}_{Iter}\right) \tag{15}$$

The results of all algorithms are given in Table 6. Also, the convergence characteristics of these four algorithms for Unimodal test function F4 and Multimodal test function F10 are shown in Fig. 3. In general, it can be said that the proposed control parameter has no sensible positive effect on DE/best algorithms. However, it can be concluded from Table 6 that it has a positive and acceptable impact on DE/rand/2 and DE/current-to-rand/1 algorithms and has converted them into more robust SCDE/rand/2 and SCDE/current-to-rand/1 algorithms.

The results of the Wilcoxon signed-rank test for comparing SCDE/best/1 with some other versions of DE are presented in Table 7. However, except for the first row of Table 7, since the SPR values are greater than the SNR values, it is concluded that SCDE/best/1 outperforms DE/current-to-rand/1, SCDE/current-to-rand/1, DE/rand/2, and SCDE/rand/2, but not significantly. For the comparison of SCDE/best/1 with DE/best/1, the p-value is relatively high and the SPR and SNR values are very close together. It can be concluded that the performance of these two algorithms in solving 25 CEC2005 benchmark functions is very close together.

4.4. SCEPSDE

In this section, the proposed control parameter is tested on EPSDE [28]. The mutations of this algorithm are comprised of mutations

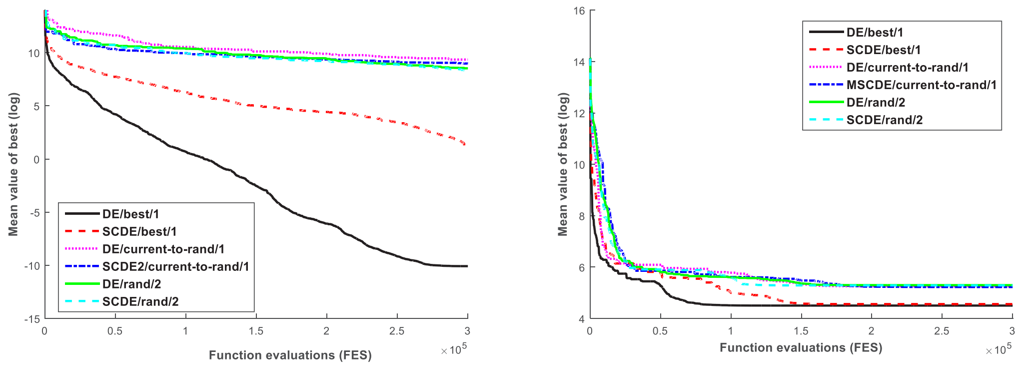


Fig. 3. The convergence characteristics of the algorithms for Unimodal test function F4 and Multimodal test function F10.

Table 6
The results of DE/best/1 and SCDE/best/1 ($F = 0.7$, $CR = 0.9$), DE/current-to-rand/1and SCDE/current-to-rand/1 ($F = 0.8$, $CR = 0.2$), and DE/rand/2 and SCDE/rand/2 ($F = 0.45$, $CR = 0.7$) for the 30-D real-parameter functions.

Function	DE/best/1 Mean ± Std Dev Rank	SCDE/best/1 Mean ± Std Dev Rank	DE/current-to-rand/1 Mean ± Std Dev Rank	SCDE/current-to-rand/1 Mean ± Std Dev Rank	DE/rand/2 Mean ± Std Dev Rank	SCDE/rand/2 Mean ± Std Dev Rank
F1	0.00E+00 ± 0.00E+00 1	0.00E+00 ± 0.00E+00 =, 1	0.00E+00 ± 0.00E+00 1	0.00E+00 ± 0.00E+00 =, 1	7.58E−21 ± 4.10E−21 5	2.70E−21 ± 1.44E−21 +, 4
F2	4.25E−28 ± 4.90E−28 3	3.87E−28 ± 6.75E−28 =, 2	3.79E+03 ± 6.55E+02 17	3.03E+03 ± 5.14E+02 +, 16	2.86E+03 ± 5.52E+02 15	1.87E+03 ± 6.26E+02 +, 13
F3	4.29E+04 ± 2.17E+04 2	6.30E+04 ± 2.23E+04 −, 3	3.81E+07 ± 7.84E+06 14	3.81E+07 ± 6.08E+06 =, 14	6.68E+07 ± 1.46E+07 16	6.20E+07 ± 1.27E+07 +, 15
F4	4.20E−05 ± 1.27E−04 4	3.36E+00 ± 2.14E+01 −, 10	1.15E+04 ± 2.09E+03 18	7.87E+03 ± 1.53E+03 +, 17	5.05E+03 ± 9.63E+02 14	4.20E+03 ± 8.77E+02 +, 13
F5	1.20E+02 ± 1.32E+02 6	3.90E+01 ± 7.00E+01 +, 5	4.13E+03 ± 3.69E+02 18	3.61E+03 ± 3.60E+02 +, 17	1.89E+02 ± 1.09E+02 7	2.32E+01 ± 2.02E+01 +, 4
F6	1.44E+00 ± 1.95E+00 5	7.97E−01 ± 1.63E+00 +, 3	2.87E+01 ± 1.69E+00 12	2.72E+03 ± 1.56E+00 +, 16	1.08E+01 ± 1.29E+00 10	5.29E+00 ± 8.94E−01 +, 7
F7	2.10E−02 ± 2.42E−02 12	1.72E−02 ± 1.13E−02 +, 9	2.61E−01 ± 4.09E−02 13	2.61E−01 ± 4.95E−02 =, 13	7.98E−03 ± 5.32E−03 5	1.79E−02 ± 4.60E−02 −, 10
F8	2.09E+01 ± 4.66E−02 1	2.09E+01 ± 6.28E−02 =, 1	2.09E+01 ± 6.02E−02 1	2.09E+01 ± 4.29E−02 =, 1	2.09E+01 ± 6.82E−02 1	2.09E+01 ± 5.33E−02 =, 1
F9	6.24E+01 ± 1.88E+01 8	6.61E+01 ± 2.20E+01 −, 9	3.43E+01 ± 3.63E+00 4	3.32E+01 ± 3.12E+00 +, 3	1.47E+02 ± 9.32E+00 12	1.47E+02 ± 5.85E+00 =, 12
F10	9.03E+01 ± 3.41E+01 8	9.56E+01 ± 3.04E+01 −, 9	1.92E+02 ± 1.02E+01 13	1.86E+02 ± 1.13E+01 +, 12	2.00E+02 ± 1.08E+01 15	1.98E+02 ± 1.04E+01 +, 14
F11	1.40E+01 ± 2.51E+00 2	1.33E+01 ± 2.55E+00 +, 1	3.42E+01 ± 1.28E+00 7	3.42E+01 ± 1.70+00 =, 7	4.00E+01 ± 8.67E−01 14	3.95E+01 ± 9.75E−01 +, 13
F12	3.32E+04 ± 4.28E+04 10	2.18E+04 ± 2.06E+04 +, 8	1.01E+05 ± 1.53E+04 16	9.20E+04 ± 1.49E+04 +, 15	2.61E+05 ± 9.34E+04 18	1.25E+05 ± 1.01E+05 +, 17
F13	5.62E+00 ± 1.84E+00 4	6.34E+00 ± 1.71E+00 −, 5	7.59E+00 ± 4.91E−01 7	7.40E+00 ± 5.81E−01 +, 6	1.63E+01 ± 1.28E+00 10	1.63E+01 ± 9.14E−01 =, 10
F14	1.21E+01 ± 5.40E−01 1	1.21E+01 ± 4.10E−01 =, 1	1.32E+01 ± 1.44E−01 6	1.32E+01 ± 1.20E−01 =, 6	1.34E+01 ± 9.59E−02 7	1.34E+01 ± 1.08E−01 =, 7
F15	3.86E+02 ± 7.87E+01 15	3.66E+02 ± 9.65E+01 +, 11	2.44E+02 ± 2.36E+01 5	2.24E+02 ± 3.04E+01 +, 3	3.52E+02 ± 8.23E+01 9	3.28E+02 ± 7.92E+01 +, 6
F16	2.20E+02 ± 1.51E+02 8	2.84E+02 ± 1.63E+02 −, 16	2.50E+02 ± 1.64E+01 13	2.37E+02 ± 2.12E+01 +, 10	2.31E+02 ± 4.52E+01 9	2.46E+02 ± 6.87E+01 −, 12
F17	2.01E+02 ± 1.52E+02 10	2.19E+02 ± 1.51E+02 −, 11	2.94E+02 ± 1.96E+01 17	2.69E+02 ± 3.45E+01 +, 14	2.48E+02 ± 2.82E+01 12	2.64E+02 ± 5.82E+01 −, 13
F18	9.13E+02 ± 1.09E+01 10	9.09E+02 ± 2.57E+00 +, 8	9.09E+02 ± 5.13E−01 8	9.09E+02 ± 3.95E−01 =, 8	9.06E+02 ± 1.54E−01 4	9.06E+02 ± 2.05E−01 =, 4
F19	9.09E+02 ± 3.80E+00 7	9.09E+02 ± 2.35E+01 =, 7	9.09E+02 ± 5.07E−01 7	9.09E+02 ± 4.08E−01 =, 7	9.06E+02 ± 1.82E−01 5	9.06E+02 ± 1.80E−01 =, 5
F20	9.11E+02 ± 5.35E+00 8	9.08E+02 ± 1.57E+01 +, 6	9.09E+02 ± 3.96E−01 7	9.09E+02 ± 5.34E−01 =, 7	9.06E+02 ± 1.57E−01 4	9.06E+02 ± 5.77E−01 =, 4
F21	5.51E+02 ± 1.51E+02 3	5.28E+02 ± 9.80E+01 +, 2	5.00E+02 ± 1.28E−13 1	5.00E+02 ± 1.08E−13 =, 1	5.00E+02 ± 1.35E−13 1	5.00E+02 ± 1.44E−13 =, 1
F22	9.19E+02 ± 2.43E+01 12	9.19E+02 ± 2.84E+01 =, 12	9.46E+02 ± 9.55E+00 15	9.38E+02 ± 6.97E+00 +, 14	8.86E+02 ± 1.03E+01 8	8.76E+02 ± 9.66E+00 +, 7
F23	6.57E+02 ± 1.35E+00 3	7.05E+02 ± 1.98E+02 −, 4	5.34E+02 ± 1.72E−04 1	5.34E+02 ± 1.72E−04 =, 1	5.34E+02 ± 3.06E−04 1	5.34E+02 ± 2.77E−04 =, 1
F24	2.62E+02 ± 2.14E+00 5	3.06E+02 ± 2.62E+02 −, 6	2.00E+02 ± 1.35E−12 1	2.00E+02 ± 1.16E−12 =, 1	2.00E+02 ± 1.56E−12 1	2.00E+02 ± 1.17E−12 =, 1
F25	3.64E+02 ± 3.38E+02 6	3.35E+02 ± 2.84E+02 +, 5	2.00E+02 ± 1.43E−12 1	2.00E+02 ± 8.63E−13 =, 1	2.00E+02 ± 1.60E−12 1	2.00E+02 ± 1.68E−12 =, 1
+/ − / =	–	10/9/6	–	12/0/13	–	**11/3/11**
Nb/Mr	3/8.556	4/8.611	6/12.389	6/11.722	5/11.889	5/10.833

Table 7
Wilcoxon's test results between the SCDE/best/1 and some modified versions of DE.

i	j	MPR	MNR	SPR	SNR	F(i) < F(j)	F(j) < F(i)	p-value	0.95 Confidence interval	
SCDE/best/1	DE/best/1	9.1818	12.1111	101	109	11	9	0.8960408	−12.65003	22.00001
	DE/current-to-rand/1	10.4286	12.1429	146	85	14	7	0.2970736	−5680.81999	27.99998
	SCDE/current-to-rand/1	10.9286	11.1429	153	78	14	7	0.1984315	−3862.31997	23.37804
	DE/rand/2	13.9231	10.8182	181	119	13	11	0.3834959	−1428.99997	13.99999
	SCDE/rand/2	13	11.9091	169	131	13	11	0.5970654	−933.50007	19.25345

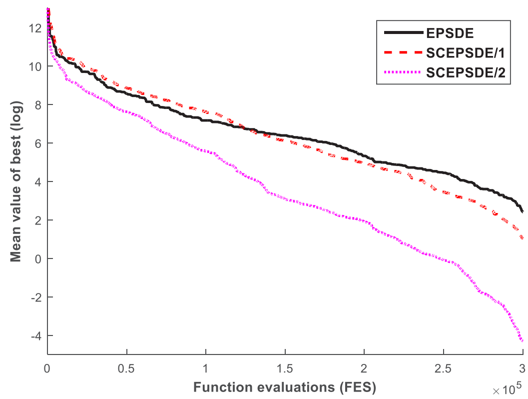
of DE/rand/2, DE/rand/1, and DE/current-to-rand/1, which are also selected for use in self-competitive EPSDE (SCEPSDE). The selected code for the EPSDE algorithm is the same code with the same chosen parameters as in (http://dces.essex.ac.uk/staff/qzhang/).

In simulations, two algorithms of SCEPSDE known as SCEPSDE/1 and SCEPSDE/2 are obtained. The SCEPSDE/1 is identical to the original EPSDE algorithm in all respects except for the DE/current-to-rand/1 mutation which is substituted by its self-competitive counterpart, given in (14), is used. Also, SCEPSDE/2 is exactly similar to the original EPSDE algorithm except for the DE/rand/2 mutation which is substituted by its self-competitive counterpart, given in (15). The results of all algorithms are given in Table 8. Also, the convergence characteristics of these three algorithms for Unimodal test function F4 and Multimodal test function F10 are shown in Fig. 4. The proposed control parameter has a positive impact on the EPSDE algorithm and both SCEPSDE/2 and SCEPSDE/1 outperform EPSDE; from which its SCEPSDE/2 shows the best performance.

The results of the Wilcoxon signed-rank test for comparing SCEPSDE/2 with some versions of EPSDE and SCEPSDE/1 are presented in Table 9. It is seen from Table 9 that the p-value for comparing SCEPSDE/2 with EPSDE is lower than the significance level and the SPR value is higher than the SNR value, which shows the significance and superiority of SCEPSDE/2 over EPSDE. When comparing SCEPSDE/2 with SCEPSDE/1, the p-value is higher relatively high and the SPR and SNR values are close together. It can be concluded that the performance of these two algorithms in solving 25 CEC2005 benchmark functions is close together.

4.5. A global comparison between different variants of DE using the friedman test

In order to compare the performance of all studied algorithms the Friedman test was carried out, and its results were presented in Table 10. In this table, the Mean Rank index is the average of the Rank indices of the algorithm for all test functions and the RankT index shows the rank of each algorithm in the list of sorted Mean Rank indices. It is observed from this table that, each self-competitive variant of DE outperforms its original counterpart. Furthermore, SCJADE outperforms all other algorithms in terms of the RankT index. The p-value



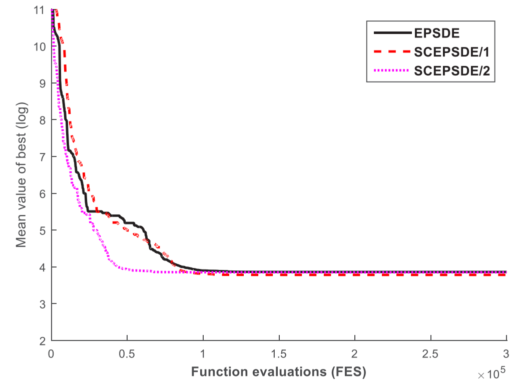


Fig. 4. The convergence characteristics of the algorithms for Unimodal test function F4 and Multimodal test function F10.

Table 8
The results of EPSDE algorithms on the 30-*D* real-parameter functions.

Function	EPSDE Mean Std Dev, Rank	SCEPSDE/1 Mean Std Dev, Rank	SCEPSDE/2 Mean Std Dev, Rank
F1	0.00E+00 ± 0.00E+00 1	2.02E−30 ± 1.00E−29- −, 2	0.00E+00 ± 0.00E+00 =, 1
F2	4.95E−15 ± 2.19E−14 6	3.18E−26 ± 3.66E−26 +, 4	4.01E−26 ± 4.96E−26 +, 5
F3	2.06E+06 ± 6.16E+06 10	1.11E+06 ± 4.32E+05 −, 8	1.80E+05 ± 3.66E+05 +, 5
F4	1.10E+01 ± 3.06E+01 12	2.18E+00 ± 7.86E+01 +, 9	1.37E−02 ± 5.98E−02 +, 6
F5	1.33E+03 ± 6.90E+02 13	1.15E+03 ± 5.19E+03 +, 11	1.27E+03 ± 6.91E+02 +, 12
F6	9.57E−01 ± 1.74E+00 4	3.19E−01 ± 1.10E+00 +, 1	3.36E+00 ± 1.44E+01 −, 6
F7	1.92E−02 ± 1.64E−02 11	1.07E−02 ± 9.58E−03 +, 7	1.18E−02 ± 1.11E−02 +, 8
F8	2.09E+01 ± 5.88E−02 1	2.09E+01 ± 6.18E−02 =, 1	2.09E+01 ± 7.20E−02 =, 1
F9	0.00E+00 ± 0.00E+00 1	0.00E+00 ± 0.00E+00 =, 1	0.00E+00 ± 0.00E+00 =, 1
F10	4.76E+01 ± 9.33E+00 7	4.39E+01 ± 9.28E+00 +, 5	4.71E+01 ± 9.69E+00 +, 6
F11	3.53E+01 ± 3.83E+00 9	3.57E+01 ± 3.00E+00 −, 10	3.41E+01 ± 4.22E+00 +, 6
F12	3.61E+04 ± 5.95E+03 13	3.44E+04 ± 6.83E+03 +, 11	3.46E+04 ± 7.23E+03 +, 12
F13	1.94E+00 ± 2.05E−01 3	1.94E+00 ± 1.69E−01 =, 3	1.94E+00 ± 1.99E−01 =, 3
F14	1.35E+01 ± 2.84E−01 8	1.35E+01 ± 2.77E−01 =, 8	1.35E+01 ± 2.83E−01 =, 8
F15	2.26E+02 ± 4.58E+01 4	2.23E+02 ± 4.62E+01- +, 2	2.08E+02 ± 7.32E+00 +, 1
F16	1.59E+02 ± 1.40E+02 6	1.82E+02 ± 1.51E+01 −, 7	1.29E+02 ± 1.13E+02 +, 4
F17	1.60E+02 ± 6.58E+01 6	1.34E+02 ± 7.81E+01 +, 2	1.86E+02 ± 1.12E+02 −, 8
F18	8.20E+02 ± 3.51E+00 2	8.18E+02 ± 2.42E+00 +, 1	8.18E+02 ± 1.75E+00 +, 1
F19	8.22E+02 ± 4.48E+00 3	8.18E+02 ± 1.28E+00 +, 2	8.17E+02 ± 1.54E+00 +, 1
F20	8.21E+02 ± 3.55E+00 2	8.17E+02 ± 1.05E+00 +, 1	8.17E+02 ± 1.09E+00 +, 1
F21	8.49E+02 ± 7.27E+01 6	8.49E+02 ± 7.25E+01 =, 6	8.63E+02 ± 4.27E+00 −, 7
F22	5.19E+02 ± 5.42E+01 3	5.03E+02 ± 3.82E+00 +, 2	5.02E+02 ± 3.67E+00 +, 1
F23	8.55E+02 ± 9.31E+01 7	8.44E+02 ± 8.83E+01 +, 6	8.69E+02 ± 3.51E+00 −, 8
F24	2.13E+02 ± 1.76E+00 4	2.12E+02 ± 1.69E+00= 3	2.13E+02 ± 1.04E+00 =, 4
F25	2.13E+02 ± 1.72E+00 4	2.11E+02 ± 9.27E−01 +, 3	2.11E+02 ± 1.26E+00 +, 3
+/ − / =	–	15/5/5	15/5/5
Nb/*Mr*	3/8.111	5/6.444	8/6.611

Table 9
Wilcoxon's test results between SCEPSDE/2 and other versions of EPSDE.

i	j	MPR	MNR	SPR	SNR	F(i) < F(j)	F(j) < F(i)	*p*-value	0.95 Confidence interval	
SCEPSDE/2	EPSDE	9.666667	11.25000	145	45	15	4	0.046328135	−30.999990	−0.0000297043
	SCEPSDE/1	8.875000	10.00000	71	100	8	10	0.541865596	−7.000067	26.0000878920

calculated using the Friedman test is 2.2187e−23 which shows that the performance of different variants of DE differs significantly.

4.6. A comparison with some modern algorithms

In order to further investigate the effectiveness of the proposed self-competitive strategy, the results of the most promising version of self-competitive DE, i.e., SCJADE, along with those of some of the state-of-the-art methods are presented in Table 11. It is obvious from Table 11 that, SCJADE outperforms all other methods in optimizing CEC2005 benchmark functions.

The results of the Wilcoxon signed-rank test for comparing SCJADE with some state-of-the-art methods are presented in Table 12. It can be observed from Table 12 that the p-values for comparing SCJADE with BES and GPEAed are lower than the significance level and their SPR values are greater than SNR values, which demonstrates the significance and superiority of SCJADE over these two algorithms. When comparing SCJADE with FMPSO, the *p*-value is slightly higher than 0.05. Nevertheless, since the SPR value is greater than the SNR value, it is concluded that SCJADE outperforms FMPSO, but not significantly. Finally, while comparing SCJADE with HCLPSO, the *p*-value is high and SPR and SNR values are close together. The ranking of these

Table 10
Average ranking of different DE algorithms and their self-competitive counterparts according to the Friedman test.

Algorithm	RankT	Mean rank
SCJADE	1	4.46
JADE	2	4.76
MSCjDE	3	5.02
SCjDE	4	6.46
jDE	5	7.14
SCEPSDE/2	6	7.92
SCEPSDE/1	7	8
EPSDE	8	9.08
SCDE/best/1	9	10.04
DE/best/1	10	10.12
SCDE/rand/2	11	11.16
DE/rand/2	12	11.6
SCDE/current-to-pbest/1	13	11.8
SCDE/current-to-rand/1	14	11.8
SCDE/rand/1	15	12.14
DE/current-to-rand/1	16	12.28
DE/rand/1	17	13.52
DE/current-to-pbest/1	18	13.7

algorithms can be extracted using the Friedman rank test. Table 13 demonstrates the results of the Friedman rank test. It is obvious from the results presented in these tables that the proposed SCJADE has the best average rank and thus, outperforms all other methods in solving the CEC2005 benchmark functions.

5. Application of SCDE algorithms in designing the optimal PID controller for AVR system

The frequency of the power system is mainly dependent on the active power, while the magnitude of the voltages and the reactive power are mainly interdependent. Hence, the frequency and voltage of the power systems can be controlled separately. The Load Frequency Control (LFC) loop [62,63] controls the system's frequency by adjusting the active power and the Automatic Voltage Regulator (AVR) loop controls the reactive power and voltage magnitudes [64,65].

The proper operation of LFC and AVR systems guarantees the quality and stability of the power system. The AVR system is widely used for synchronous generators (SGs) to achieve proper voltage stability under different operating conditions of the generator; this task is accomplished by controlling the current of the excitation system whose controller is mainly of PID type. The performance of this controller depends on the optimal tuning of its coefficients, i.e., k_P, k_I, and, k_D. Recent advances in the field of optimization and metaheuristic algorithms have enabled designers to achieve optimal design and control of energy systems.

5.1. Structure of PID controller

Proportional–integral–derivative (PID) controller is widely used in practical applications due to its simple and meanwhile efficient structure compared to advanced controllers [66]. This controller can be efficiently applied to many processes with the first or second order of dynamics, whose operational range is not wide [66]. Although this controller consists of three parameters to be identified, many methods have been proposed to find the optimal values of these parameters during the last few decades since the presentation of the Ziegler–Nichols method [67]. Furthermore, the microscopic controller (MIC) has been used to design the PID controller [68], which is suitable for systems with small changes around the operating point and small values of the modeling error; however, this method may cause zero-pole cancellation for which a proper response from the controller is not expected. For highly nonlinear and time-varying processes, neither the application of the phase and gain margin concepts (frequency response) is suitable [69], nor is applying corrective methods to improve their performance for a wide range of the operational region [70]. In this paper, the PID controller parameters in an AVR system are optimally identified and tuned using the proposed algorithms.

Table 11
The comparison of the results of SCJADE with those of some of the state of the art methods.

Function	FMPSO [58] Mean ± Std Dev Rank	HCLPSO [59] Mean ± Std Dev Rank	BES [60] Mean ± Std Dev Rank	GPEAed [61] Mean ± Std Dev Rank	SCJADE Mean ± Std Dev Rank
F1	0.00E+00 ± 0.00E+00 =	0.00E+00 ± 0.00E+00 =	2.54E−13 ± 9.64E−14 +	2.28E−19 ± 1.23E−18 +	0.00E+00 ± 0.00E+00 –
F2	4.37E−25 ± 4.65E−25 +	1.70E−06 ± 1.71E−06 +	3.58E−04 ± 5.58E−04 +	4.29E+00 ± 3.78E+00 +	1.00E−28 ± 1.43E−28 –
F3	1.10E+06 ± 7.34E+05 +	6.42E+05 ± 2.61E+05 +	4.25E+05 ± 1.78E+05 +	6.78E+05 ± 2.37E+05 +	5.62E+03 ± 4.20E+03 –
F4	1.43E+03 ± 6.75E+02 +	5.22E+02 ± 3.09E+02 +	1.18E+03 ± 9.03E+02 +	1.12E+04 ± 5.77E+03 +	4.80E−18 ± 1.61E−17 –
F5	4.17E+03 ± 1.25E+03 +	2.97E+03 ± 4.55E+02 +	3.81E+03 ± 7.51E+02 +	3.19E+03 ± 7.24E+02 +	8.63E−11 ± 2.22E−10 –
F6	9.44E−01 ± 1.42E+00 −	2.39 ± 4.27 −	1.46E+01 ± 1.18E+01 +	6.03E+01 ± 6.41E+01 +	1.03E+01 ± 4.20E+01 –
F7	1.29E−02 ± 9.41E−03 +	0.02 ± 0.02 +	1.78E−02 ± 2.03E−02 +	4.70E+03 ± 3.67E−01 +	1.80E−03 ± 3.69E−03 –
F8	1.97E+01 ± 1.31E+00 −	20.87 ± 0.09 −	2.10E+01 ± 5.01E−02 +	2.10E+01 ± 6.22E−02 +	2.09E+01 ± 3.75E−02 –
F9	3.50E+01 ± 7.50E+00 +	0.00E+00 ± 0.00E+00 =	9.64E+01 ± 2.74E+01 +	7.34E+01 ± 1.79E+01 +	0.00E+00 ± 0.00E+00 –
F10	4.75E+01 ± 5.01E+00 +	56.08 ± 12.90 +	1.25E+02 ± 4.56E+01 +	1.27E+02 ± 3.31E+01 +	2.35E+01 ± 2.96E+00 –
F11	1.90E+01 ± 2.69E+00 −	20.32 ± 2.94 −	2.67E+01 ± 5.55E+00 +	3.91E+01 ± 8.91E+00 +	2.52E+01 ± 1.55E+00 –
F12	2.12E+04 ± 1.65E+04 +	3.91E+03 ± 3.69E+03 −	6.92E+03 ± 9.40E+03 +	1.00E+06 ± 2.47E+05 +	5.20E+03 ± 6.08E+03 –
F13	2.74E+00 ± 8.13E−01 +	1.45 ± 0.28 =	8.49E+00 ± 3.67E+00 +	1.16E+01 ± 5.57E+00 +	1.45E+00 ± 1.17E−01 –
F14	1.19E+01 ± 4.24E−01 −	11.93 ± 0.58 −	1.27E+01 ± 2.26E−01 +	1.37E+01 ± 3.73E−01 +	1.23E+01 ± 2.37E−01 –
F15	3.18E+02 ± 9.02E+01 −	88.04 ± 113.02 −	4.25E+02 ± 9.25E+01 +	5.01E+02 ± 2.94E+00 +	3.44E+02 ± 9.61E+02 –
F16	1.04E+02 ± 2.30E+01 −	104.07 ± 36.41 −	3.50E+02 ± 1.47E+02 +	2.94E+02 ± 1.29E+02 +	1.30E+02 ± 1.52E+02 –
F17	1.65E+02 ± 6.52E+01 +	109.59 ± 34.01 −	2.61E+02 ± 1.59E+02 +	3.00E+02 ± 1.59E+02 +	1.36E+02 ± 1.53E+02 –
F18	8.00E+02 ± 0.00E+00 −	894.42 ± 43.04 −	9.34E+02 ± 3.63E+01 +	9.04E+02 ± 3.61E+01 =	9.04E+02 ± 7.53E−01 –
F19	8.00E+02 ± 0.00E+00 −	913.49 ± 2.39 +	9.35E+02 ± 5.07E+02 +	8.94E+02 ± 4.82E+01 −	9.04E+02 ± 7.33E−01 –
F20	7.00E+02 ± 1.33E+02 −	914.03 ± 2.36 +	9.46E+02 ± 2.57E+01 +	8.99E+02 ± 4.01E+01 −	9.04E+02 ± 9.75E−01 –
F21	9.26E+02 ± 2.44E+02 +	5.00E+02 ± 0.00E+00 =	7.31E+02 ± 3.33E+02 +	5.44E+02 ± 1.68E+02 +	5.00E+02 ± 1.74E−12 –
F22	8.87E+02 ± 2.70E+01 +	910.68 ± 15.75 +	9.99E+02 ± 3.84E+01 +	9.79+02 ± 4.17E+01 +	8.65E+02 ± 1.94E+01 –
F23	8.79E+02 ± 2.76E+02 +	534.16 ± 4.07E–04 +	8.40E+02 ± 2.91E+02 +	6.08E+02 ± 1.73E+02 +	5.34E+02 ± 3.63E−13 –
F24	5.48E+02 ± 3.58E+02 +	2.00E+02 ± 0.00E+00 =	3.46E+02 ± 3.78E+02 +	2.00E+02 ± 5.77E−01 =	2.00E+02 ± 2.88E−14 –
F25	5.75E+02 ± 3.75E+02 +	2.00E+02 ± 0.00E+00 =	3.40E+02 ± 3.63E+02 +	2.20+02 ± 5.09E+00 +	2.00E+02 ± 2.90E−14 –
+/ − /=	**15**/9/1	**10**/9/6	**25/0/0**	**21**/2/2	

Table 12
Wilcoxon's test results between the SCJADE and the state-of-the-art methods.

i	j	MPR	MNR	SPR	SNR	F(i) < F(j)	F(j) < F(i)	*p*-value	0.95 Confidence interval	
SCJADE	FMPSO [56]	14.266667	9.555556	214	86	15	9	0.0696053	−714.99996	0.79998
	HCLPSO [57]	10.5	9.444444	105	85	10	9	0.7022383	−260.98492	13.14999
	BES [58]	13	N/A	325	0	25	0	5.960e−08	−590.20	−54.27
	GPEAed [59]	12.619048	5.5	265	11	21	2	0.0001192	−2401.74912	−38.84495

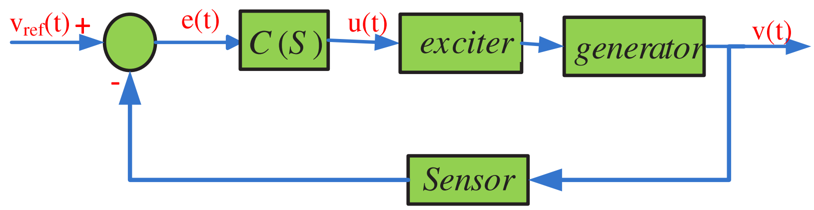


Fig. 5. Block diagram of AVR control system for a generating unit [104].

Table 13
Average ranking of SCJADE and state-of-the-art algorithms according to Friedman test.

Algorithm	RankT	Mean rank
SCJADE	1	1.98
HCLPSO	2	2.2
FMPSO	3	2.68
GPEAed	4	4.04
BES	5	4.1

Many papers have worked on the optimal tuning of k_P, k_I and, k_D coefficients of PID controller in an AVR system using metaheuristic methods, some of which include teaching-learning based optimization (TLBO) [71], firefly algorithm (FA) [72], RAO algorithm for optimizing a multi-term fractional-order PID (MFOPID) controller for improving the performance of the AVR System [73], bat algorithm (BA) [74], hybrid of PSO with gravitational search algorithm (PSOGSA) [75], an improved Lévy flight distribution algorithm with fitness-distance balance (FDB)-based guiding mechanism [76], chaotic optimization (CO) [77], model predictive controller aided with leader Harris hawks optimization (MPC-LHHO) algorithm [78], many optimizing liaisons (MOL) [79], local unimodal sampling (LUS) optimization algorithm [80], cooperation search algorithm (CSA) [81], Taguchi combined genetic algorithm method (TGA) [82], a new hybrid SA and gorilla troops optimization (GTO) [83], a hybrid genetic algorithm (GA) [84], a new improved kidney-inspired algorithm (IKA) [85], cuckoo search (CS) [86], a novel hybrid optimizer via Harris hawks optimization (HHO) and SA technique (HHO-SA) for proportional + integral + derivative plus second order derivative (PID+DD) controller adopted in the AVR [87], whale optimization algorithm (WOA) [88, 89], symbiotic organism search (SOS) [90], salp swarm algorithm (SSA) [91], Jaya algorithm and its improved version [92,93], multi-objective extremal optimization [94], arithmetic optimization algorithm (AOA) [95], enhanced crow search algorithm [96], a novel modified smoothed function algorithm (MSFA) [97], equilibrium optimizer [98], artificial ecosystem-based optimization [99], water cycle algorithm [100], a new improved artificial bee colony (IABC) high-order approximation (HOA)-based fractional order PID (IABC/HOA-FOPID) controller [101], sine–cosine algorithm [102], and a hybrid simulated annealing — Manta ray foraging optimization algorithm [103].

Fig. 5 shows the block diagram of the AVR control system. Note that C(s) represents the transfer function of the PID controller. The goal is to design a controller so that the output of the power system has some defined characteristics.

The transfer function (TF), $C(s)$, of the PID controller is defined as follows [104]:

$$C(s) = K_P + \frac{K_I}{S} + K_D S \tag{16}$$

where K_P, K_I, and K_D represent the proportional, integral, and derivative coefficients, respectively. The control system output can be written using the following equation [104]:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \tag{17}$$

In order to design a PID controller for a given system, we have to optimally adjust the coefficients K_P, K_I, and K_D to obtain the desired performance.

For the problem under consideration in this study, i.e. PID design for synchronous generator's AVR system, we mount a first-order low-pass filter in the derivative path of the PID controller for obtaining a smoother voltage profile [71]. The TF of the low-pass filter may be defined as [71]:

$$G_{Filter}(s) = \frac{s}{s\left(\frac{1}{N}\right)T + 1} \tag{18}$$

where T and N represent the time constant and the filter coefficient, respectively, whose values are defined in the interval of (0.01, 0.1) s and (0.1, 2.0), respectively. Thus, considering this filter, the TF of the PID controller used in this study may be rewritten as [71]:

$$G_{PID}(s) = K_P + \frac{K_i}{S} + K_d S\left(\frac{1}{s\left(\frac{1}{N}\right)T + 1}\right) \tag{19}$$

In different recently published articles, various objectives have been considered for the problem of optimal design of PID controllers. In this study, we have used four important criteria related to the time response of the studied AVR system as indicators of the optimal performance of the system, which include the following:

1. Rise time T_R

T_R is the time during which the time response of the system changes from 10% to 90% of its final value [104].

2. Settling time T_S

Settling time is the time at which the system time response remains within the range of ±2% of its final response and does not exceed this range under any circumstances [104].

3. Maximum Overshoot M_P (p.u.)

Another important characteristic of the step response of a system is its maximum value of overshoot. In this paper, the maximum value of the first overshoot of step response subtracted by 1 is used. this way, if the maximum value of the first overshoot of the step response is less than 1, the value of M_P, as the fitness value, is considered to be zero [104]

4. Steady-state error E_{ss} (p.u.)

E_{ss} is defined as the difference between the final value of the system response to the unit step input and one, i.e. $E_{ss} = 1 - V_{t(end)}$ [104].

Table 14 presents an overview of some objective functions used in the literature for the optimal design of PID controllers.

Table 14
Overview of the objective functions for the optimal PID design problems used in the literature.

Reference	Cost
[79,88,105]	$ISE = \int e^2(t)dt$
[79,106]	$IAE = \int e(t)dt$
[79,80,107]	$ITSE = \int te^2(t)dt$
[79]	$ITAE = \int t\,\lvert e(t)\rvert\,dt$
[80]	$Cost = w_1 * (ISE \text{ or } ITSE \text{ or } IAE \text{ or } ITAE) + w_2 * T_S + w_3 * M_P$
[90]	$Cost = w_1 * \int t\,\lvert e(t)\rvert\,dt + w_2 * N_{cp} + w_3 * S_r$
[108,109]	$Cost = w_1 * M_P + w_2 * T_R + w_3 * T_S + w_4 * E_{SS} + \int \left(w_5 \lvert e(t)\rvert + w_6 \left(V_v(t)\right)^2 \right) dt + \frac{w_7}{P_m} + \frac{w_8}{G_m}$
[108]	$Cost = \left(1 - w_1\right) P_m + \omega_{gc}$
[110,111]	$Cost = \left(w_1 * M_P\right)^2 + w_2 * \left(T_S\right)^2 + w_3 * (\max\_dv)^{-2}$
[86]	$Cost = w_1 * M_P + w_2 * \left(\int t\,\lvert e(t)\rvert\,dt\right) + w_3 * T_S + w_4 * E_{SS}$
[82]	$Cost = w_1 * M_P + w_2 * T_R + w_3 * T_S + w_4 * E_{SS}$
[77,112]	$Cost = \left(\int t\,\lvert e(t)\rvert\,dt\right) + \left(w_1 * M_P\right)$
[85]	$Cost = \mu \left(\int te^2(t)dt\right) + \left(1 - e^{-\beta}\right)\left(M_P + E_{SS}\right) + e^{-\beta}\left(T_S - T_R\right)$
[71,104,108,113–115]	$Cost = \left(1 - e^{-\beta}\right)\left(M_P + E_{SS}\right) + e^{-\beta}\left(T_S - T_R\right)$
[116]	$Cost = w_1 * M_P + w_2 * T_R + w_3 * T_S$
[111]	$Cost = w_1 * M_P + w_2 * T_R + w_3 * T_S + w_4 \int e(t)dt + w_5 \int u^2(t)\,dt$
[117]	$Cost = \left[\int te^2(t)dt \quad \int \Delta u^2(t)\,dt \quad \int te^2_{load}(t)\,dt\right]$
[118]	$Cost = \left[\omega_{gc} \quad P_m\right]$
[84]	$Cost = \frac{e^{-\beta}T_S}{(1-e^{-\beta})*(1-T_R)} + e^{-\beta}M_P + E_{SS}$
[103]	$Cost = \left(1 - e^{-\beta}\right)\left(\frac{M_P}{\alpha} + E_{SS}\right) + e^{-\beta}\left(T_S - T_R\right)$

5.2. Model of AVR system's components

In this section, the model of the components of the AVR systems is briefly introduced [104]:

The amplifier, as one of the components of AVR, is modeled using a gain factor K_A and a time constant τ_A, whose TF is as follows [104]:

$$\frac{V_R(s)}{V_C(s)} = \frac{K_A}{1 + \tau_A S} \tag{20}$$

The value of the gain K_A is set to be in the interval of 10 to 400, and the time constant of the amplifier is set to small values between 0.02 s and 0.1 s so that it is ignored in most cases. In our study, K_A and τ_A have been set to 10 and 0.1 s, respectively.

To model the exciter system in its simplest form, by ignoring the magnetic saturation phenomenon and other nonlinear factors, its TF can be defined using a time constant τ_E with a small value, and gain K_E as follows [104]:

$$\frac{V_E(s)}{V_R(s)} = \frac{K_E}{1 + \tau_E S} \tag{21}$$

where the value of K_E is in the interval of 1 to 200, and the value of the exciter time constant τ_E is defined between 0.5 s and 1 s. In this study, K_E and τ_E have been set to 1 and 0.4 s, respectively.

The generator, as another component of the AVR system, can be modeled using gain K_G and time constant τ_G as the following equation [104]:

$$\frac{V_F(s)}{V_E(s)} = \frac{K_G}{1 + \tau_G S} \tag{22}$$

These parameters depend on the (electrical) load so that from full-load to no-load conditions, K_G can change between 0.7 and 1, and τ_G has a variation between 1 s and 2 s. In our study, K_G and τ_G have been set to 1 and 1 s, respectively.

Eventually, the sensor can be modeled using a simple first-order TF defined as follows [104]:

$$\frac{V_S(s)}{V_F(s)} = \frac{K_R}{1 + \tau_R S} \tag{23}$$

where we normally set the gain K_R to the constant value of 1, and the time constant τ_R to a very small value in the range of 0.01 s to 0.06 s. In our study, K_R and τ_R have been set to 1 and 0.01 s, respectively.

Using the models described above, the block diagram of an AVR system with a PID controller can be constructed as shown in Fig. 6.

5.3. Numerical results obtained by optimization methods and performance comparison

In this study, the transfer function (TF) of the AVR system using the above-mentioned values of the parameters and without considering the TF of the PID controller is as follows [71]:

$$\frac{V_F(s)}{V_{ref}(s)} = \frac{0.1S + 10}{0.0004S^4 + 0.0454S^3 + 0.555S^2 + 1.51S + 11} \tag{24}$$

The system response to a unit step input is plotted in Fig. 7. As can be seen, this AVR system shows an oscillating step response without a controller, which is not desired in practical systems. The step response of the studied AVR system has a peak value of 1.5 p.u., a rise time of 0.261 s, a settling time of 6.9834 s, and a maximum steady-state error of $E_{SS} = 0.0918$.

Therefore, this AVR system substantially needs a PID controller with appropriately tuned coefficients, whose optimal values are computed using the proposed algorithm in this study.

5.3.1. PID design employing modified DEs

In this section, the proposed versions of DE algorithms are used to optimally design a PID controller for the AVR system. The diagram of the AVR system employing DE algorithms for the optimal design of the PID controller is demonstrated in Fig. 8.

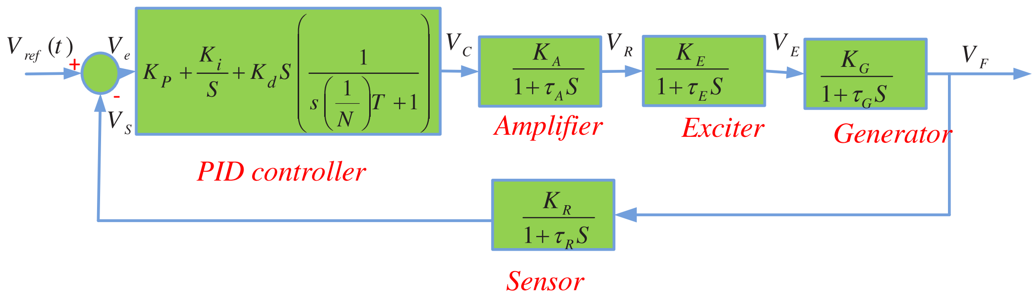


Fig. 6. Block diagram of an AVR system with a PID controller.

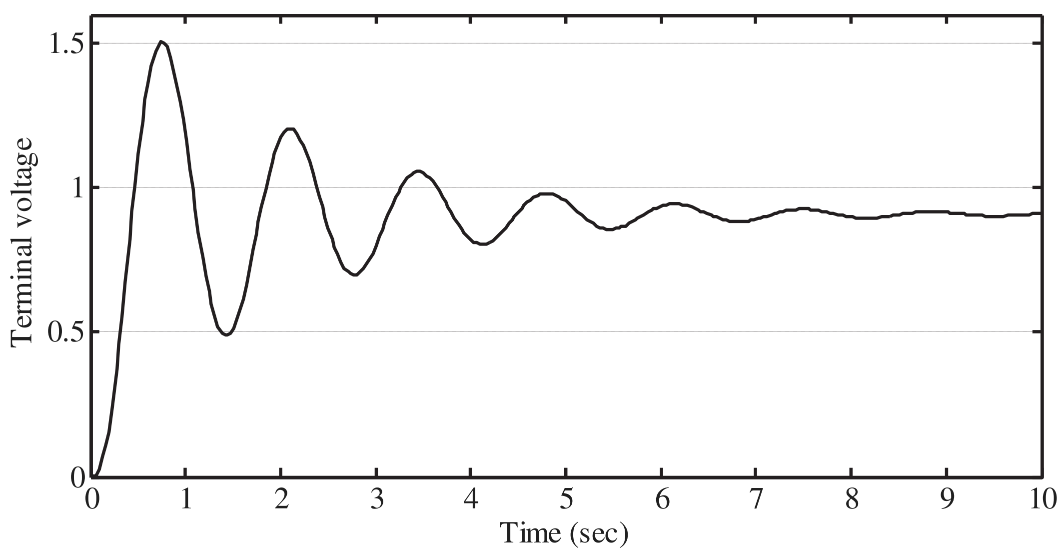


Fig. 7. Terminal voltage step response of an AVR system without a PID controller.

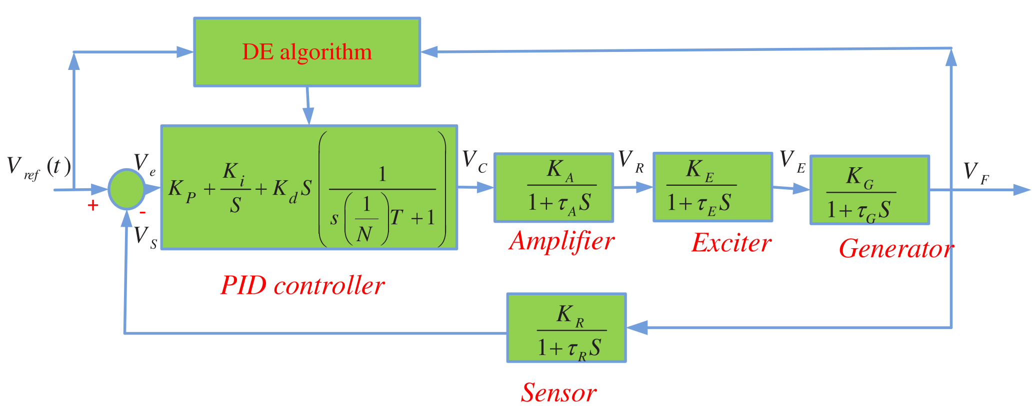


Fig. 8. Block diagram of the AVR system employing DEs for the optimal design of PID controller.

To optimally design this controller using DE algorithms, we consider the objective function as follows [104]:

$$Cost = \left(1 - e^{-\beta}\right)\left(M_P + E_{SS}\right) + e^{-\beta}\left(T_S - T_R\right) \tag{25}$$

Minimizing this objective function makes the system output stable in terms of both transient response and steady-state response. The lower values of maximum overshoot, rise time, and settling time make the response of the considered system fast enough with acceptable oscillations. The coefficient β is set to values around 1 in most articles; however, in this paper, β is set to values between 0.1 and 2.

In this study, we have performed 20 independent runs for each algorithm with an iteration number of 100, and a population number of 30. A summary of the best results obtained by the best algorithm concluded in the previous section, i.e., SCJADE, along with its original version, i.e., JADE, for various values of β ranging from 0.1 to 2 is given in Table 15. Moreover, the convergence characteristics of both algorithms for the best-obtained solution are shown in Fig. 9 for different values of β. According to the results presented in Table 15, it is clear that increasing the β coefficient reduces the amount of steady-state error and maximum overshoot, and on the other hand, increases the system rise time and settling time; thus, it can be concluded that an optimal value for βmay be considered as 1. Furthermore, the AVR system with the PID controller optimized by these proposed algorithms has a much better time response than the original AVR system (Eq. (24)), and all four control parameters have been significantly reduced, meaning that the AVR system with the optimally designed PID has a shorter settling

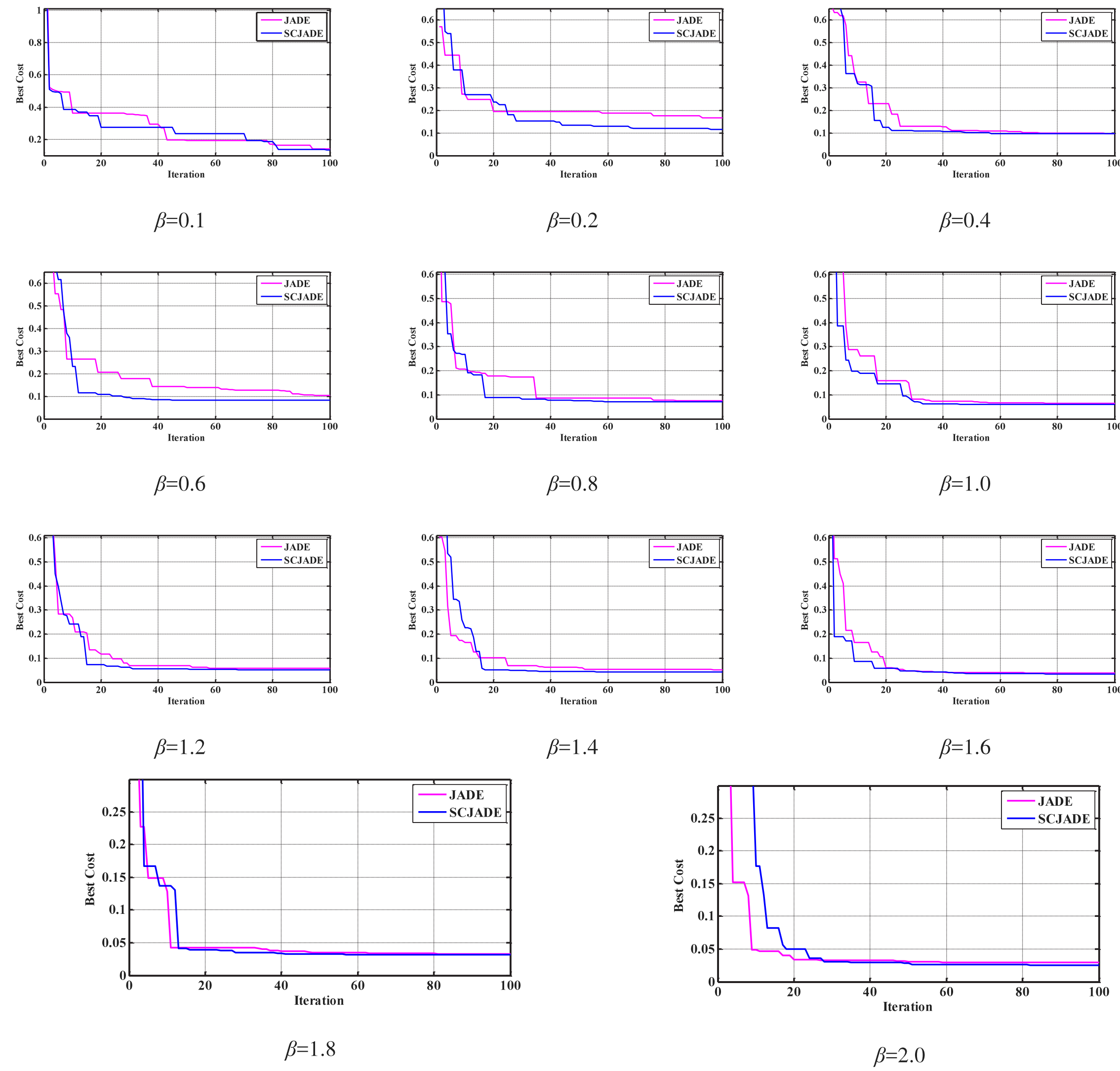


Fig. 9. The convergence characteristics of the algorithms for the optimal design of PID controller for AVR system.

Table 15
A comparison between the proposed JADE and SCJADE algorithms for the optimal design of PID controller for the AVR system.

β	Algorithm	K_P	K_I	K_D	N	T	T_R	T_S	M_P	E_{SS}	Cost
0.1	JADE	0.50641	0.001	0.21337	200	0.26028	0.2852	0.4314	0.0545	0.0401	0.14129
	SCJADE	0.49655	0.54171	0.20799	128.9041	1.0459	0.2783	0.4244	0.0248	0.0082	**0.13533**
0.2	JADE	0.30212	0.001	0.16167	158.7075	0.42877	0.3117	0.4749	0.0	0.1758	0.16545
	SCJADE	0.48231	0.001	0.21483	155.5519	0.23492	0.2722	0.4087	0.0064	0.0130	**0.11525**
0.4	JADE	0.4779	0.0018692	0.21493	200	0.30674	0.2714	0.4075	0.0014	0.0175	0.097494
	SCJADE	0.47888	0.001	0.21874	122.9127	0.18781	0.2679	0.4021	0.0022	0.0168	**0.096219**
0.6	JADE	0.4922	0.51544	0.18136	200	1.9518	0.3040	0.4660	0.0258	0.0074	0.10385
	SCJADE	0.4776	0.001	0.22083	189.9675	0.29077	0.2660	0.3992	0.0015	0.0174	**0.081617**
0.8	JADE	0.47531	0.001	0.20925	200	0.28824	0.2845	0.4304	0.0163	0.0025	0.075934
	SCJADE	0.47556	0.010883	0.21729	155.5248	0.23975	0.2709	0.4081	5.5e−04	0.0154	**0.070476**
1	JADE	0.50245	0.22881	0.20226	45.2717	0.09611	0.2891	0.4388	0.0109	0.0018	0.063026
	SCJADE	0.46092	0.007824	0.20957	95.8451	0.14064	0.2867	0.4375	0.0073	0.0011	**0.060779**
1.2	JADE	0.48373	0.33097	0.17486	63.5694	0.21274	0.3283	0.5083	0.0053	5.7e−04	0.058342
	SCJADE	0.47305	0.074305	0.20254	148.3506	0.23859	0.2936	0.4511	0.0029	0.0026	**0.051286**
1.4	JADE	0.46357	0.37206	0.151	144.7177	1.62	0.3523	0.5496	0.0040	8.30e−04	0.052276
	SCJADE	0.46848	0.10511	0.19645	92.2368	0.15438	0.3035	0.4713	8.3e−04	6.35e−04	**0.042484**

(*continued on next page*)

Table 15 (*continued*).

β	Algorithm	K_P	K_I	K_D	N	T	T_R	T_S	M_P	E_{SS}	Cost
1.6	JADE	0.47924	0.23657	0.18616	108.3171	0.24465	0.3150	0.4888	0.0011	0.0014	0.037144
	SCJADE	0.44369	0.028777	0.19861	198.6772	0.2987	0.3041	0.4720	4.95e−4	5.31e−04	**0.034722**
1.8	JADE	0.47606	0.34514	0.16521	150.6778	0.76759	0.3418	0.5360	0.0	0.0010	0.032972
	SCJADE	0.47791	0.32631	0.17342	115.243	0.41843	0.3318	0.5192	0.0	7.72e−04	**0.031624**
2	JADE	0.4443	0.36236	0.14596	135.3583	1.946	0.3632	0.5775	0.0	1.25e−04	0.029109
	SCJADE	0.47541	0.25743	0.1798	124.5152	0.3018	0.3253	0.5094	0.0	6.1e−04	**0.02544**

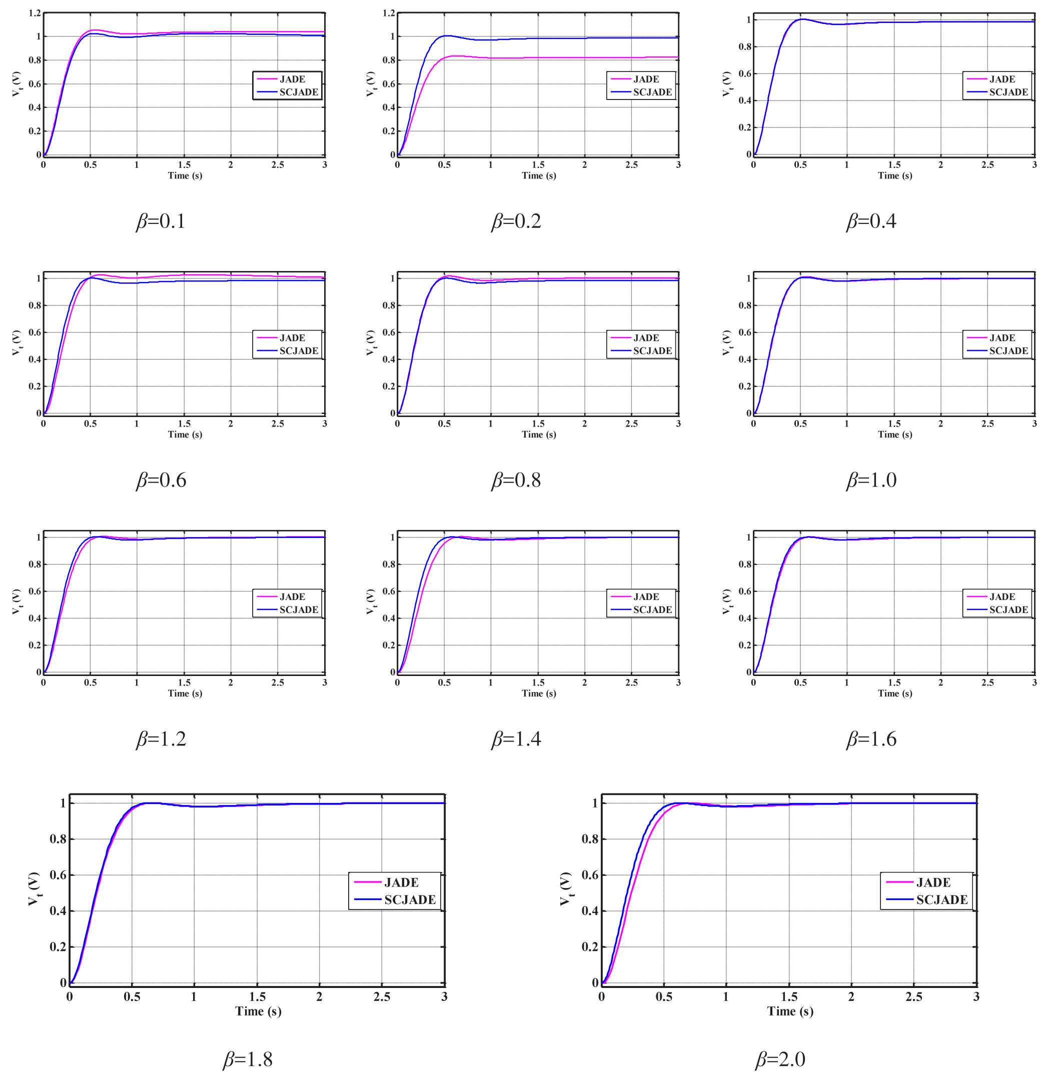


Fig. 10. Terminal voltage step response of an AVR system with optimized PID controller using JADE and SCJADE algorithms.

and rise time with smaller values of maximum overshoot and a steady-state error compared to the AVR system without a controller. Overall, by comparing the results obtained by SCJADE and JADE algorithms, we find that the proposed modifications to the standard DE algorithm, despite their simplicity, have been very effective.

The step response of the terminal voltage of the AVR system for the optimal coefficients, presented in Table 15, is shown in Fig. 10 for changing values of β from 0.1 to 2. By comparing these responses with those obtained for the AVR system without a controller, depicted in Fig. 7, the effect of the optimal design of the PID controller on improving the performance of the AVR system is clearly observed. As seen, the oscillations and steady-state error have been drastically reduced; and even for some cases, the overshoot value has been reduced to zero, which may be the important features of the optimal response we have been looking for.

6. Conclusions

This paper proposes a self-competitive control strategy for improving different variants of DE algorithms without imposing any additional computational burden on the DE algorithm. The proposed strategy includes employing a competitive control parameter to the mutation operator of the original variants of the DE algorithm. In order to investigate the effectiveness of the proposed strategy, the improved and original variants of the DE algorithms were used for solving 25 real parameter test functions and also optimal tuning of PID controller for an AVR system. Identical conditions were used for each algorithm and its improved version. Two non-parametric statistical tests, i.e., the Friedman rank test and the Wilcoxon signed-rank test, were used for assessing the rank and significance of the proposed improved variants of the DE algorithm. The extensive simulations and comparisons between the optimal results of 25 real parameter test functions and the optimal design of the AVR system prove the efficiency of the proposed self-competitive strategy in improving different variants of DE.

It is possible to utilize improved DE algorithms in future studies for solving different real-world problems. In recent years, a large number of evolutionary algorithms have been presented. The performance of the proposed self-competitive strategy for these algorithms can be investigated. Also, the performance of this strategy can be investigated on the hybridizations of DE and particle swarm optimization algorithms. Furthermore, the performance of the proposed strategy can be examined for multi-objective DE algorithms.

Declaration of competing interest

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

Data availability

Data will be made available on request.

References

[1] E. Çelik, A powerful variant of symbiotic organisms search algorithm for global optimization, Eng. Appl. Artif. Intell. 87 (2020) 103294.

[2] E.H. Houssein, D. Oliva, E. Çelik, M.M. Emam, R.M. Ghoniem, Boosted sooty tern optimization algorithm for global optimization and feature selection, Expert Syst. Appl. 213 (2023) 119015.

[3] E. Çelik, N. Öztürk, Y. Arya, Advancement of the search process of salp swarm algorithm for global optimization problems, Expert Syst. Appl. 182 (2021) 115292.

[4] E. Çelik, Incorporation of stochastic fractal search algorithm into efficient design of PID controller for an automatic voltage regulator system, Neural Comput. Appl. 30 (2018) 1991–2002.

[5] E. Çelik, N. Öztürk, A hybrid symbiotic organisms search and simulated annealing technique applied to efficient design of PID controller for automatic voltage regulator, Soft Comput. 22 (2018) 8011–8024.

[6] M. Ghasemi, M.-A. Akbari, C. Jun, S.M. Bateni, M. Zare, A. Zahedi, H.-T. Pai, S.S. Band, M. Moslehpour, K.-W. Chau, Circulatory system based optimization (CSBO): an expert multilevel biologically inspired meta-heuristic algorithm, Eng. Appl. Comput. Fluid Mech. 16 (2022) 1483–1525, http://dx.doi.org/10.1080/19942060.2022.2098826.

[7] R. Storn, K. Price, et al., Differential Evolution–a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces:, Technical Report TR-95-012, Int. Comput. Sci. Berkeley, Calif., 1995.

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

[9] K. Price, R.M. Storn, J.A. Lampinen, Differential Evolution: A Practical Approach To Global Optimization, Springer Science & Business Media, 2006.

[10] F. Neri, V. Tirronen, Recent advances in differential evolution: a survey and experimental analysis, Artif. Intell. Rev. 33 (2009) 61–106, http://dx.doi.org/10.1007/s10462-009-9137-2.

[11] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P.N. Suganthan, Q. Zhang, Multiobjective evolutionary algorithms: A survey of the state of the art, Swarm Evol. Comput. 1 (2011) 32–49, http://dx.doi.org/10.1016/j.swevo.2011.03.001.

[12] H. Wang, S. Rahnamayan, H. Sun, M.G.H. Omran, Gaussian bare-bones differential evolution, IEEE Trans. Cybern. 43 (2013) 634–647, http://dx.doi.org/10.1109/tsmcb.2012.2213808.

[13] B. Dorronsoro, P. Bouvry, Improving classical and decentralized differential evolution with new mutation operator and population topologies, IEEE Trans. Evol. Comput. 15 (2011) 67–98, http://dx.doi.org/10.1109/tevc.2010.2081369.

[14] A.W. Iorio, X. Li, Improving the performance and scalability of differential evolution on problems exhibiting parameter interactions, Soft Comput. 15 (2010) 1769–1792, http://dx.doi.org/10.1007/s00500-010-0614-y.

[15] Y. Cai, J. Wang, Differential evolution with neighborhood and direction information for numerical optimization, IEEE Trans. Cybern. 43 (2013) 2202–2215, http://dx.doi.org/10.1109/tcyb.2013.2245501.

[16] J. Brest, A. Zamuda, B. Bošković, S. Greiner, V. Žumer, An analysis of the control parameters' adaptation in DE, in: Adv. Differ. Evol., Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 89–110, http://dx.doi.org/10.1007/978-3-540-68830-3_3.

[17] S. Das, A. Mandal, R. Mukherjee, An adaptive differential evolution algorithm for global optimization in dynamic environments, IEEE Trans. Cybern. 44 (2014) 966–978, http://dx.doi.org/10.1109/tcyb.2013.2278188.

[18] W.-J. Yu, M. Shen, W.-N. Chen, Z.-H. Zhan, Y.-J. Gong, Y. Lin, O. Liu, J. Zhang, Differential evolution with two-level parameter adaptation, IEEE Trans. Cybern. 44 (2014) 1080–1099, http://dx.doi.org/10.1109/tcyb.2013.2279211.

[19] J. Zhang, A.C. Sanderson, JADE: Adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (2009) 945–958, http://dx.doi.org/10.1109/tevc.2009.2014613.

[20] S.-M. Guo, C.-C. Yang, P.-H. Hsu, J.S.-H. Tsai, Improving differential evolution with a successful-parent-selecting framework, IEEE Trans. Evol. Comput. 19 (2015) 717–730, http://dx.doi.org/10.1109/tevc.2014.2375933.

[21] J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems, IEEE Trans. Evol. Comput. 10 (2006) 646–657, http://dx.doi.org/10.1109/tevc.2006.872133.

[22] Y. Wang, Z. Cai, Q. Zhang, Differential evolution with composite trial vector generation strategies and control parameters, IEEE Trans. Evol. Comput. 15 (2011) 55–66, http://dx.doi.org/10.1109/tevc.2010.2087271.

[23] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2009) 398–417, http://dx.doi.org/10.1109/tevc.2008.927706.

[24] W. Zhu, Y. Tang, J. Fang, W. Zhang, Adaptive population tuning scheme for differential evolution, Inf. Sci. (Ny). 223 (2013) 164–191, http://dx.doi.org/10.1016/j.ins.2012.09.019.

[25] J. Liang, W. Xu, C. Yue, K. Yu, H. Song, O.D. Crisalle, B. Qu, Multimodal multiobjective optimization with differential evolution, Swarm Evol. Comput. 44 (2019) 1028–1059, http://dx.doi.org/10.1016/j.swevo.2018.10.016.

[26] S.-M. Guo, C.-C. Yang, Enhancing differential evolution utilizing Eigenvector-Based crossover operator, IEEE Trans. Evol. Comput. 19 (2015) 31–49, http://dx.doi.org/10.1109/tevc.2013.2297160.

[27] R. Mallipeddi, P.N. Suganthan, Differential evolution algorithm with ensemble of parameters and mutation and crossover strategies, Swarm Evol. Memet. Comput. (2010) 71–78, http://dx.doi.org/10.1007/978-3-642-17563-3_9.

[28] R. Mallipeddi, P.N. Suganthan, Q.K. Pan, M.F. Tasgetiren, Differential evolution algorithm with ensemble of parameters and mutation strategies, Appl. Soft Comput. 11 (2011) 1679–1696, http://dx.doi.org/10.1016/j.asoc.2010.04.024.

[29] Z. Zhang, A new multi-population-based differential evolution, Int. J. Comput. Sci. Math. 6 (2015) 88–96, http://dx.doi.org/10.1504/IJCSM.2015.067546.

[30] S. Biswas, S. Kundu, S. Das, An improved parent-centric mutation with normalized neighborhoods for inducing niching behavior in differential evolution, IEEE Trans. Cybern. 44 (2014) 1726–1737, http://dx.doi.org/10.1109/tcyb.2013.2292971.

[31] M.G. Epitropakis, V.P. Plagianakos, M.N. Vrahatis, Balancing the exploration and exploitation capabilities of the differential evolution algorithm, in: 2008 IEEE Congr. Evol. Comput. (IEEE World Congr. Comput. Intell., 2008, http://dx.doi.org/10.1109/cec.2008.4631159.

[32] M. Ghasemi, M. Taghizadeh, S. Ghavidel, A. Abbasian, Colonial competitive differential evolution: An experimental study for optimal economic load dispatch, Appl. Soft Comput. 40 (2016) 342–363, http://dx.doi.org/10.1016/j.asoc.2015.11.033.

[33] M.G. Epitropakis, D.K. Tasoulis, N.G. Pavlidis, V.P. Plagianakos, M.N. Vrahatis, Enhancing differential evolution utilizing proximity-based mutation operators, IEEE Trans. Evol. Comput. 15 (2011) 99–119, http://dx.doi.org/10.1109/tevc.2010.2083670.

[34] C. Lin, A. Qing, Q. Feng, A new differential mutation base generator for differential evolution, J. Global Optim. 49 (2011) 69–90, http://dx.doi.org/10.1007/s10898-010-9535-7.

[35] S. Biswas, S. Kundu, S. Das, A.V. Vasilakos, Teaching and learning best differential evoltuion with self adaptation for real parameter optimization, in: 2013 IEEE Congr. Evol. Comput., 2013, http://dx.doi.org/10.1109/cec.2013.6557691.

[36] H.-Y.Y. Fan, J. Lampinen, A trigonometric mutation operation to differential evolution, J. Global Optim. 27 (2003) 105–129, http://dx.doi.org/10.1023/A:1024653025686.

[37] H. Wang, S. Rahnamayan, Z. Wu, Parallel differential evolution with self-adapting control parameters and generalized opposition-based learning for solving high-dimensional optimization problems, J. Parallel Distrib. Comput. 73 (2013) 62–73, http://dx.doi.org/10.1016/j.jpdc.2012.02.019.

[38] M. Ghasemi, J. Aghaei, E. Akbari, S. Ghavidel, L. Li, A differential evolution particle swarm optimizer for various types of multi-area economic dispatch problems, Energy 107 (2016) 182–195, http://dx.doi.org/10.1016/j.energy.2016.04.002.

[39] B. Chen, Y. Lin, W. Zeng, D. Zhang, Y.-W. Si, Modified differential evolution algorithm using a new diversity maintenance strategy for multi-objective optimization problems, Appl. Intell. 43 (2015) 49–73, http://dx.doi.org/10.1007/s10489-014-0619-9.

[40] S.-Z. Zhao, P.N. Suganthan, Comprehensive comparison of convergence performance of optimization algorithms based on nonparametric statistical tests, in: 2012 IEEE Congr. Evol. Comput., 2012, http://dx.doi.org/10.1109/cec.2012.6252910.

[41] J. Brest, B. Boskovic, A. Zamuda, I. Fister, E. Mezura-Montes, Real parameter single objective optimization using self-adaptive differential evolution algorithm with more strategies, in: 2013 IEEE Congr. Evol. Comput. CEC 2013, 2013, pp. 377–383, http://dx.doi.org/10.1109/CEC.2013.6557594.

[42] A. Zamuda, J. Brest, E. Mezura-Montes, Structured population size reduction differential evolution with multiple mutation strategies on CEC 2013 real parameter optimization, in: 2013 IEEE Congr. Evol. Comput., 2013, http://dx.doi.org/10.1109/cec.2013.6557794.

[43] S. Rahnamayan, H.R. Tizhoosh, M.M.A. Salama, Opposition-based differential evolution, IEEE Trans. Evol. Comput. 12 (2008) 64–79, http://dx.doi.org/10.1109/tevc.2007.894200.

[44] P. Kaelo, M.M. Ali, Differential evolution algorithms using hybrid mutation, Comput. Optim. Appl. 37 (2007) 231–246, http://dx.doi.org/10.1007/s10589-007-9014-3.

[45] Y. Chen, L. Li, H. Peng, J. Xiao, Q. Wu, Dynamic multi-swarm differential learning particle swarm optimizer, Swarm Evol. Comput. 39 (2018) 209–221, http://dx.doi.org/10.1016/j.swevo.2017.10.004.

[46] Q. Huang, K. Zhang, J. Song, Y. Zhang, J. Shi, Adaptive differential evolution with a Lagrange interpolation argument algorithm, Inf. Sci. (Ny). 472 (2019) 180–202, http://dx.doi.org/10.1016/j.ins.2018.09.004.

[47] S. Wang, Y. Li, H. Yang, Self-adaptive mutation differential evolution algorithm based on particle swarm optimization, Appl. Soft Comput. 81 (2019) 105496, http://dx.doi.org/10.1016/j.asoc.2019.105496.

[48] G. Wu, R. Mallipeddi, P.N. Suganthan, R. Wang, H. Chen, Differential evolution with multi-population based ensemble of mutation strategies, Inf. Sci. (Ny). 329 (2016) 329–345, http://dx.doi.org/10.1016/j.ins.2015.09.009.

[49] S. Kukkonen, C.A.C. Coello, Applying exponential weighting moving average control parameter adaptation technique with generalized differential evolution, in: 2016 IEEE Congr. Evol. Comput., 2016, http://dx.doi.org/10.1109/cec.2016.7744398.

[50] X. Zhou, Z. Wu, H. Wang, S. Rahnamayan, Enhancing differential evolution with role assignment scheme, Soft Comput. 18 (2014) 2209–2225, http://dx.doi.org/10.1007/s00500-013-1195-3.

[51] W. Gong, Z. Cai, C.X. Ling, H. Li, Enhanced differential evolution with adaptive strategies for numerical optimization, IEEE Trans. Syst. Man Cybern. Part B 41 (2011) 397–413, http://dx.doi.org/10.1109/tsmcb.2010.2056367.

[52] M. Vasile, E. Minisci, M. Locatelli, An inflationary differential evolution algorithm for space trajectory optimization, IEEE Trans. Evol. Comput. 15 (2011) 267–281, http://dx.doi.org/10.1109/tevc.2010.2087026.

[53] R.-J. He, Z.-Y. Yang, Differential evolution with adaptive mutation and parameter control using Lévy probability distribution, J. Comput. Sci. Tech. 27 (2012) 1035–1055, http://dx.doi.org/10.1007/s11390-012-1283-3.

[54] A.M. Sutton, M. Lunacek, L.D. Whitley, Differential evolution and non-separability: Using selective pressure to focus search, in: Proc. GECCO 2007 Genet. Evol. Comput. Conf., ACM Press, New York, New York, USA, 2007, pp. 1428–1435, http://dx.doi.org/10.1145/1276958.1277221.

[55] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (2011) 3–18, http://dx.doi.org/10.1016/j.swevo.2011.02.002.

[56] H. Buch, I.N. Trivedi, P. Jangir, Moth flame optimization to solve optimal power flow with non-parametric statistical evaluation validation, Cogent Eng. 4 (2017) 1286731, http://dx.doi.org/10.1080/23311916.2017.1286731.

[57] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y.-P. Chen, A. Auger, S. Tiwari, Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization, Nanyang Technological University Singapore, 2005.

[58] X. Xia, Y. Xing, B. Wei, Y. Zhang, X. Li, X. Deng, L. Gui, A fitness-based multi-role particle swarm optimization, Swarm Evol. Comput. 44 (2019) 349–364, http://dx.doi.org/10.1016/J.SWEVO.2018.04.006.

[59] N. Lynn, P.N. Suganthan, Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation, Swarm Evol. Comput. 24 (2015) 11–24, http://dx.doi.org/10.1016/J.SWEVO.2015.05.002.

[60] H.A. Alsattar, A.A. Zaidan, B.B. Zaidan, Novel meta-heuristic bald eagle search optimisation algorithm, Artif. Intell. Rev. 53 (2020) 2237–2264, http://dx.doi.org/10.1007/S10462-019-09732-5/METRICS.

[61] Z. Hu, C. Gao, Q. Su, A novel evolutionary algorithm based on even difference grey model, Expert Syst. Appl. 176 (2021) 114898, http://dx.doi.org/10.1016/J.ESWA.2021.114898.

[62] E. Çelik, N. Öztürk, Novel fuzzy 1PD-TI controller for AGC of interconnected electric power systems with renewable power generation and energy storage devices, Eng. Sci. Technol. Int. J. (2022) 101166.

[63] E. Çelik, Performance analysis of SSA optimized fuzzy 1PD-PI controller on AGC of renewable energy assisted thermal and hydro-thermal power systems, J. Ambient Intell. Humaniz. Comput. (2022) 1–20.

[64] E. Çelik, N.I. Öztürk, G. Bal, Design of PID controlled-automatic voltage regulator system based on a new neighborhood strategy of simulated annealing, in: 4th Int. Conf. Eng. Nat. Sci., Kiev, Ukraine, 2018.

[65] E. Çelik, N. Öztürk, E.H. Houssein, Influence of energy storage device on load frequency control of an interconnected dual-area thermal and solar photovoltaic power system, Neural Comput. Appl. 34 (2022) 20083–20099.

[66] K.H. Ang, G. Chong, Y. Li, PID control system analysis, design, and technology, IEEE Trans. Control Syst. Technol. 13 (2005) 559–576, http://dx.doi.org/10.1109/tcst.2005.847331.

[67] C.C. Hang, K.J. Åström, W.K. Ho, Refinements of the Ziegler–Nichols tuning formula, IEE Proc. D Control Theory Appl. 138 (1991) 111, http://dx.doi.org/10.1049/ip-d.1991.0015.

[68] S.P. Boyd, C.H. Barratt, Linear Controller Design: Limits of Performance, Citeseer, 1991.

[69] K.J. Åström, T. Hägglund, PID Controllers: Theory, Design, and Tuning, Instrument society of America Research, Triangle Park, NC, 1995.

[70] S.-Y. Chu, C.-C. Teng, Tuning of PID controllers based on gain and phase margin specifications using fuzzy neural network, Fuzzy Sets and Systems 101 (1999) 21–30, http://dx.doi.org/10.1016/s0165-0114(97)00039-0.

[71] S. Chatterjee, V. Mukherjee, PID controller for automatic voltage regulator using teaching–learning based optimization technique, Int. J. Electr. Power Energy Syst. 77 (2016) 418–429, http://dx.doi.org/10.1016/j.ijepes.2015.11.010.

[72] O. Bendjeghaba, Continuous firefly algorithm for optimal tuning of PID controller in AVR system, J. Electr. Eng. 65 (2014) 44–49.

[73] N. Paliwal, L. Srivastava, M. Pandit, Rao algorithm based optimal multi-term FOPID controller for automatic voltage regulator system, Optim. Control Appl. Methods 43 (2022) 1707–1734.

[74] B. Omar, I.B. Saida, Bat algorithm for optimal tuning of pid controller in an avr system, in: Int. Conf. Control. Eng. Inf. Technol., 2014, pp. 158–170.

[75] S. Kansit, W. Assawinchaichote, Optimization of PID controller based on PSOGSA for an automatic voltage regulator system, Procedia Comput. Sci. 86 (2016) 87–90, http://dx.doi.org/10.1016/j.procs.2016.05.022.

[76] H. Bakir, U. Guvenc, H.T. Kahraman, S. Duman, Improved Lévy flight distribution algorithm with FDB-based guiding mechanism for AVR system optimal design, Comput. Ind. Eng. 168 (2022) 108032.

[77] L. dos Santos Coelho, Tuning of PID controller for an automatic regulator voltage system using chaotic optimization approach, Chaos Solitons Fractals 39 (2009) 1504–1514, http://dx.doi.org/10.1016/j.chaos.2007.06.018.

[78] V. Kumar, V. Sharma, R. Naresh, Leader Harris Hawks algorithm based optimal controller for automatic generation control in PV-hydro-wind integrated power network, Electr. Power Syst. Res. 214 (2023) 108924.

[79] S. Panda, B.K. Sahu, P.K. Mohanty, Design and performance analysis of PID controller for an automatic voltage regulator system using simplified particle swarm optimization, J. Franklin Inst. 349 (2012) 2609–2625, http://dx.doi.org/10.1016/j.jfranklin.2012.06.008.

[80] P.K. Mohanty, B.K. Sahu, S. Panda, Tuning and assessment of proportional–Integral–derivative controller for an automatic voltage regulator system employing local unimodal sampling algorithm, Electr. Power Compon. Syst. 42 (2014) 959–969, http://dx.doi.org/10.1080/15325008.2014.903546.

[81] W. Aribowo, B. Suprianto, U.T. Kartini, Cooperation search algorithm for tuning automatic voltage regulator system, Int. J. Power Electron. Drive Syst. 13 (2022) 1761–1769.

[82] H.M. Hasanien, Design optimization of PID controller in automatic voltage regulator system using taguchi combined genetic algorithm method, IEEE Syst. J. 7 (2013) 825–831, http://dx.doi.org/10.1109/jsyst.2012.2219912.

[83] S. Alghamdi, H.F. Sindi, M. Rawa, A.A. Alhussainy, M. Calasan, M. Micev, Z.M. Ali, S.H.E. Abdel Aleem, Optimal PID controllers for AVR systems using hybrid simulated annealing and gorilla troops optimization, Fractal Fract. 6 (2022) 682.

[84] D.H. Kim, Hybrid GA–BF based intelligent PID controller tuning for AVR system, Appl. Soft Comput. 11 (2011) 11–22, http://dx.doi.org/10.1016/j.asoc.2009.01.004.

[85] S. Ekinci, B. Hekimoglu, Improved kidney-inspired algorithm approach for tuning of PID controller in AVR system, IEEE Access 7 (2019) 39935–39947, http://dx.doi.org/10.1109/access.2019.2906980.

[86] Z. Bingul, O. Karahan, A novel performance criterion approach to optimum design of PID controller using cuckoo search algorithm for AVR system, J. Franklin Inst. 355 (2018) 5534–5559, http://dx.doi.org/10.1016/j.jfranklin.2018.05.056.

[87] D. Izci, S. Ekinci, H. Lale Zeynelgil, Controlling an automatic voltage regulator using a novel harris hawks and simulated annealing optimization technique, in: Adv. Control Appl. Eng. Ind. Syst., e121, (n.d.).

[88] A.M. Mosaad, M.A. Attia, A.Y. Abdelaziz, Whale optimization algorithm to tune PID and PIDA controllers on AVR system, Ain Shams Eng. J. 10 (2019) 755–767, http://dx.doi.org/10.1016/j.asej.2019.07.004.

[89] O.A.M. Omar, M.I. Marei, M.A. Attia, Comparative study of AVR control systems considering a novel optimized PID-based model reference fractional adaptive controller, Energies 16 (2023) 830.

[90] E. Çelik, R. Durgut, Performance enhancement of automatic voltage regulator by modified cost function and symbiotic organisms search algorithm, Eng. Sci. Technol. Int. J. 21 (2018) 1104–1111, http://dx.doi.org/10.1016/j.jestch.2018.08.006.

[91] E. Çelik, N. Öztürk, E.H. Houssein, Improved load frequency control of interconnected power systems using energy storage devices and a new cost function, Neural Comput. Appl. (2022) 1–17.

[92] T.A. Jumani, M.W. Mustafa, Z. Hussain, M. Md. Rasid, M.S. Saeed, M.M. Memon, I. Khan, K.S. Nisar, Jaya optimization algorithm for transient response and stability enhancement of a fractional-order PID based automatic voltage regulator system, Alex. Eng. J. 59 (2020) 2429–2440, http://dx.doi.org/10.1016/j.aej.2020.03.005.

[93] J. Bhookya, R.K. Jatoth, Improved Jaya algorithm-based FOPID/PID for AVR system, COMPEL - Int. J. Comput. Math. Electr. Electron. Eng. 39 (2020) 775–790, http://dx.doi.org/10.1108/COMPEL-08-2019-0319.

[94] G.Q. Zeng, J. Chen, Y.X. Dai, L.M. Li, C.W. Zheng, M.R. Chen, Design of fractional order PID controller for automatic regulator voltage system based on multi-objective extremal optimization, Neurocomputing 160 (2015) 173–184, http://dx.doi.org/10.1016/j.neucom.2015.02.051.

[95] D. Izci, S. Ekinci, H. Çetin, Arithmetic optimization algorithm based controller design for automatic voltage regulator system, in: 2022 Innov. Intell. Syst. Appl. Conf., 2022, pp. 1–5.

[96] A.K. Bhullar, R. Kaur, S. Sondhi, Enhanced crow search algorithm for AVR optimization, Soft Comput. 24 (2020) 11957–11987, http://dx.doi.org/10.1007/s00500-019-04640-w.

[97] R. Mok, M.A. Ahmad, Fast and optimal tuning of fractional order PID controller for AVR system based on memorizable-smoothed functional algorithm, Eng. Sci. Technol. Int. J. 35 (2022) 101264.

[98] M. Micev, M. Ćalasan, D. Oliva, Design and robustness analysis of an automatic voltage regulator system controller by using equilibrium optimizer algorithm, Comput. Electr. Eng. 89 (2021) 106930, http://dx.doi.org/10.1016/j.compeleceng.2020.106930.

[99] M. Ćalasan, M. Micev, Ž. Djurovic, H.M.A. Mageed, Artificial ecosystem-based optimization for optimal tuning of robust PID controllers in AVR systems with limited value of excitation voltage, Int. J. Electr. Eng. Educ. (2020) 002072092094060, http://dx.doi.org/10.1177/0020720920940605.

[100] N. Pachauri, Water cycle algorithm-based PID controller for AVR, COMPEL - Int. J. Comput. Math. Electr. Electron. Eng. 39 (2020) 551–567, http://dx.doi.org/10.1108/COMPEL-01-2020-0057.

[101] A. Idir, L. Canale, Y. Bensafia, K. Khettab, Design and robust performance analysis of low-order approximation of fractional PID controller based on an IABC algorithm for an automatic voltage regulator system, Energies 15 (2022) 8973.

[102] M.S. Ayas, E. Sahin, FOPID controller with fractional filter for an automatic voltage regulator, Comput. Electr. Eng. (2020) 106895, http://dx.doi.org/10.1016/j.compeleceng.2020.106895.

[103] M. Micev, M. Ćalasan, Z.M. Ali, H.M. Hasanien, S.H.E.E. Abdel Aleem, Optimal design of automatic voltage regulation controller using hybrid simulated annealing – Manta ray foraging optimization algorithm, Ain Shams Eng. J. 12 (2021) 641–657, http://dx.doi.org/10.1016/j.asej.2020.07.010.

[104] Z.-L. Gaing, A particle swarm optimization approach for optimum design of PID controller in AVR system, IEEE Trans. Energy Convers. 19 (2004) 384–391, http://dx.doi.org/10.1109/tec.2003.821821.

[105] A.M. Mosaad, M.A. Attia, A.Y. Abdelaziz, Comparative performance analysis of AVR controllers using modern optimization techniques, Electr. Power Compon. Syst. 46 (2018) 2117–2130, http://dx.doi.org/10.1080/15325008.2018.1532471.

[106] D.-L. ZHANG, Y.-G. TANG, X.-P. GUAN, Optimum design of fractional order PID controller for an AVR system using an improved artificial bee colony algorithm, Acta Autom. Sin. 40 (2014) 973–979, http://dx.doi.org/10.1016/s1874-1029(14)60010-0.

[107] H. Gozde, M.C. Taplamacioglu, Comparative performance analysis of artificial bee colony algorithm for automatic voltage regulator (AVR) system, J. Franklin Inst. 348 (2011) 1927–1946, http://dx.doi.org/10.1016/j.jfranklin.2011.05.012.

[108] M.J. Blondin, P. Sicard, P.M. Pardalos, Controller tuning approach with robustness, stability and dynamic criteria for the original AVR system, Math. Comput. Simulation 163 (2019) 168–182, http://dx.doi.org/10.1016/j.matcom.2019.02.019.

[109] M. Zamani, M. Karimi-Ghartemani, N. Sadati, M. Parniani, Design of a fractional order PID controller for an AVR using particle swarm optimization, Control Eng. Pract. 17 (2009) 1380–1387, http://dx.doi.org/10.1016/j.conengprac.2009.07.005.

[110] A. Chatterjee, V. Mukherjee, S.P. Ghoshal, Velocity relaxed and craziness-based swarm optimized intelligent PID and PSS controlled AVR system, Int. J. Electr. Power Energy Syst. 31 (2009) 323–333, http://dx.doi.org/10.1016/j.ijepes.2009.03.012.

[111] M.E. Ortiz-Quisbert, M.A. Duarte-Mermoud, F. Milla, R. Castro-Linares, G. Lefranc, Optimal fractional order adaptive controllers for AVR applications, Electr. Eng. 100 (2016) 267–283, http://dx.doi.org/10.1007/s00202-016-0502-2.

[112] H. Zhu, L. Li, Y. Zhao, Y. Guo, Y. Yang, CAS algorithm-based optimum design of PID controller in AVR system, Chaos Solitons Fractals 42 (2009) 792–800, http://dx.doi.org/10.1016/j.chaos.2009.02.006.

[113] M.J. Blondin, J. Sanchis, P. Sicard, J.M. Herrero, New optimal controller tuning method for an AVR system using a simplified Ant Colony Optimization with a new constrained Nelder–Mead algorithm, Appl. Soft Comput. 62 (2018) 216–229, http://dx.doi.org/10.1016/j.asoc.2017.10.007.

[114] A. Sikander, P. Thakur, R.C. Bansal, S. Rajasekar, A novel technique to design cuckoo search based FOPID controller for AVR in power systems, Comput. Electr. Eng. 70 (2018) 261–274, http://dx.doi.org/10.1016/j.compeleceng.2017.07.005.

[115] Y. Tang, M. Cui, C. Hua, L. Li, Y. Yang, Optimum design of fractional order PIλdμ controller for AVR system using chaotic ant swarm, Expert Syst. Appl. 39 (2012) 6887–6896, http://dx.doi.org/10.1016/j.eswa.2012.01.007.

[116] M.A. Sahib, B.S. Ahmed, A new multiobjective performance criterion used in PID tuning optimization algorithms, J. Adv. Res. 7 (2016) 125–134, http://dx.doi.org/10.1016/j.jare.2015.03.004.

[117] I. Pan, S. Das, Chaotic multi-objective optimization based design of fractional order PIλdμ controller in AVR system, Int. J. Electr. Power Energy Syst. 43 (2012) 393–407, http://dx.doi.org/10.1016/j.ijepes.2012.06.034.

[118] I. Pan, S. Das, Frequency domain design of fractional order PID controller for AVR system using chaotic multi-objective optimization, Int. J. Electr. Power Energy Syst. 51 (2013) 106–118, http://dx.doi.org/10.1016/j.ijepes.2013.02.021.