Research Article

An Adaptive Particle Swarm Optimization Algorithm for Unconstrained Optimization

Feng Qian,¹ Mohammad Reza Mahmoudi ,² Hamid Parvin ,^{3,4,5} Kim-Hung Pho,⁶ and Bui Anh Tuan⁷

¹*School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, China*

²*Department of Statistics, Faculty of Science, Fasa University, Fasa 7461, Iran*

³*Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam*

⁴*Faculty of Information Technology, Duy Tan University, Da Nang 550000, Vietnam*

⁵*Department of Computer Science, Nourabad Mamasani Branch, Islamic Azad University, Mamasani 7351, Iran*

⁶*Fractional Calculus, Optimization and Algebra Research Group, Faculty of Mathematics and Statistics, Ton Duc Thang University, Ho Chi Minh, Vietnam*

⁷*Department of Mathematics Education, Teachers College, Can Tho University, Can Tho, Vietnam*

Correspondence should be addressed to Hamid Parvin; parvin@alumni.iust.ac.ir

Received 8 December 2019; Revised 15 March 2020; Accepted 28 April 2020; Published 9 September 2020

Academic Editor: Carlos Aguilar-Ibanez

Copyright © 2020 Feng Qian et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Conventional optimization methods are not efficient enough to solve many of the naturally complicated optimization problems. Thus, inspired by nature, metaheuristic algorithms can be utilized as a new kind of problem solvers in solution to these types of optimization problems. In this paper, an optimization algorithm is proposed which is capable of finding the expected quality of different locations and also tuning its exploration-exploitation dilemma to the location of an individual. A novel particle swarm optimization algorithm is presented which implements the conditioning learning behavior so that the particles are led to perform a natural conditioning behavior on an unconditioned motive. In the problem space, particles are classified into several categories so that if a particle lies within a low diversity category, it would have a tendency to move towards its best personal experience. But, if the particle's category is with high diversity, it would have the tendency to move towards the global optimum of that category. The idea of the birds' sensitivity to its flying space is also utilized to increase the particles' speed in undesired spaces in order to leave those spaces as soon as possible. However, in desirable spaces, the particles' velocity is reduced to provide a situation in which the particles have more time to explore their environment. In the proposed algorithm, the birds' instinctive behavior is implemented to construct an initial population randomly or chaotically. Experiments provided to compare the proposed algorithm with the state-of-the-art methods show that our optimization algorithm is one of the most efficient and appropriate ones to solve the static optimization problems.

1. Introduction

Optimization is a subfield of artificial intelligence [1–10] in which a raised problem could be transformed into an optimization function with a random initial solution which is improved through the next designed steps [1, 2]. Since there is more than one solution to a problem, we can receive the best response to an arbitrary problem using mathematical global optimization tools [11]. In order to determine the best

answer in local optimization, some factors should be taken into account including the solution, the amount of the permissible error, and the problems [12–14] such as the best work of art, the most beautiful landscapes, and the most pleasant piece of music. Some popular stochastic algorithms, such as naturally motivated ones and population-based ones, emulate problem-solving strategies in nature including the race of some creatures with the main goal of survival which makes these creatures to be evolved in several shapes. An

artificial intelligence technique based on the collective behavior in decentralized and self-organized systems, which usually includes a number of simple agents locally interacting with each other and with the environment they are in, is swarm intelligence (SI). In these algorithms, although there is not a usual centralized control for the agents' behaviors, a collective behavior is often provided due to the local interactions of the agents.

Pavlov, a Russian physiologist, recognized that the dogs splashed their saliva only when they see someone who had fed them before even if those people had not any food in their hand the next time. Based on this observation, he proposed the conditioning learning in which the animal finds out how to relate a reward or punishment with a natural motive. This theory is used to move the swarms with high spatial distribution diversity to the global optimum and also to move the swarms with low level of diversity to the local optima in the problem space. In addition, in order to modify the proposed algorithm to an improved version of the PSO algorithm, some other ideas related to the instinctive behavior of the birds and their speed are considered.

There are some defects in the original PSO algorithm including the following instances: first, this algorithm is weak in creation of the random initial population; second, quality of the problem space is not considered, and speed of the particles with that quality is not adjusted; third, PSO is time wasting to achieve an optimal solution by choosing a point between the local and global optima. These mentioned drawbacks are all resolved in this work using the instinctive conditioning behavior of the birds. The rest of this paper is organized as follows. In Section 2, a summary of the related previous works is outlined. The proposed algorithm of the paper is presented in Section 3. Simulation results and the consequences are provided in Section 4, and Section 5 deals with the conclusions and future works.

2. Literature

There are two groups of the naturally motivated meta-heuristic optimization algorithms. The first one is single-solution-based algorithms which offer a gradually improving single random solution. The second group, which is more popular in the literature, is called the multisolution-based algorithm. The group which we focus on offers multiple solutions that are gradually enhanced as a whole.

One of the best evolutionary algorithms, proposed by Holland [15] in 1975, is the genetic algorithm (GA) which is a random evolutionary optimization algorithm inspired by the evolution of creatures in nature. This is a general purpose optimizer because it has an extraordinary optimization performance. At the beginning of this algorithm, an initial population of the chromosomes or potential solutions which is usually a chromosome or a binary vector is considered and then a relevant fitness function is used to measure the merit of each chromosome which results in production of a new solution set [16] by choosing the best chromosomes and entering them into a mating stage where they are crossed over and mutated.

Similar to other related articles, we can show that optimization problems of this work lead to cost function

minimizations. Using instinctive and collective behavior of the flying birds, a naturally motivated optimization algorithm is provided, whose mechanism is similar to that of the particle swarm optimization algorithm. Conditioning learning behavior of the animals, which is the basic of this article and is a type of consistent learning, is expected to result in a mechanism to be able to solve any complex problem. The proposed algorithm offers some useful advantages. One advantage in the proposed PSO version is that this algorithm utilizes a beneficial collective behavior inspired by nature, and another one is that in comparison with other optimization algorithms, many of its specific parameters can be automatically set without significant loss of performance. But, a drawback of this proposed algorithm is that it is highly probable that an improper problem for the original PSO version is an improper problem for the proposed optimization algorithm.

The social behavior of birds to search for food had been the motive for proposing of a population-based general purpose optimization algorithm in 1995 [17], namely, PSO, which is a computation-oriented random intelligence optimization algorithm. Many advantages, such as computational efficiency, search mechanism, simple concept, and ease of implementation, have introduced SI-based algorithms as helpful tools in optimization applications. Each particle in PSO represents a population member with small or low-volume random mass which induces reaching a better behavior with speed and acceleration. Therefore, a particle could be a solution in a multidimensional space which can set its position in the search space based on the best position it has reached by itself (pbest), the best position reached by swarm (gbest) during the search process, and its speed. SI-based method has been used in some applications of the PSO algorithm [18–20] to calculate the trajectory in the binary search space to solve the knapsack problem (KP). Some problems could also be solved using a combination of PSO with guided local search and some concepts derived from the GA [21]. A binary PSO algorithm has been proposed to solve the KP using the mutation operator.

Optimization algorithms motivated by the bees' behavior result in some SI algorithms different from PSO; the most well-known one is called the artificial bee colony (ABC) optimization algorithm. It is a prevalent optimization algorithm developed from the bees' SI foraging behavior [22]. ABC algorithm includes the artificial bee colony containing three types of worker, onlooker, and scout bees. Waiting in a dance area, onlooker bees create and choose a food source, worker bees go to the food sources which are previously recognized, and the scout bees make a random search for new food sources. A possible solution to the optimization problem is represented by location of a food source, and quality of the solution depends on the quantity of the nectar from the food source. Also, a virtual swarm of bees moving randomly in the two-dimensional search space is supposed, and when some target solution is found, bees have interacted with each other. Some other methods are derived from the original ABC method. Job-shop scheduling problem [23] (JSSP) has been presented when behavioral change of the onlooker behavior has changed, and SI is used

to convert continuous values to binary ones. Also, the combinatorial ABC (CABC) algorithm has been presented [24] with discrete coding for the travel salesman path (TSP) problem.

Another algorithm, namely, the bees algorithm (BA), introduced on a mathematical function [25], includes a D -dimensional bee vector, which corresponds to the problem variables, and is named as a candidate solution here, which represents a site (food source) visit possessing a certain fitness value. In BA, scout bees randomly searching for new sites and worker bees frequently searching the neighbors for higher values of fitness function balance the exploration-exploitation of the algorithm. Best bees are those with the best fitness values, and elite sites are the sites visited by them. This algorithm assigns the majority of the bees near to the best sites to search neighbors of the best selected sites. Adjoined optimization functions, scheduling tasks [26], and binary data clustering [27] are examples of BA applications.

A metaheuristic algorithm searching for suitable conditions during improvisation of jazz music in the natural process of musical performance is called the harmony search algorithm (HSA) [28]. An assigned form by the standard of aesthetics is searched by the improvisation to find harmony in a piece of jazz music (a proper condition), which is equivalent to an optimization process in which a global solution (a proper condition) found by a specified objective function is searched.

Motivated by the behavior of a set of imperialists who compete with each other to overcome the colonies, an evolutionary optimization technique called the imperialist competitive algorithm (ICA) has been introduced. This algorithm also begins with an initial population whose individual members are classified into two groups of colonists and imperialists. Power of the colonies assigns them to their related imperialists. Power of any state is reversely related with its cost, so that the imperialists with more power are more dominant [29]. Interaction of the imperialist powers and their colonies has a characteristic in which the culture of the colonies is changed after a while so that they gradually become similar to one of their ruling imperialists. This is named the attraction policy which means the colonies move towards their imperialists and is executed by imperialist countries after the nineteenth century. In recent years, many other algorithms in line with improving famous optimization algorithms such as the particle bee optimization algorithm (PBOA) [30], novel particle swarm optimization algorithm (NPSOA) [31], cuckoo search optimization algorithm (CSOA or CS) [32], differential search optimization algorithm (DSOA or DSA) [33], and bird mating optimization algorithm (BMOA or BMO) [34] are presented. In order to improve the PSO algorithm [35], many algorithms have been presented using a dynamic multipopulation method. Some of these algorithms are the sinusoidal differential evolution optimization algorithm (SDEOA) [36], joint operation optimization algorithm (JOOA) [37], and dynamic multiswarm particle swarm optimizer with cooperative algorithm (DMSPSOCA) [35].

In order to make different algorithms more efficient, parameters and ideas from other methods could be used. For

instance, the CSOA could be enhanced by employing chaos parameters [38]. In 2015, the PSO algorithm was enhanced employing some cognitive learning mechanisms for solving an optimization problem [39]. In 2016, the ant colony optimization algorithm (ACOA) was integrated with GA, and a new optimization algorithm was introduced [40]. A study used GA to discover the closest solution to the best one to deal with nonlinear multimodal optimization problems [41]. There are also other methods among the latest comparable optimizers [31, 42–45].

3. Adaptive PSO

The proposed method is presented in this section. The population is initialized at first. After that, the population is randomly partitioned into a set of N subpopulations. Also, the problem space is classified into several virtual subspaces. Each subspace is a hypercube. Indeed, each dimension out of all of the D dimensions is partitioned into θ equal-size slices. Therefore, we have θ^D subspaces. Then, the particles will be transferred with lower speed in the more valuable subspaces. Also, in each subpopulation, a special set of the movement coefficients (C_1 and C_2) is used. Also, the special set of the movement coefficients for each subpopulation adaptively changes during optimization. Finally, the best solution produced during optimization is considered to be the optimal solution to the problem found by the proposed optimization algorithm called also as the adaptive particle swarm optimization algorithm (APSOA).

The pseudocode of the APSOA is depicted in Algorithm 1. Variables PS , N , α , D , θ , MG , $MaxV$, $MinV$, and function F are inputs of this algorithm. Here, $MaxV$ and $MinV$ are up-bound and low-bound vectors of the problem space. D is the number of dimensions. PS is the population size. N is the number of subpopulations. Finally, F shows the objective function. First, the algorithm calls the initialization function, denoted by $\mathbb{I}$. The pseudocode of the initialization function is depicted in Algorithm 2. It takes variables PS , D , $MaxV$, $MinV$, θ , and function F . It returns a population $\hat{I}$, the best particle $\hat{g}$, and a sparse subspace rating vector $\overline{SSR}$. $\overline{SSR}$ is an array whose key is a string. If it is called with a numeric string s as its key, it will return the value of the s th subspace where the value of s is an integer in $\{1, 2, \dots, \theta^D\}$. As it is a sparse subspace rate array, if a subspace has still no value, we assume its value as 1 by default. If $\overline{SSR}$ is called with “sum” as its key, it will return the summation of all values associated with all subspaces. Indeed, $\overline{SSR}('sum')$ is θ^D at first. Each time the best particle of a subpopulation is located in the s th subspace, its value is added by one unit. Also, each time the best particle of the whole population is located in the s th subspace, its value is added by one unit. $\hat{I}$ is a population of PS individuals, where $\hat{I}_i$ stands for the i th individual in population $\hat{I}$. Each individual in population is an object containing the following fields:

$\hat{I}_i \cdot x_d$: the position of the i th individual

$\hat{I}_i \cdot v_d$: the velocity of the i th individual

APSO: adaptive particle swarm optimization function

Input:

PS: population size

N: subpopulation Size

α : coefficient update rate

D: problem size

θ : fragmentation size

MG: maximum generations of the algorithm

MaxV: an array of D values; $MaxV_d$ is the maximum value in the domain of the d th dimension of the problem space

MinV: an array of D values; $MinV_d$ is the minimum value in the domain of the d th dimension of the problem space

C_1 : an array of N values; $C_1(i)$ is the first movement coefficient of the i th subpopulation

C_2 : an array of N values; $C_2(i)$ is the second movement coefficient of the i th subpopulation

F: a given objective function

Output:

$\hat{g}$: the best found particle

(1) $[\hat{I}, \hat{g}, \overline{SSR}] = II(PS, D, MaxV, MinV, \theta, F)$

(2) For $i = 1 : MG$

(2.1) For $p = 1 : PS$

(2.1.1) $\delta = \lceil [p/PS] \times N \rceil$

(2.1.2) $I_p = U_I(I_p, \hat{g}, D, C_1(\delta), C_2(\delta), \theta, \overline{SSR}, MaxV, MinV, F)$

(2.1.3) $temp1(\delta) = N \times (\hat{I}_p \cdot x)^2 / PS$

(2.1.4) $temp2(\delta) = N \times \hat{I}_p \cdot x / PS$

(2.2) $i = \arg \max_{i \in \{L, L+1, \dots, U\}} \hat{I}_i \cdot \hat{f}$

(2.3) $key = num2str(I_i \cdot \varphi)$

(2.4) $\hat{g} = \hat{I}_i$

(2.5) $HasValue(\overline{SSR}(key)) \rightarrow \overline{SSR}(key) = \overline{SSR}(key) + 1 \cdot \overline{SSR}(key) = 2$

(2.6) $\overline{SSR}('sum') = \overline{SSR}('sum') + 1$

(2.7) $L = 1$

(2.8) For $s = 1 : N$

(2.8.1) $U = S \times PS/N$

(2.8.2) $i = \arg \max_{i \in \{L, L+1, \dots, U\}} \hat{I}_i \cdot \hat{f}$

(2.8.3) $key = num2str(\hat{I}_i \cdot \varphi);$

(2.8.4) $HasValue(key) \rightarrow \overline{SSR}(key) = \overline{SSR}(key) + 1 \cdot \overline{SSR}(key) = 2$

(2.8.5) $\overline{SSR}('sum') = \overline{SSR}('sum') + 1$

(2.8.6) $L = U$

(2.8.7) $\sigma(s) = temp1(s) - (temp1(s))^2$

(2.9) For $s = 1 : N$

(2.9.1) $[C_1(\delta), C_2(\delta)] = \mathbb{U}_C(C_1(\delta), C_2(\delta), \alpha, \sigma(s) < Med(\sigma))$

ALGORITHM 1: Pseudocode of the proposed APSOA.

$\hat{I}_i \cdot \bar{x}_d$: the position of the best place met by the i th individual

$\hat{I}_i \cdot \varphi$: index of the subspace in which the i th individual is located

$\hat{I}_i \cdot f$: fitness of the i th individual

$\hat{I}_i \cdot \bar{f}$: fitness of the best memory of the i th individual

After the proposed method (depicted in Algorithm 1) called the initialization function, it iterates a loop MG times (statement 2 in Algorithm 1). Each time, all population individuals are updated using an individual update function $\mathbb{U}_I$ (depicted in Algorithm 3 and explained in the following paragraph) (statement 2.1 in Algorithm 1). After updating all population individuals, the global best, i.e., $\hat{g}$, will be updated. Then, $\overline{SSR}$ of the subspace where the global best is located is updated (statement 2.2 to statement 2.6 in Algorithm 1). In the following step, $\overline{SSR}$ of the subspaces where the local bests of subpopulations are located is updated (statement 2.8 in Algorithm 1). Finally, the

coefficients of any subpopulation are updated using the coefficient update function $\mathbb{U}_C$ (depicted in Algorithm 4 and explained in the following paragraph) (statement 2.9 in Algorithm 1).

For the individual update function $\mathbb{U}_I$, which is depicted in Algorithm 3, the only different section from the original PSO is the statement 2.2 in Algorithm 3 where a coefficient (denoted by β) is computed to be multiplied in the movement equation in the following statement. It speeds up the individuals in the useless subspaces and slows them down in the useful subspaces. The coefficient update function $\mathbb{U}_C$, which is depicted in Algorithm 4, adaptively changes the values of coefficients of each subpopulation.

4. Implementation of Experimentations

In this section, simulation results of the proposed method are provided in several parts, and they are compared with other similar methods. The results are provided in four parts, and

$\mathbb{I}$: initialization function

Input:

PS : population size

D : problem size

$MaxV$: an array of D values; $MaxV_d$ is the maximum value in the domain of the d th dimension of the problem space

$MinV$: an array of D values; $MinV_d$ is the minimum value in the domain of the d th dimension of the problem space

θ : fragmentation size

$\mathbb{F}$: a given objective function

Output:

$\hat{\mathbb{I}}$: a population

$\hat{g}$: global best particle

$\overline{SSR}$: sparse subspace rate array

(1) $T = [0/\theta, 1/\theta, \dots, \theta - 1/\theta, \theta/\theta + \epsilon]$

(2) For $i = 1 : PS$

(2.1) $\varphi = 0$

(2.2) For $d = 1 : D$

(2.2.1) $r = A \text{ random or chaotic value from uniform distribution in interval } [0, 1]$

(2.2.2) $\hat{I}_i.x_d = r \times (MaxV_d - MinV_d) + MinV_d$

(2.2.2) $v = \hat{I}_i.x_d - MinV_d / MaxV_d - MinV_d$

(2.2.2) $\forall j \in \{1, 2, \dots, \theta\} \bullet (T(j) \leq v < T(j+1) \longrightarrow \varphi = \varphi + \theta + j)$

(2.2.3) $\hat{I}_i.\bar{x}_d = \hat{I}_i.x_d$

(2.2.4) $r = A \text{ random or chaotic value from uniform distribution in interval } [0, 1]$

(2.2.5) $\hat{I}_i.v_d = r \times (MaxV_d - MinV_d) + MinV_d$

(2.3) $\hat{I}_i.\varphi = \varphi$

(2.4) $\hat{I}_i.f = \mathbb{F}(\hat{I}_i.x)$

(2.5) $\hat{I}_i.\bar{f} = \hat{I}_i.f$

(3) $i = \arg \max_{i \in \{1, 2, \dots, PS\}} \hat{I}_i.\bar{f}$

(4) $key = num2str(\hat{I}_i.\varphi)$

(5) $\hat{g} = \hat{I}_i$

(6) $\overline{SSR}(key) = 2$

(7) $\overline{SSR}('sum') = \theta^D + 1$

ALGORITHM 2: Pseudocode of the population initialization.

$\mathbb{U}_I$: velocity and location update function

Input:

$\hat{I}$: an individual or particle

g : global best particle

D : problem size

C_1 : first coefficient

C_2 : second coefficient

θ : fragmentation size

$\overline{SSR}$: sparse subspace rate array

$MaxV$: an array of D values; $MaxV_d$ is the maximum value in the domain of the d th dimension of the problem space

$MinV$: an array of D values; $MinV_d$ is the minimum value in the domain of the d th dimension of the problem space

$\mathbb{F}$: a given objective function

Output:

$\hat{I}$: an individual or particle

(1) $T = [0/\theta, 1/\theta, \dots, \theta - 1/\theta, \theta/\theta + \epsilon]$

(2) For $d = 1 : D$

(2.1) $[r_1, r_2] = \text{two random or chaotic values of uniform distribution in interval } [0, 1]$

(2.2) $HasValue(\overline{SSR}(num2str(\hat{I}.\varphi))) \longrightarrow \beta = (1 - \overline{SSR}(num2str(\hat{I}.\varphi)) / \overline{SSR}('sum')) \cdot \beta = (1 - 1 / \overline{SSR}('sum'))$

(2.3) $\hat{I}.v_d = \beta \times [w \times \hat{I}.v_d + C_1 \times r_1 \times (\hat{I}.\bar{x}_d - \hat{I}.x_d) + C_2 \times r_2 (\hat{g}.x_d - \hat{I}.x_d)]$

(2.4) $\hat{I}.x_d = \max(\min(\hat{I}.x_d + \hat{I}.v_d, MaxV_d), MinV_d)$

(2.5) $v = \hat{I}.x_d - MinV_d / MaxV_d - MinV_d$

(2.6) $\forall j \in \{1, 2, \dots, \theta\} \bullet (T(j) \leq v < T(j+1) \longrightarrow \varphi = \varphi + \theta + j)$

(3) $\hat{I}_i.\varphi = \varphi$

(4) $\hat{I}_i.f = \mathbb{F}(\hat{I}_i.x)$

(5) if $(\hat{I}_i.\bar{f} < \hat{I}_i.f)$

(5.1) $\hat{I}_i.\bar{f} = \hat{I}_i.f$

(5.2) $\hat{I}_i.\bar{x} = \hat{I}_i.x$

ALGORITHM 3: Pseudocode of the population update.

$\mathbb{U}_C$: coefficient update function

Input:

C_1 : first coefficient

C_2 : second coefficient

α : coefficient change rate

π : an exploration flag

Output:

C_1 : first coefficient

C_2 : second coefficient

(1) if π

(1.1) $C_1 = \alpha \times C_1$

(1.2) $C_2 = c_2/\alpha$

else

(1.3) $C_1 = c_1/\alpha$

(1.4) $C_2 = \alpha \times C_2$

ALGORITHM 4: Pseudocode of updating the coefficients of the algorithm.

different modern methods have been compared in each part on different problems which include a real-world industrial application.

In the first three parts, the average value and standard deviation of the cost function value, i.e., $F_i(x) - F^*$, are calculated to evaluate the performance of each algorithm, where $F_i(x)$ is the minimum value (of the i th objective function) found by an algorithm in a run and F^* is the actual optimal value (of the i th objective function).

4.1. Experimental Results: CEC 2009. Here, problems defined by the CEC 2009 benchmark [46] are provided in Table 1. By comparison with the proposed method performance when the chaotic number generator (CNG) is used against the proposed method performance when the random number generator (RNG) is used (on objective functions of the CEC 2009 benchmark), we concluded that slightly better results are obtained when CNG is used. We only used the logistic map [47] with parameter $\alpha = 4$ (as CNG) and uniform distribution (as RNG). Therefore, though we can use both throughout this paper, we totally experimented using the logistic map [47] with parameter $\alpha = 4$ (as CNG).

The proposed method in this part is compared with the following algorithms: GA [15], differential evolution (DE) [48], PSO [17], BA [25], PBOA [30], NPSO [31], moderate-random-search strategy PSO (MRPSO) [49], ensemble of mutation strategies and control parameter S with DE (EPSDE) [50], cooperative coevolution inspired ABC (CCABC) [51], and firefly algorithm (FA) [52]. The results of each method are derived for the same parameters used by its main corresponding method. But, some of the shared parameters are fixed in all methods including the population size (PS) and the number of fitness evaluations (NFE), respectively, equal to 40 and $10,000 \times D$, where D is the problem size and is always set to 50. We set the target objective cost function (denoted by CF) always as $CF_i(x) = F_i(x) - F_i^*$ where i is an integer in $\{1, 2, \dots, 26\}$ and F_i^* is the optimal target value of the i th objective function. We

TABLE 1: Summary of test functions in the CEC 2009 benchmark.

Function	Domain
F1 (Beale)	$[-4.5, 4.5]^D$
F2 (Easom)	$[-100, 100]^D$
F3 (Matyas)	$[-10, 10]^D$
F4 (Colville)	$[-10, 10]^D$
F5 (Zakharov)	$[-5, 10]^D$
F6 (Schwefel 2.22)	$[-10, 10]^D$
F7 (Schewefel 1.2)	$[-100, 100]^D$
F8 (Dixon-price)	$[-10, 10]^D$
F9 (Step)	$[-5.12, 5.12]^D$
F10 (Sphere)	$[-100, 100]^D$
F11 (SumSquares)	$[-10, 10]^D$
F12 (Quartic)	$[-1.28, 1.28]^D$
F13 (Schaffer)	$[-100, 100]^D$
F14 (6 H Camel)	$[-5, 5]^D$
F15 (Bohachevsky2)	$[-100, 100]^D$
F16 (Bohachevsky3)	$[-100, 100]^D$
F17 (Shubert)	$[-10, 10]^D$
F18 (Rosenbrock)	$[-30, 30]^D$
F19 (Griewank)	$[-600, 600]^D$
F20 (Ackley)	$[-32, 32]^D$
F21 (Bohachevsky1)	$[-100, 100]^D$
F22 (Booth)	$[-10, 10]^D$
F23 (Michalewicz2)	$[0, \pi]^D$
F24 (Michalewicz5)	$[0, \pi]^D$
F25 (Michalewicz10)	$[0, \pi]^D$
F26 (Rastrigin)	$[-5.12, 5.12]^D$

have used mean and std. dev as two criteria in presenting the results which are, respectively, defined as the average of the best costs of the particles and the standard deviation of the best costs of the particles in different runs on all of the 26 functions shown in Table 1. Table 2 shows the results for assessment of other methods on all of the 26 functions shown in Table 1 in detail; also, it validates the results using the Friedman test with a P value of $3.12E-03$. The results presented in Table 2 have been summarized in Table 3.

The results show that, in almost all of the functions in the CEC 2009 benchmark, the proposed method converges to the optimum as the particles' average cost is counted. Since the proposed algorithm fully converges to the optimal point, the particles will have a zero standard deviation at the end of the assessments in different methods which means that the results of the proposed method together with other methods have presented the best performance. The results also show that there are some functions such as F8, F12, and F18 in which only the proposed method achieves the optimal point which proves the superiority of the proposed method. In the process of the Friedman test, the results' process is proved to be normal, and it is proved that those results are not obtained by random.

4.2. Experimental Results: CEC 2005. The proposed method in this part is compared with the following algorithms: the modified bat algorithm hybridizing by differential evolution (MBADE [10]), rain optimization algorithm (ROA) [42], CS [32], teaching-learning-based optimization (TLBO) algorithm [53], DSA [33], and BMO algorithm [34]. In this part, we have compared the proposed method based on 25 problems defined by the CEC 2005 benchmark [54]. All of

TABLE 2: Assessment of different methods in numerical optimization problems of the CEC 2009 benchmark.

Method	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}	F_{17}	F_{18}	F_{19}	F_{20}	F_{21}	F_{22}	F_{23}	F_{24}	F_{25}	F_{26}	F_{27}	F_{28}	F_{29}	F_{30}	
BBNBO	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	
PSO	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
GA	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
DE	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
CS	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
QABC	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
PSOA	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
MONA	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
MONA	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
MONA	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
MONA	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
MONA	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
MONA	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
MONA	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
MONA	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
MONA	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
MONA	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
MONA	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
MONA	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
MONA	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
MONA	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
MONA	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.	

The value V_1 & $(V_2 \pm V_3)$ below F_{V_4} means that the average of the best values found by the method. $\mathcal{M}$ on 30 independent runs over function F_{V_4} is V_2 and the standard deviation is V_3 . It also means that the rank of the method $\mathcal{M}$ among all methods over function F_{V_1} is V_1 . The sign & indicates that the proposed method performance is superior/inferior/equal to that of the method $\mathcal{M}$, if its value is $+/-/\times$.

TABLE 3: Summary of the results presented in Table 2 (average rank of each method on all functions of the CEC 2009 benchmark).

GA	MRPSO	EPSDE	PSO	FA	BA	CCABC	DE	PBOA	NPSO	APSOA
7.08 (184)	4.42 (115)	4.04 (105)	3.62 (94)	3.50 (91)	3.38 (88)	3.35 (87)	2.81 (73)	2.38 (62)	1.77(46)	1.08 (28)

TABLE 4: Summary of test functions in the CEC 2005 benchmark.

Function	Domain	Type and F^*
$F1$ (Shifted Sphere Function)	$[-100, 100]^D$	U, -450
$F2$ (Shifted Schwefel's Problem 1.2)	$[-100, 100]^D$	U, -450
$F3$ (Shifted Rotated High-Conditioned Elliptic Function)	$[-100, 100]^D$	U, -450
$F4$ (Shifted Schwefel's Problem 1.2 with Noise in Fitness)	$[-100, 100]^D$	U, -450
$F5$ (Schwefel's Problem 2.6 with Global Optimum on Bounds)	$[-100, 100]^D$	U, -310
$F6$ (Shifted Rosenbrock's Function)	$[-100, 100]^D$	M, 390
$F7$ (Shifted Rotated Griewank's Function without Bounds)	$[0, 600]^D$	M, -180
$F8$ (Shifted Rotated Ackley's Function with Global Optimum on Bounds)	$[-32, 32]^D$	M, -140
$F9$ (Shifted Rastrigin's Function)	$[-5, 5]^D$	M, -330
$F10$ (Shifted Rotated Rastrigin's Function)	$[-5, 5]^D$	M, -330
$F11$ (Shifted Rotated Weierstrass Function)	$[-0.5, 0.5]^D$	M, 90
$F12$ (Schwefel's Problem 2.13)	$[-\pi, \pi]^D$	M, -460
$F13$ (Expanded Extended Griewank's plus Rosenbrock's Function ($F8F2$))	$[-3, 1]^D$	E, -130
$F14$ (Expanded Rotated Extended Scaffe's $F6$)	$[-100, 100]^D$	E, -300
$F15$ (Hybrid Composition Function 1)	$[-5, 5]^D$	HC, 120
$F16$ (Rotated Hybrid Composition Function 1)	$[-5, 5]^D$	HC, 120
$F17$ (Rotated Hybrid Composition Function 1 with Noise in Fitness)	$[-5, 5]^D$	HC, 120
$F18$ (Rotated Hybrid Composition Function 2)	$[-5, 5]^D$	HC, 10
$F19$ (Rotated Hybrid Composition Function 2 with a Narrow Basin for the Global Optimum)	$[-5, 5]^D$	HC, 10
$F20$ (Rotated Hybrid Composition Function 2 with the Global Optimum on the Bounds)	$[-5, 5]^D$	HC, 10
$F21$ (Rotated Hybrid Composition Function 3)	$[-5, 5]^D$	HC, 360
$F22$ (Rotated Hybrid Composition Function 3 with High Condition Number Matrix)	$[-5, 5]^D$	HC, 360
$F23$ (Noncontinuous Rotated Hybrid Composition Function 3)	$[-5, 5]^D$	HC, 360
$F24$ (Rotated Hybrid Composition Function 4)	$[-5, 5]^D$	HC, 260
$F25$ (Rotated Hybrid Composition Function 4 without Bounds)	$[2, 5]^D$	HC, 260

U means unimodal function, M means multimodal function, E means extended function, and HC means hybrid composition function.

the problems in the CEC 2005 benchmark [54] are summarized in Table 4. The results of each method are derived for the same parameters used by its main corresponding method. But, some of the shared parameters are fixed in all methods including PS and NFE , respectively, equal to 40 and $10,000 \times D$, where D is always set to 50. We set the target objective cost function always as $CF_i(x) = F_i(x) - F_i^*$ where i is an integer in $\{1, 2, \dots, 25\}$ and F_i^* is the optimal target value of the i th objective function. We use mean and std. dev as two criteria in presenting the results which are, respectively, defined as the average of the best costs of the particles and the standard deviation of the best costs of the particles in different runs on all of the 25 functions shown in Table 4. Table 5 shows the results for assessment of other methods on all of the 25 functions shown in Table 4 in detail; also, it validates the results using the Friedman test with a P value of $7.92E-05$. The results presented in Table 5 have been summarized in Table 6.

According to Table 5, it is easy to conclude that the proposed algorithm has a better quality in almost all functions compared with other methods; it is always in top-3 methods. Studying Table 5 also shows that the proposed method has the best performance in 16 functions.

The proposed approach has the best performance due to reaching the zero error global optimum with the CS algorithm in $F1$ and also with DSA in $F9$. There are only 9

functions in which the proposed method does not have the best performance. We can also conclude that the proposed method has a desirable ability to result a satisfactory diversity in the problem space. It is shown by the statistical test that results of the proposed method are with the best values among different methods.

Finally, the proposed algorithm complexity according to the specified guidelines of CEC 2005 [54] is shown in Table 7.

All the mentioned methods are tested on different numbers of fitness evaluations, and the results are shown in Figure 1, where D , PS , and NFE are, respectively, equal to 30, 40, and $10,000 \times D$. We can see from Figure 1 that regardless of the changes in NFE , the proposed algorithm converges to a better solution, and also it needs less NFE rather than other methods to converge to a solution with the same quality in most of the fitness functions. This test also shows that the proposed algorithm is one of the best methods on all different numbers of fitness assessments and meets the best cost among them. In addition, when the fitness assessment number is increased, better results are derived.

In Figure 2, the results of the proposed method in terms of some criteria for different D in the set of $\{10, 20, 30, 40, 50\}$ are presented for the first six objective functions in the CEC 2005 benchmark. The considered population size in this experiment is 20, for $10,000 \times D$

TABLE 5: Assessment of different methods in numerical optimization problems of the CEC 2005 benchmark.

Function	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21	F22	F23	F24	F25	F26	F27	F28	F29	F30
Mean	1.12E+01	7.10E+01	7.42E+01	7.14E+01	6.91E+01	7.12E+01	7.12E+01	7.12E+01	7.12E+01	7.12E+01	7.12E+01	7.12E+01	7.12E+01	7.12E+01	7.12E+01	7.12E+01	7.12E+01	7.12E+01	7.12E+01	7.12E+01	7.12E+01	7.12E+01	7.12E+01	7.12E+01	7.12E+01	7.12E+01	7.12E+01	7.12E+01	7.12E+01	7.12E+01
Std	4.10E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00
Q1	7.10E+01	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00
Q3	7.10E+01	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00
Median	7.10E+01	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00	2.91E+00
Rank	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The value V_1 & ($V_2 \pm V_3$) below column F_{V_4} means that the average of the best values found by the method $\mathcal{M}$ on 30 independent runs over function F_{V_4} is V_2 and the standard deviation is V_3 . It also means that the rank of the method $\mathcal{M}$ among all methods over function F_{V_4} is V_1 . The sign & indicates that the proposed method performance is superior/inferior/equal to that of the method $\mathcal{M}$, if its value is $+/-/\times$.

TABLE 6: Summary of the results presented in Table 5 (average rank of each method on all functions of the CEC 2005 benchmark).

DSA	BMO	TLBO	CS	ROA	MBADE	APSOA
5.16 (129)	5.04 (126)	4.68 (117)	4.56 (114)	4.12 (103)	2.84 (71)	1.48 (37)

TABLE 7: Computational complexity analysis on the CEC 2005 benchmark.

D	T_0	T_1	T_2	$(T_2 - T_1)/T_0$
10	0.118	38.36	38.56	1.69
30	0.118	47.71	47.99	2.37
50	0.118	54.36	54.82	3.90

fitness function assessments. Figure 2 includes the best and the mean costs of the population members and also their standard deviation and the execution time when applying the APSOA method on test benchmark functions $F1$ – $F6$ for different dimension sizes. Using the presented results in Figure 2, we can express that increasing dimension sizes makes the problem more complicated which, in turn, causes the best cost value to be raised.

4.3. Experimental Results: CEC 2010 and Real-World Problems. In this part, we are going to compare our proposed method with other recently proposed metaheuristic methods. Here, we use the benchmark functions of the test series CEC 2010 [10] and set PS and NFE to 40 and $10,000 \times D$, respectively. 20 famous objective functions of the CEC 2010 [10] benchmark are used as $F1$ – $F20$ for assessment of this section ($F1$ – $F3$ are separable, $F4$ – $F8$ are 1-group N-nonseparable, $F9$ – $F13$ are $(D/2 \times N)$ -groups N-nonseparable, $F14$ – $F18$ are (D/N) -groups N-nonseparable, and finally $F19$ – $F20$ are nonseparable). D is 1,000 throughout this section; besides, N , i.e., the number of variables in each nonseparable subcomponent, is 50 here. Also, a set of 4 real-world problems is used as $F21$ – $F24$ for assessment of this section. The first two real-world problems, i.e., $F21$ and $F22$, are, respectively, problem number 1 and problem number 7 in CEC 2011 [6]. $F23$ and $F24$ are the problem of the linear equation system [7] and the problem of the polynomial fitting system [8]. It is worthy to be mentioned that 51 independent runs are performed, and the results are averaged over them. The comparison is taken with the following methods: SDEOA [36], JOOA [37], diversity neighborhood search-enhanced particle swarm optimization (DNSPSO [9]), and D-PSO-C [35].

According to the results presented in Table 8, except for function $F13$ in which APSOA is not in top-3, APSOA is always among the top-3. It is the best at the end of the specified number of fitness function evaluations, for half of our problems. The summary of the results presented in Table 8 is shown in Table 9. The proposed APSOA exhibits the best performance among all methods

according to the results presented in Table 9. The results in Table 8 are validated by the Friedman test with a P value of $1.92E - 02$.

4.4. Experimental Results: A Real-World Problem. Artificial intelligence is an appropriate candidate in solving many of the real-world electrical problems [55–71]. In this part, we have solved a problem of combined heat and power economic dispatch [6, 71] using our proposed method and different optimization algorithms. A particle is shown as a vector with size 9 denoted by $P = [\text{pow}_1, \text{pow}_2, \text{pow}_3, \text{pow}_4, \text{pöw}_1, \text{pöw}_2, \text{hëat}_1, \text{hëat}_2, \text{heat}_1]$. Then, the cost function of the problem is

$$\begin{aligned} \text{cost}(P) = & \sum_{i=1}^4 \text{PowerFunction}_i(\text{pow}_i) \\ & + \sum_{i=1}^2 \text{CombinedFunction}_i(\text{pöw}_i, \text{hëat}_i) \\ & + \text{HeatFunction}_1(\text{heat}_1), \end{aligned} \quad (1)$$

subject to

$$\begin{aligned} \sum_{i=1}^4 \text{pow}_i + \sum_{i=1}^2 \text{pöw}_i &= P_D + P_L, \\ \sum_{i=1}^1 \text{heat}_i + \sum_{i=1}^2 \text{hëat}_i &= H_D, \\ 10^{-7} \times \begin{bmatrix} \text{pow}_1 \\ \text{pow}_2 \\ \text{pow}_3 \\ \text{pow}_4 \\ \text{pöw}_1 \\ \text{pöw}_2 \end{bmatrix} \begin{bmatrix} 49 & 14 & 15 & 15 & 20 & 25 \\ 14 & 45 & 16 & 20 & 18 & 19 \\ 15 & 16 & 39 & 10 & 12 & 15 \\ 15 & 20 & 10 & 40 & 14 & 11 \\ 20 & 18 & 12 & 14 & 35 & 17 \\ 25 & 19 & 15 & 11 & 17 & 39 \end{bmatrix} \begin{bmatrix} \text{pow}_1 \\ \text{pow}_2 \\ \text{pow}_3 \\ \text{pow}_4 \\ \text{pöw}_1 \\ \text{pöw}_2 \end{bmatrix} &= P_L, \end{aligned}$$

FIGURE 1: Continued.

FIGURE 1: Convergence of different optimization algorithms in terms of the number of fitness evaluations on (a) $F1$, (b) $F2$, (c) $F3$, (d) $F4$, (e) $F5$, (f) $F6$, (g) $F7$, (h) $F8$, (i) $F9$, (j) $F10$, (k) $F11$, and (l) $F12$ benchmark functions in the CEC 2005 benchmark.

$$10 \leq \text{pow}_1 \leq 75,$$

$$20 \leq \text{pow}_2 \leq 125,$$

$$30 \leq \text{pow}_3 \leq 175,$$

$$40 \leq \text{pow}_4 \leq 250,$$

$$10 \leq \text{heat}_1 \leq 2695.2,$$

$$\text{CombinedConstrain } t_1(\text{heat}_1, \text{pow}_1),$$

$$\text{CombinedConstrain } t_2(\text{heat}_2, \text{pow}_2),$$

(2)

where the functions $\text{PowerConstrain } t_1(x)$, $\text{PowerConstrain } t_2(x)$, $\text{PowerConstrain } t_3(x)$, $\text{PowerConstrain } t_4(x)$, $\text{CombinedConstrain } t_3(x, y)$, $\text{CombinedConstrain } t_4(x, y)$, and $\text{HeatConstrain } t_1(x)$ are defined as follows:

$$\text{PowerFunction}_1(x) = 25 + 2 \times x + 0.008 \times x^2 + 100 \times |\sin(0.042 \times (10 - x))|,$$

$$\text{PowerFunction}_2(x) = 60 + 1.8 \times x + 0.003 \times x^2 + 140 \times |\sin(0.04 \times (20 - x))|,$$

$$\text{PowerFunction}_3(x) = 100 + 2.1 \times x + 0.0012 \times x^2 + 160 \times |\sin(0.038 \times (30 - x))|,$$

$$\text{PowerFunction}_4(x) = 120 + 2 \times x + 0.001 \times x^2 + 180 \times |\sin(0.037 \times (40 - x))|,$$

(3)

$$\text{CombinedFunction}_3(x, y) = 2650 + 14.5 \times x + 0.0345 \times x^2 + 4.2 \times y + 0.03 \times y^2 + 0.031 \times x \times y,$$

$$\text{CombinedFunction}_4(x, y) = 1250 + 36 \times x + 0.0435 \times x^2 + 0.6 \times y + 0.027 \times y^2 + 0.11 \times x \times y,$$

$$\text{HeatFunction}_1(x) = 950 + 2.0109 \times x + 0.038 \times x^2.$$

FIGURE 2: The best costs of population individuals and their mean and standard deviation and the proposed method execution time on F1–F6 functions of the test benchmark versus different dimension sizes.

Also, two other conditions $\text{CombinedFunction}_1(x, y)$ and $\text{CombinedFunction}_2(x, y)$, for two cogeneration power-heat units depicted in Figure 3 should be met. In Figure 3, the valid work regions for cogeneration power-heat unit numbers 1 and 2 are depicted.

According to the parameters defined in the papers related to different optimization algorithms, their population size has been set here. Let P_A^B and $F(P_A^B)$ represent the best solution in the population of a method A after B fitness evaluations and its cost function value, respectively. We have used P_A^{10000} and the

The value V_1 & $(V_2 \pm V_3)$ below column F_{V_4} means that the average of the best values found by the method $\mathcal{M}$ on 30 independent runs over function F_{V_4} is V_2 and the standard deviation is V_3 . It also means that the rank of the method $\mathcal{M}$ among all methods over function F_{V_4} is V_1 . The sign & indicates that the proposed method performance is superior/inferior/equal to that of the method $\mathcal{M}$, if its value is $+/-/\times$.

[illegible]

TABLE 9: Summary of the results presented in Table 8 (average rank of each method on all the functions of numerical optimization problems of the CEC 2010 benchmark and some real-world problems).

DNSPSO	D-PSO-C	SDEOA	JOOA	APSOA
4.25 (102)	4.33 (104)	2.38 (57)	2.29 (55)	1.75 (42)

FIGURE 3: Two conditions for two cogeneration power-heat units.

TABLE 10: Solutions of different optimization algorithms to the combined heat and power economic dispatch problem.

	ABC	EP	PSO	RCGA	Proposed
p_1	43.95	61.36	18.46	74.68	44.56
p_2	98.59	95.12	124.26	97.96	98.54
p_3	112.93	99.94	112.78	167.23	112.67
p_4	209.77	208.73	209.82	124.91	209.82
ph_1	98.80	98.80	98.81	98.80	95.15
ph_2	44.00	44.00	44.01	44.00	40.00
hp_1	12.10	18.07	57.92	58.10	21.50
hp_2	78.02	77.55	32.76	32.41	75.00
h_1	59.88	54.37	59.32	59.49	53.50
Cost	10317	10390	10613	10667	10097
Consumed time	5.16	5.28	5.38	6.47	1.84

related cost value in method A for the results to be fair. For this problem, the provided solutions using different optimization algorithms are shown in Table 10 in which we can see that the proposed method has the best performance.

5. Conclusions and Future Works

Nature-inspired social and solitary behaviors have motivated numerous algorithms in different scientific studies which are usually successful and efficient. In this paper, instinctive behaviors of birds are used to provide a more accurate, more target-oriented, and more controlled algorithm than the basic swarm algorithms. According to the classical conditioning learning behavior, a model is presented in which a normal task based on a natural stimulant for each particle is implemented in the search space. This model implies that when a particle experiences a low diversity category, it will move towards the local optimal point, while if it lies in a high diversity category, it will be moved towards the global optimum of its category.

An initial population according to the elite particles is also generated based on the assumption that those birds which have no sufficient energy will encounter flight problems. Another goal in the proposed algorithm is to provide more exploitation time for the particles in valuable spaces which motivated us to reduce their velocity by creating some changes in the velocity equation and vice-versa. Simulation results of the proposed method provided in four parts proved that it is an efficient and reliable algorithm in static functions compared with other algorithms. We concluded that our method uses a mechanism that finds more accurate solutions in a simpler and faster way, and it also has a better operation in industrial applications in comparison with prior methods.

For future works, we propose to use the following ideas based on what this paper was accomplished: applying the idea of chaos theory in the initial population, studying the quantum particles in the abovementioned algorithm, and implementing the algorithm of this paper to solve dynamic optimization problems.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] M. Song and D. Chen, "An improved knowledge-informed NSGA-II for multi-objective land allocation (MOLA)," *Geospatial Information Science*, vol. 21, no. 4, pp. 273–287, 2018.
- [2] T. Pukkala, "Optimized cellular automaton for stand delineation," *Journal of Forestry Research*, vol. 30, no. 1, pp. 107–119, 2019.
- [3] A. Kalantari, A. Kamsin, S. Shamshirband, A. Gani, H. Alinejad-Rokny, and A. T. Chronopoulos, "Computational intelligence approaches for classification of medical data: state-of-the-art, future challenges and research directions," *Neurocomputing*, vol. 276, pp. 2–22, 2018.
- [4] R. Andonie, A. T. Chronopoulos, D. Grosu, and H. Gálmeanu, "An efficient concurrent implementation of a neural network algorithm," *Concurrency and Computation: Practice and Experience*, vol. 18, no. 12, pp. 1559–1573, 2006.
- [5] G. E. Phillips-Wren, L. S. Iyer, U. R. Kulkarni, and T. Ariyachandra, "Business analytics in the context of big data: a roadmap for research," *Communications of the Association for Information Systems*, vol. 37, p. 23, 2015.
- [6] S. Das and P. N. Suganthan, *Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems*, Jadavpur University, Kolkata, India, 2010.
- [7] C. García-Martínez, M. Lozano, F. Herrera, D. Molina, and A. M. Sánchez, "Global and local real-coded genetic algorithms based on parent-centric crossover operators," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1088–1113, 2008.
- [8] F. Herrera and M. Lozano, "Gradual distributed real-coded genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 1, pp. 43–63, 2000.
- [9] K. Tang, X. Li, P. N. Suganthan, Z. Yang, and T. Weise, "Benchmark functions for the CEC'2010 special session and competition on large-scale global optimization," *Nature Inspired Computer Application Lab*, University of Science and Technology of China, Hefei, China, 2009, <http://goanna.cs.rmit.edu.au/xiaodong/publications/lsgo-cec10.pdf>.
- [10] G. Yildizdan and Ö. K. Baykan, "A novel modified bat algorithm hybridizing by differential evolution algorithm," *Expert Systems with Applications*, vol. 141, p. 112949, 2020.
- [11] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms*, John Wiley & Sons, Hoboken, NJ, USA, 2nd edition, 2004.
- [12] S. B. L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.
- [13] W. Sun and Y. Yuan, *Optimization Theory and Methods: Nonlinear Programming*, Springer, Berlin, Germany, 2006.
- [14] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd edition, 2006.
- [15] J. Holland, "Genetic algorithms and the optimal allocation of trials," *SIAM Journal on Computing*, vol. 2, pp. 88–105, 1973.
- [16] S. Binitha and S. S. Sathya, "A survey of bio inspired optimization algorithms," *International Journal of Soft Computing and Engineering*, vol. 2, 2012.
- [17] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the 4th IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Australia, December 1995.
- [18] N. F. Wan and L. Nolle, "Solving a multi-dimensional knapsack problem using hybrid particle," in *Proceedings of the 23rd European Conference on Modelling and Simulation*, Lancaster, UK, October 2008.
- [19] K. B. Deep, "A socio-cognitive particle swarm optimization for multi-dimensional," in *Proceedings of the First International Conference on Emerging Trends in Engineering*, pp. 355–360, Nagpur, India, July 2008.
- [20] X. Shen, Y. Li, C. Chen, J. Yang, and D. Zhang, "Greedy continuous particle swarm optimisation algorithm for the knapsack problems," *International Journal of Computer Applications in Technology*, vol. 44, no. 2, pp. 37–144, 2012.
- [21] H. S. Lopes and L. S. Coelho, "Particle swarm optimization with fast local search for the blind traveling salesman problem," in *Proceedings of the Fifth International Conference on Hybrid Intelligent Systems*, pp. 245–250, Rio de Janeiro, Brazil, November 2005.
- [22] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.
- [23] A. Banharsakun, B. Sirinaovakul, and T. Achalakul, "Job shop scheduling with the best-so-far ABC," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 3, pp. 583–593, 2012.
- [24] D. Karaboga and B. Gorkemli, "A combinatorial artificial bee colony algorithm for traveling salesman problem," in *Proceedings of the International Symposium on Intelligent Systems and Applications*, pp. 50–53, Istanbul, Turkey, June 2011.
- [25] D. T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, and M. Zaidi, *The Bees Algorithm. Technical Note*, Cardiff University, Cardiff, UK, 2005.
- [26] D. Pham, E. Koc, J. Lee, and J. Phrueksanant, "Using the bees algorithm to schedule jobs for a machine," in *Proceedings of Eighth International Conference on Laser Metrology CMM and Machine*, Cardiff, UK, 2007.
- [27] D. T. Pham, S. Otri, A. Afify, M. Mahmuddin, and H. Al-Jabbouli, "Data clustering using the bees algorithm," in *Proceedings of the 40th CIRP International Seminar on Manufacturing Systems*, Liverpool, UK, May 2007.
- [28] Z. Geem, J. Kim, and G. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, 2001.
- [29] X. Miao, J. Chu, L. Zhang, and J. Qiao, "An evolutionary neural network approach to simple prediction of dam deformation," *Journal of Information & Computational Science*, vol. 10, pp. 1315–1324, 2013.
- [30] M.-Y. Cheng and L.-C. Lien, "Hybrid artificial intelligence-based PBA for benchmark functions and Facility Layout Design optimization," *Journal of Computing in Civil Engineering*, vol. 26, no. 5, pp. 612–624, 2012.
- [31] W. Feng and C. Liu, "A novel particle swarm optimization algorithm for global optimization," *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 9482073, 9 pages, 2016.
- [32] X. S. Yang and S. Deb, "Cuckoo search via Levy flights," in *Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pp. 210–214, Coimbatore, India, December 2009.
- [33] P. Civicioglu, "Transforming geocentric Cartesian coordinates to geodetic coordinates by using differential search

- algorithm," *Computers & Geosciences*, vol. 46, no. 9, pp. 229–247, 2012.
- [34] A. Askarzadeh, "Bird mating optimizer: an optimization algorithm inspired by bird-mating strategies," *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 4, pp. 1213–1228, 2014.
 - [35] X. Xu, Y. Tang, J. Li, C. Hua, and X. Guan, "Dynamic multi-swarm particle swarm optimizer with cooperative learning strategy," *Applied Soft Computing*, vol. 29, pp. 169–183, 2015.
 - [36] A. Draa, S. Bouzoubia, and I. Boukhalfa, "A sinusoidal differential evolution algorithm for numerical optimisation," *Applied Soft Computing*, vol. 27, pp. 99–126, 2015.
 - [37] G. Sun, R. Zhao, and Y. Lan, "Joint operations algorithm for large-scale global optimization," *Applied Soft Computing*, vol. 38, pp. 1025–1039, 2016.
 - [38] J. Wang, B. Zhou, and S. Zhou, "An improved Cuckoo search optimization algorithm for the problem of chaotic systems parameter estimation," *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 2959370, 8 pages, 2016.
 - [39] E. R. Tanweer, S. Suresh, and N. Sundararajan, "Self-regulating particle swarm optimization algorithm," *Innovative Applications of Artificial Neural Networks in Engineering*, vol. 294, pp. 182–202, 2015.
 - [40] F. T. Zhao, Z. Yao, J. Luan, and X. Son, "A novel fused optimization algorithm of genetic algorithm and Ant colony optimization," *Mathematical Problems in Engineering*, vol. 2016, Article ID 2167413, 10 pages, 2016.
 - [41] M. Thankur, "A new genetic algorithm for global optimization of multimodal continuous functions," *Journal of Computational Science*, vol. 5, no. 2, pp. 298–311, 2014.
 - [42] N. Zare, H. Shameli, and H. Parvin, "An innovative natural-derived meta-heuristic optimization method," *Applied Intelligence*, 2016.
 - [43] C. Cubukcuoglu, I. Chatzikonstantinou, M. Tasgetiren, I. Sariyildiz, Q.-K. Pan, and Q. K. Pan, "A multi-objective harmony search algorithm for sustainable design of floating settlements," *Algorithms*, vol. 9, no. 3, p. 51, 2016.
 - [44] I. Obagbuwa and A. Abidoye, "Binary Cockroach swarm optimization for combinatorial optimization problem," *Algorithms*, vol. 9, no. 3, p. 59, 2016.
 - [45] R. M. Rizk Allah, "Hybridization of Fruit fly optimization algorithm and Firefly algorithm for solving Nonlinear Programming problems," *International Journal of Swarm Intelligence and Evolutionary Computation*, vol. 5, no. 2, 2016.
 - [46] Q. Zhang, A. Zhou, S. Zhao, P. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization test instances for the CEC 2009 special session and competition," Technical Report CES-487, American Society of Mechanical Engineers, New York, NY, USA, 2009.
 - [47] R. M. May, "Simple mathematical models with very complicated dynamics," *Nature*, vol. 261, no. 5560, pp. 459–467, 1976.
 - [48] R. Storn and K. Price, "Differential evolution a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
 - [49] H. Gao and W. Xu, "A new particle swarm algorithm and its globally convergent modifications," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 5, pp. 1334–1351, 2011.
 - [50] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, and M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Applied Soft Computing*, vol. 11, no. 2, pp. 1679–1696, 2011.
 - [51] Y. Liang, Y. Liu, and L. Zhang, "An improved Artificial Bee Colony (ABC) algorithm for large scale optimization," in *Proceedings of the 2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA)*, Toronto, Canada, December 2013.
 - [52] X. S. Yang, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, Washington, DC, USA, Second edition, 2011.
 - [53] S. C. Satapathy and A. Naik, "Modified Teaching-Learning-Based Optimization algorithm for global numerical optimization-a comparative study," *Swarm and Evolutionary Computation*, vol. 16, pp. 28–37, 2014.
 - [54] P. N. Suganthan, N. Hansen, and J. J. Liang, *Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real Parameter Optimization*, Nanyang Technological University, Nanyang, Singapore, 2005.
 - [55] M. Gitizadeh, S. Goodarzi, and A. R. Abbasi, "An efficient linear network model for transmission expansion planning based on piecewise McCormick Relaxation," *IET Generation Transmission & Distribution*, vol. 13, no. 23, pp. 5404–5412, 2019.
 - [56] A. R. Abbasi, M. R. Mahmoudi, and Z. Avazzadeh, "Diagnosis and clustering of power transformer winding fault types by cross-correlation and clustering analysis of FRA results, IET Generation," *Transmission & Distribution*, vol. 12, no. 19, pp. 4301–4309, 2018.
 - [57] A. Abbasi and A. Seifi, "A novel method mixed power flow in transmission and distribution systems by using master-slave splitting method," *Electric Power Components and Systems*, vol. 36, no. 11, pp. 1141–1149, 2008.
 - [58] A. R. Abbasi and A. R. Seifi, "Considering cost and reliability in electrical and thermal distribution networks reinforcement planning," *Energy*, vol. 84, pp. 25–35, 2015.
 - [59] A. R. Abbasi and A. R. Seifi, "A new coordinated approach to state estimation in integrated power systems," *International Journal of Electrical Power & Energy Systems*, vol. 45, no. 1, pp. 152–158, 2013.
 - [60] A. R. Abbasi and A. R. Seifi, "Energy expansion planning by considering electrical and thermal expansion simultaneously," *Energy Conversion and Management*, vol. 83, pp. 9–18, 2014.
 - [61] A. R. Abbasi and A. R. Seifi, "Unified electrical and thermal energy expansion planning with considering network reconfiguration, IET Generation," *Transmission & Distribution*, vol. 9, no. 6, pp. 592–601, 2015.
 - [62] A. Zare, A. Kavousi-Fard, A. Abbasi, and F. Kavousi-Fard, "A sufficient stochastic framework to capture the uncertainty of load models in the management of distributed generations in power systems," *Journal of Intelligent & Fuzzy Systems*, vol. 28, no. 1, pp. 447–456, 2015.
 - [63] A. R. Abbasi and A. R. Seifi, "Simultaneous Integrated stochastic electrical and thermal energy expansion planning," *IET Generation, Transmission & Distribution*, vol. 8, no. 6, pp. 1017–1027, 2014.
 - [64] A. R. Abbasi and A. R. Seifi, "Fast and perfect damping circuit for ferroresonance phenomena in coupling capacitor voltage transformers," *Electric Power Components and Systems*, vol. 37, no. 4, pp. 393–402, 2009.
 - [65] A. Kavousi-Fard, S. Abbasi, A. Abbasi, and S. Tabatabaie, "Optimal probabilistic reconfiguration of smart distribution grids considering penetration of plug-in hybrid electric vehicles," *Journal of Intelligent & Fuzzy Systems*, vol. 29, no. 5, pp. 1847–1855, 2015.
 - [66] M. Javidsharifi, T. Niknam, J. Aghaei, G. Mokryani, and P. Papadopoulos, "Multi-objective day-ahead scheduling of

- microgrids using modified grey wolf optimizer algorithm,” *Journal of Intelligent & Fuzzy Systems*, vol. 36, no. 3, pp. 2857–2870, 2019.
- [67] M. Mohammadi, S. Soleymani, T. Niknam, and T. Amraee, “Stochastic multi-objective distribution automation strategies from reliability enhancement point of view in the presence of plug in electric vehicles,” *Journal of Intelligent & Fuzzy Systems*, vol. 36, no. 3, pp. 2933–2945, 2019.
- [68] M. Saeidi, T. Niknam, J. Aghaei, and M. Zare, “Multi-objective coordination of local and centralized volt/var control with optimal switch and distributed generations placement,” *Journal of Intelligent & Fuzzy Systems*, vol. 36, no. 6, pp. 6605–6617, 2019.
- [69] M. Ahmadi, K. Kazemi, A. Aarabi, T. Niknam, and M. S. Helfroush, “Image segmentation using multilevel thresholding based on modified bird mating optimization,” *Multimedia Tools and Applications*, vol. 78, no. 16, pp. 23003–23027, 2019.
- [70] S. Pirouzi, J. Aghaei, T. Niknam et al., “Power conditioning of distribution networks via single-phase electric vehicles equipped,” *IEEE Systems Journal*, vol. 13, no. 3, pp. 3433–3442, 2019.
- [71] M. J. Mokarram, T. Niknam, J. Aghaei, M. Shafie-khah, and J. P. S. Catalão, “Hybrid optimization algorithm to solve the nonconvex multiarea economic dispatch problem,” *IEEE Systems Journal*, vol. 13, no. 3, pp. 3400–3409, 2019.