# A New Metaheuristic Algorithm Based on Shark Smell Optimization

**OVEIS ABEDINIA,[1] NIMA AMJADY,[1] AND ALI GHASEMI[2]**

[1]*Department of Electrical Engineering, Semnan University, Semnan, Iran and* [2]*Young Researchers and Elite Club, Ardabil Branch, Islamic Azad University, Ardabil, Iran*

*Received 5 July 2014; revised 20 October 2014; accepted 8 November 2014*

*In this article, a new metaheuristic optimization algorithm is introduced. This algorithm is based on the ability of shark, as a superior hunter in the nature, for finding prey, which is taken from the smell sense of shark and its movement to the odor source. Various behaviors of shark within the search environment, that is, sea water, are mathematically modeled within the proposed optimization approach. The effectiveness of the suggested approach is compared with many other heuristic optimization methods based on standard benchmark functions. Also, to illustrate the efficiency of the proposed optimization method for solving real-world engineering problems, it is applied for the solution of load frequency control problem in electrical power systems. The obtained results confirm the validity of the proposed metaheuristic optimization algorithm.*  *Complexity 21: 97–116, 2016*

**Key Words:** shark smell optimization; metaheuristic algorithm; optimization problem

## 1. INTRODUCTION

Constrained optimization is a mathematical procedure for determining optimal allocation of scarce resources subject to a set of constraints. Optimization is maximizing or minimizing some objective functions relative to some sets, often representing a range of choices available in a certain situation. The objective function allows comparison of the different choices for determining which might be "best" [1]. The solution space of the problem is formed considering the decision variables and constraints wherein the optimum point of the objective function should be found [2]. Many of real-world optimization problems involve with complexities such as nonlinearity, nonconvexity, nonsmoothness, nondifferentiability, mixed integer nature, and discontinuous domain, which challenge the numerical optimization methods [3]. Accordingly, to tackle the mentioned complexities, several metaheuristic optimization techniques have been proposed in the literature in the recent decades such as genetic algorithm (GA) [4], particle swarm optimization (PSO) [5,6], ant colony optimization (ACO) [7,8], honey bee mating optimization (HBMO) [9,10], artificial bee colony (ABC) [11,12], bacterial foraging (BF) [13], clonal selection algorithm (CSA) [14], invasive weed optimization (IWO) [15], shuffled frog leaping (SFL) [16], evolutionary algorithm (EA) [17], differential evolution (DE) [18],

*Correspondence to: O. Abedinia, E-mail: abediniaoveis@gmail.com*

simulated annealing (SA) [19], and gravitational search algorithm (GSA) [20,21]. Due to their high flexibility, simplicity and modeling efficiency, these optimization methods have been widely used in many scientific and engineering areas. These techniques are usually based on the evolution of a candidate solution or a population of candidate solutions to search the solution space for finding optimum solutions. For the search process, different evolutionary operators such as random generation, crossover, mutation, and selection have been proposed in the literature. A bibliography of heuristics optimization methods can be found in [1,22].

Most of heuristic optimization methods are inspired by natural phenomena such as the genetic process in natural evolution used in GA, the colonization of invasive weeds used in IWO, swarming movement of particles used in PSO, HBMO, and ABC, foraging behavior of ant species and *Escherichia coli* bacteria used in ACO and BF, respectively, frog leaping used in SFL, selection mechanism of biological immunity system used in CSA, annealing process of metals used in SA and gravitational forces of masses used in GSA.

At first glance, all animals have abilities, which help them live in the nature [23]. However, some animals have special capabilities, which make them superior ones. In a natural hunting process, finding the prey and movement of the hunter toward the prey are important factors. The animal that finds the prey in a short time with correct movements in the search space could be a successful hunter. A well-known superior hunter in the nature is shark [24]. Superiority of this hunter is greatly related to its ability for finding prey within large search spaces in a short time based on its strong smell sense. This article focuses on shark's hunting ability, which is one of the incredible phenomena in the nature.

In this article, a new metaheuristic optimization method inspired from shark hunting ability based on its smell sense is proposed, called hereafter shark smell optimization (SSO). The remaining parts of the article are organized as follows. In section 2, the underlying idea of the proposed metaheuristic algorithm is presented. The mathematical formulation of SSO is detailed in section 3. In section 4, the proposed SSO is applied to several standard benchmark functions and one real-world engineering problem. The results obtained from SSO are compared with the results of many other metaheuristic optimization methods. These comparisons reveal the effectiveness of the proposed SSO. Finally, section 5 concludes the article.

## 2. UNDERLYING IDEA OF SSO

Olfactory system in any animal is the primary sensory system that responds to chemical stimuli emanating from a remote source. In fishes, the smell receptors are positioned in the olfactory pits, one on each side of the head. Each pit has two outside openings through which water flows in and out. The water motion through the pit is maintained by the waving of tiny hairs on the cells lining the pit and also by the force of the fish's moving forward through the water. Dissolved chemicals make contact with a pleated surface rich in olfactory nerve endings [25]. However, unlike other sensory nerves in vertebrates, the olfactory receptors conduct their impulses directly to the brain without any nerve relays in-between, as is the case for the other senses such as vision and hearing. The portion of the brain that receives the smell impulses is called the olfactory bulb located at the very front of the brain. Fishes have two olfactory bulbs, one for each olfactory pit. Larger surface devoted to smell nerves in the olfactory pits and larger "smells information" centers in the brain make stronger the smell sense of the fish [26]. Eels and sharks have "oversized" olfactory bulbs for smell information processing. Sharks have been swimming the world's oceans for more than 400 million years [27] and remained essentially unchanged since about 350 million years ago. One of the reasons of the survival of sharks is their high ability for localizing the prey based on their strong smell sense.

The shark's nose is one of its most effective senses. As a shark swims, water flows through two nostrils located along the sides of the snout. The water enters the olfactory pits and moves past folds of skin covered with the sensory cells. Some sharks can detect even the slightest traces of blood in the water with these sensory cells [27]. For example, a shark could detect one drop of blood in an olympic-sized pool. Because of this, sharks can smell an injured animal up to 1-km away [28]. A shark's sense of smell is also directional. The nasal cavities act like ears; smell coming from the left of the shark will arrive at the left cavity just before it arrives at the right cavity. This allows a shark to find out where a smell is coming from and proceed in that direction [29,30]. In Figure 1, the movement of shark to the odor source based on its concentration is schematically presented. In this movement, the concentration is an important factor to guide the shark to the prey. In other words, if the concentration is more and stronger, the movement of shark is true. This characteristic is used in the proposed algorithm to find the solution of an optimization problem.

## 3. FORMULATION OF THE PROPOSED SSO ALGORITHM

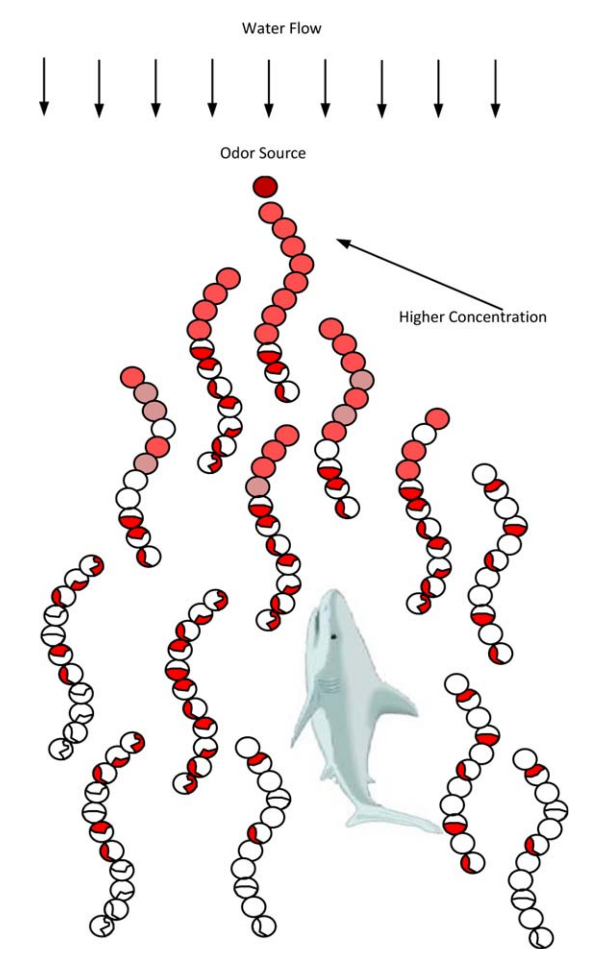
To construct the mathematical model of the shark's search process, some assumptions are considered as follows:

- The fish is injured and injects blood to the sea (search environment). So, the velocity of the fish



DOI 10.1002/cplx

**FIGURE 1**




Schematic illustration of a shark movement to odor source.

movement is low and neglected against the shark's velocity. Hence, the source (prey) is approximately assumed to be fixed.

- The blood is regularly injected to the sea and the effect of the water flows on distorting the odor particles is neglected. Thus, closer odor particles to the fish will be stronger. Consequently, by following the odor particles, the shark can approach the prey.
- There is one blood source, that is, one injured fish, in the search environment of the shark.

The steps of the proposed SSO algorithm, based on the shark's search process, are described in the following sections.

### 3.1. Finding Initial Odor Particles: Initialization of SSO Algorithm

The search process begins when the shark smells an odor particle, which is usually a weak diffusion from injured fish, that is, prey. To model this, a population of the initial solutions for the optimization problem is randomly generated within the feasible search domain, which each of them represents one odor particle, that is, one possible position for the shark, at the beginning of the search process:

$$[X_1^1, X_2^1, \ldots, X_{NP}^1], \quad NP{=}\text{Population Size} \tag{1}$$

Where the $i$th initial position vector $X_i^1$, that is, $i$th initial candidate solution for the optimization problem, is as follows:

$$X_i^1 = [x_{i,1}^1, x_{i,2}^1, \ldots, x_{i,ND}^1], \quad i=1,\ldots NP \tag{2}$$

Where $x_{i,j}^1$ represents $j$th dimension of the $i$th shark position or equivalently $j$th decision variable of the $i$th individual $X_i^1$; ND indicates number of decision variables of the optimization problem. Magnitude of odor in each position indicates its closeness to the prey. This is modeled through objective function in the SSO algorithm. Assuming a maximization problem, without loss of generality, a higher value of the objective function, similar to a stronger odor, indicates a closer position to the prey for the shark or a more optimal candidate solution for the optimization problem. Up to this point, the SSO algorithm has been initialized.

### 3.2. Movement of Shark Toward Prey: Evolution of the SSO Algorithm

Shark in each position moves with a velocity toward stronger odor particles to become closer to the prey. Thus, corresponding to the position vectors, we have NP initial velocity vectors as follows:

$$[V_1^1, V_2^1, \ldots, V_{NP}^1] \tag{3}$$

where each velocity vector has components in each dimension:

$$V_i^1 = [v_{i,1}^1, v_{i,2}^1, \ldots, v_{i,ND}^1], \quad i=1,\ldots,NP \tag{4}$$

Shark follows the odor and direction of shark movement is adjusted based on the odor intensity. Also, by increasing the concentration of the odors, the velocity of shark increases. From optimization viewpoint, we can mathematically model this type of movement through gradient of the objective function (that should be maximized), illustrating the direction that the objective function increases by the highest rate:

$$V_i^k = \eta_k . R1 . \nabla(\text{OF})|_{X_i^k}, \quad i=1,\ldots,NP, \quad k=1,\ldots,k_{\max} \tag{5}$$

where OF is the objective function and $\nabla$(OF) indicates its gradient. In the SSO model, the forward movement of shark is divided into a number of stages, denoted by $k_{\max}$,


DOI 10.1002/cplx

such that the velocity of shark in each stage, that is, $V_i^k$, is approximately constant. Superscript $k$ indicates stage number. Also, $\eta_k \in [0,1]$ is considered in (5), since the shark may not be able to reach the velocity indicated by the gradient function in each stage. $R1$ is a random number with uniform distribution in the interval [0,1]. It is considered in (5) to give more stochastic search nature to the algorithm. The idea of incorporating a random number uniformly distributed in the interval [0,1] into a velocity update process has been taken from GSA [20,21]. Based on (5), the velocity in each dimension can be written as:

$$V_{i,j}^k = \eta_k . R1 . \frac{\partial(\mathrm{OF})}{\partial x_j}\bigg|_{x_{i,j}^k} \qquad j=1,\ldots,\mathrm{ND},\ i=1,\ldots,\mathrm{NP},\ k=1,\ldots,k_{\max} \tag{6}$$

Since shark has inertia, its acceleration is limited. Thus, its current velocity depends on the previous velocity. This is modeled by modifying (6) as follows:

$$v_{i,j}^k = \eta_k . R1 . \frac{\partial(OF)}{\partial x_j}\bigg|_{x_{i,j}^k} + \alpha_k . R2 . v_{i,j}^{k-1} \qquad j=1,\ldots,\mathrm{ND},\ i=1,\ldots\mathrm{NP},\ k=1,\ldots,k_{\max} \tag{7}$$

where $\alpha_k$ belonging to [0,1] is a constant for stage $k$, called inertia coefficient or momentum rate. Larger values of this coefficient mean higher inertia and so more dependency of its current velocity on the previous velocity. From mathematical viewpoint, using the momentum term leads to smoother search paths in the solution space [31]. Another random number generator with uniform distribution in the interval [0,1], that is, $R2$, is considered for the momentum term, in addition to $R1$ of the gradient term, to further enhance the search diversity of the algorithm. For the velocity of the first stage, that is, $v_{i,j}^1$, the initial velocity of shark before starting the search process, that is, $v_{i,j}^0$, may be neglected or randomly set at a small value. Moreover, the velocity of shark can be increased up to a limit. Unlike most fishes, sharks do not have swim bladders to help them stay afloat. They must constantly swim in a slightly upward direction, using strong tail fin for propulsion, to keep from sinking [29]. The normal velocity of shark is about 20 km/h and reaches 80 km/h in attack. Hence, the ratio of high velocity/low velocity for shark is a limited value, for example, 80/20 = 4. This velocity limiter is used for each stage of the SSO algorithm as follows:

$$|v_{i,j}^k| = \min\left[\left|\eta_k . R1 . \frac{\partial(\mathrm{OF})}{\partial x_j}\bigg|_{x_{i,j}^k} + \alpha_k . R2 . v_{i,j}^{k-1}\right|, \left|\beta_k . v_{i,j}^{k-1}\right|\right] \qquad j=1,\ldots,\mathrm{ND},\ i=1,\ldots,\mathrm{NP},\ k=1,\ldots,k_{\max} \tag{8}$$

where $\beta_k$ is the velocity limiter ratio for stage $k$. The magnitude of $v_{i,j}^k$ is obtained from (8) and its sign is the sign of the selected term by the "min" operator of (8). The new position of shark due to forward movement, denoted by $Y_i^{k+1}$, is determined based on its previous position and velocity:

$$Y_i^{k+1} = X_i^k + V_i^k . \Delta t_k \qquad i=1,\ldots,NP \qquad k=1,\ldots,k_{\max} \tag{9}$$

where $\Delta t_k$ indicates time interval of the stage $k$. For simplicity, it is assumed that $\Delta t_k = 1$ for all stages. Each component $v_{i,j}^k$ ($1 \le j \le \mathrm{ND}$) of the vector $V_i^k$ is obtained from (8). In addition to forward movement, sharks usually perform rotational movements along their path to find stronger odor particles and improve the direction of their progress [29]. This kind of movement for a real shark and its typical simulation is shown in Figure 2. As seen from this figure, the rotation of shark is on a closed contour and not necessarily a circle. From optimization viewpoint, shark implements a local search in each stage to find better candidate solutions. This local search is modeled in the SSO algorithm as follows:

$$Z_i^{k+1,m} = Y_i^{k+1} + R3 . Y_i^{k+1} \qquad m=1,\ldots,M \quad i=1,\ldots,\mathrm{NP} \quad k=1,\ldots,k_{\max} \tag{10}$$

where $R3$ is a random number with uniform distribution in the range of [−1,+1]; $M$ indicates number of points in the local search of each stage. As this operator implements a local search around $Y_i^{k+1}$, the range of random number generation of $R3$ is [−1,+1]. The $M$ points of the local search, that is, $Z_i^{k+1,m}$, are in the vicinity of $Y_i^{k+1}$ (e.g., if zero is generated by the random number generator, $Y_i^{k+1}$ is obtained) and by connecting these $M$ points, a closed contour similar to the rotational movement of shark can be obtained.

If shark finds a point with stronger odor during the rotational movement, it will go to this point and continue the search path from it as shown in Figure 2. This characteristic is implemented in the SSO algorithm as follows

$$X_i^{k+1} = \arg\max\{\mathrm{OF}(Y_i^{k+1}), \mathrm{OF}(Z_i^{k+1,1}), \ldots, \mathrm{OF}(Z_i^{k+1,M})\} \qquad i=1,\ldots,\mathrm{NP} \tag{11}$$

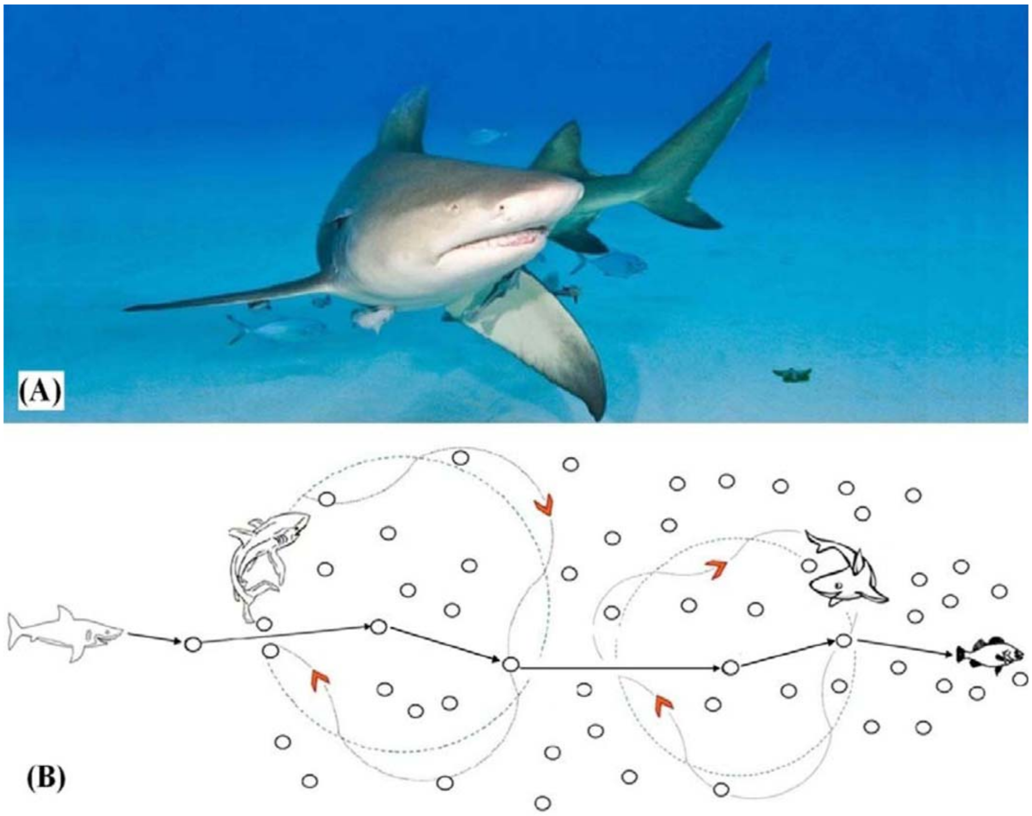
Considering that the objective function OF should be maximized. In other words, among $Y_i^{k+1}$ obtained from the forward movement and $Z_i^{k+1,m}$ ($m=1,\ldots,M$) obtained from the rotational movement, the candidate solution with the highest OF value is selected as the next position of the shark, that is, $X_i^{k+1}$. The cycle of forward and rotational movements is continued until $k$ reaches $k_{\max}$.

The performance of the SSO optimization algorithm can be summarized as shown in the flowchart of Figure 3. Like the other metaheuristic optimization methods, SSO has a number of user-defined parameters including population size NP and number of stages $k_{\max}$ as well as $\eta$, $\alpha$, and $\beta$ of each stage. In the numerical experiments of this



DOI 10.1002/cplx

FIGURE 2




The rotational movement of shark (A) real shark (B) simulation.

article, $\eta = 0.9$, $\alpha = 0.1$ (i.e., higher share for the new velocity component compared to the previous velocity), and $\beta = 4$ (the value obtained from the natural motion of shark) are considered for all stages of the SSO algorithm. We have empirically seen that SSO works well with these values of $\eta$, $\alpha$, and $\beta$. However, these parameters can be fine-tuned for each optimization problem separately. A more effective way is changing these parameters along the evolution of SSO based on an adaptation mechanism. For instance, such a mechanism may begin from large values for $\eta$ and $\beta$ and a small value for $\alpha$ and then adaptively decrease $\eta$ and $\beta$ and increase $\alpha$. In this way, the algorithm can proceed with large steps in the initial stages of the evolution process to have high exploration capability and small steps in the last stages (when the algorithm approaches the optimum solution) to benefit from high-resolution search around the optimum solution. However, further investigation of the algorithm enhancements is kept for the future research works.

After setting the parameters, the population and stage counter of the SSO are initialized. Then, the population evolves through the operators of the forward and rotational movements. Finally, the best individual in the last stage is selected as the SSO solution for the optimization problem.

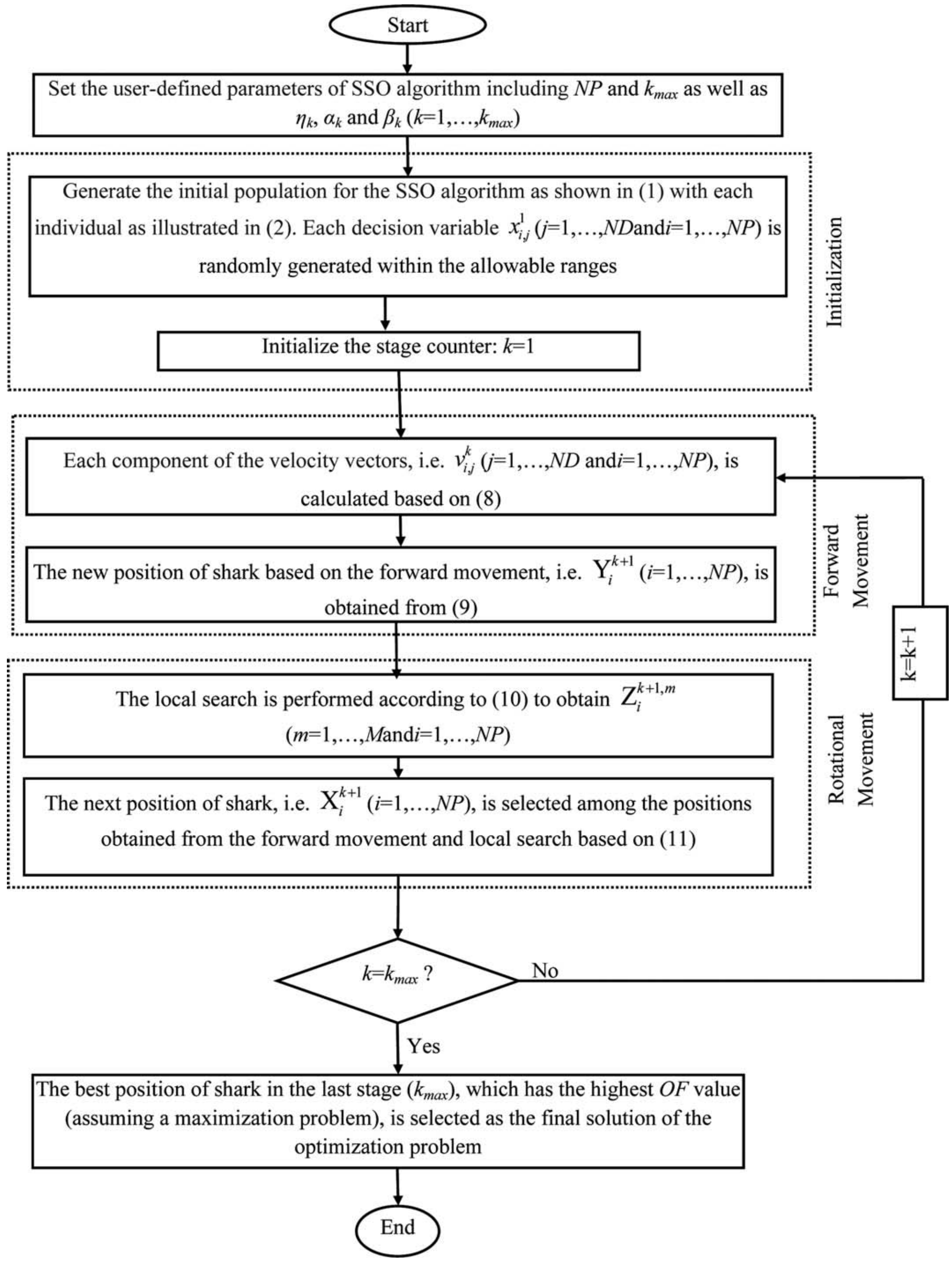
To the best of the authors' knowledge, the search operators of the proposed SSO including momentum-incorporated gradient-based forward movement and rotational movement-based local search are specific to this algorithm and have not been presented in the other metaheuristic methods.

## 4. NUMERICAL RESULTS

In this section, the proposed SSO algorithm is tested on many standard benchmark functions, usually used as test case to evaluate the performance of stochastic search methods, as well as a real-world engineering problem. These test functions challenge the SSO algorithm to test its abilities in different mathematical environments. Although SSO is a basic metaheuristic optimization method and does not have any adaptation mechanism or auxiliary operator, usually added to basic metaheuristic methods to enhance their search ability and convergence


DOI 10.1002/cplx

**FIGURE 3**



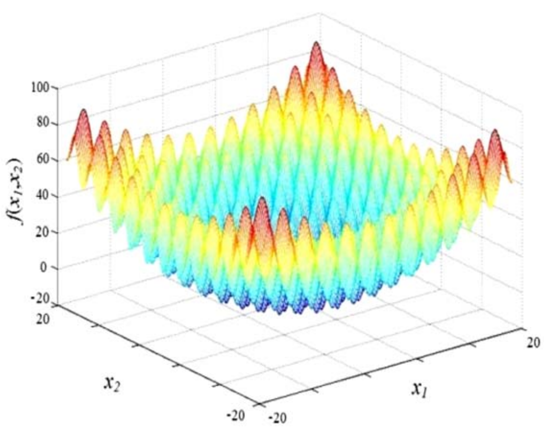

Flowchart of SSO algorithm.

behavior, such as adaptive PSO (APSO) [32], modified PSO [33], improved PSO [34], and enhanced PSO [35] obtained from basic PSO algorithm, it will be shown that SSO has high performance compared to many other metaheuristic optimization methods. The results obtained for different test cases are presented in the next subsections, respectively.



DOI 10.1002/cplx

FIGURE 4




The shape of Rastrigin function.

#### 4.1. Test Case 1: Rastrigin Benchmark Function (Reference of Data: [36])

This test case is based on cosine function, which includes several local optima, as follows.

$$
\begin{aligned}
&f(x_1,x_2)=20+\sum_{i=1}^{2}(x_i^2-10.\cos(2.\pi.x_i)) \\
&-20 \le x_i \le 20, \qquad i=1,2
\end{aligned}
\tag{12}
$$

In this benchmark, the goal is finding the minimum value. The shape of Rastrigin benchmark function is shown in Figure 4, which illustrates the large number of its local optima. The results obtained from SSO algorithm for the test case are shown in Table 1 and compared with the results obtained from 32 other metaheuristic optimization approaches including EA [17], GA [37], improved GA (IGA) [37], DE [18], PSO [6], iteration PSO (IPSO) [38], chaotic PSO (CPSO) [32], APSO [32], PSO with time varying acceleration coefficients (PSOTVAC) [6], PSO with improved inertia weight (PSOIIW) [39], hybrid GA-PSO (HGAPSO) [40], dynamic PSO (DPSO) [41], fuzzy PSO (FPSO) [42], harmony search algorithm (HSA) [43], improved HSA (IHSA) [43], ACO [7], chaotic ACO (CACO) [8], bacterial foraging algorithm (BFA) [13], GSA [20], IGSA [21], seeker optimization algorithm (SOA) [44], imperialist competitive algorithm (ICA) [45], ABC [12], improved ABC (IABC) [11], chaotic ABC (CABC) [46], parallel ABC (PABC) [36], discrete ABC (DABC) [47], rosenberg ABC (RABC) [48], modified ABC (MABC) [48], HBMO [9], improved HBMO (IHBMO) [10], and honey bee optimization (HBO) [49]. All of these 32 benchmark methods have frequently been used in engineering applications and, for this reason, have been considered here for comparison with the proposed SSO algorithm. These 32 optimization approaches have been implemented according to the procedures given in the corresponding references and tested on the Rastrigin benchmark function. Some of these 32 methods are basic metaheuristic optimization algorithms (such as EA, GA, DE, and PSO) and the other ones are improved or combined versions of these algorithms (such as IGA, IPSO, and HGAPSO). The free parameters of each method of Table 1 are fine-tuned based on 10 trial runs (the same number of trial runs is considered for all methods of Table 1 for the sake of a fair comparison). Afterward, minimum, mean, and maximum objective function values obtained among 30 other trial runs as well as computation time measured on the hardware set of 64-bit computer with 16 GB of RAM and Intel Core i7 CPU are reported for each method in Table 1.

Considering the minimum objective function value (MIN index in Table 1), the proposed SSO algorithm has better result than 22 other methods. Ten methods out of 32 alternative methods, including PSOTVAC, PSOIIW, FPSO, IHSA, BFA, IGSA, DABC, RABC, MABC, and IHBMO can reach the same MIN result of SSO, that is, zero, which is the global minimum of this test case. Thus, the MIN result of the SSO algorithm is better than or equal to the MIN result of the other methods. Regarding the mean and maximum objective function values (MEAN and MAX indices in Table 1), the proposed SSO algorithm has absolutely better results than all other 32 methods and no other method can even reach the result of SSO. These comparisons clearly illustrate higher accuracy and search ability of the proposed SSO algorithm compared to the other methods. Moreover, MIN, MEAN, and MAX results of SSO are very close to each other, indicating its high robustness in different runs. From this viewpoint, SSO is also the best method of Table 1. Finally, the proposed SSO algorithm has the lowest computation time among all methods of Table 1, which indicates its high computational efficiency. In other words, SSO obtains the best results with the lowest computation time compared to 32 other methods of Table 1. The computation times of Table 1 are measured by the computer timer.

#### 4.2. Test Case 2: Griewangk Benchmark Function (Reference of Data: [50])

This test case is a highly nonlinear and nonconvex benchmark function with absolute minimum equal to zero. The mathematical formulation of this function is presented in (13) and its shape is shown in Figure 5.

$$
\begin{aligned}
&f(x_1,x_2)=\sum_{k=1}^{2}\frac{x_k^2}{4000}-\prod_{k=1}^{2}\cos\left(\frac{x_k}{\sqrt{k}}\right)+1 \\
&-600 \le x_k \le 600 \quad k=1,2
\end{aligned}
\tag{13}
$$

The results obtained for minimizing the Griewangk test function are shown in Table 2 similar to Table 1. From

**TABLE 1**

Obtained Results for Rastrigin Benchmark Function

| Index | EA | GA | IGA | DE | PSO | IPSO | CPSO | APSO | PSOTVAC | PSOIIW | HGAPSO |
|---|---|---|---|---|---|---|---|---|---|---|---|
| MIN | 2.16 e −10 | 4.22 e −9 | 8.12 e −11 | 4.12 e −11 | 4.12 e −12 | 4.22 e −14 | 1.54 e −14 | 2.12 e −17 | 0.00 | 0.00 | 7.12 e −17 |
| MEAN | 5.12 e −3 | 3.78 e −1 | 8.53 e −2 | 5.31 e −3 | 5.23 e −3 | 5.13 e −2 | 3.30 e −2 | 5.77 e −3 | 3.52 e −5 | 2.15 e −5 | 4.12 e −4 |
| MAX | 1.17 e −2 | 1.321 | 2.43 e −1 | 6.53 e −2 | 5.67 e −2 | 2.08 e −1 | 1.28 e −1 | 2.14 e −2 | 8.19 e −5 | 8.86 e −5 | 9.13 e −4 |
| Time (s) | 3.256 | 3.001 | 2.813 | 2.382 | 2.114 | 2.110 | 2.109 | 2.108 | 2.102 | 2.105 | 2.095 |

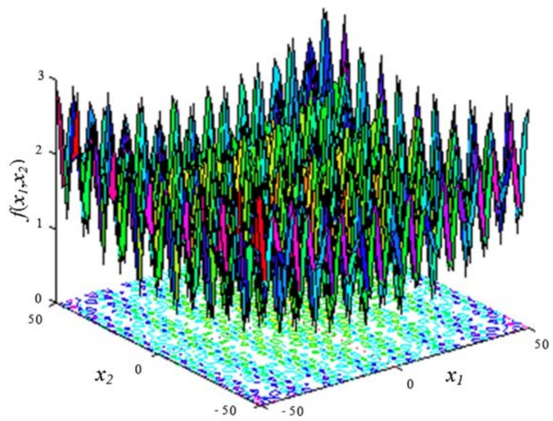
| Index | DPSO | FPSO | HSA | IHSA | ACO | CACO | BFA | GSA | IGSA | SOA | ICA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| MIN | 3.13 e −17 | 0.00 | 7.15 e −17 | 0.00 | 1.27 e −18 | 3.45e16 | 0.00 | 2.13 e −16 | 0.00 | 3.20 e −15 | 5.44 e −15 |
| MEAN | 6.63 e −3 | 5.13 e −4 | 7.65 e −3 | 4.47 e −5 | 5.84 e −3 | 2.45 e −3 | 2.53 e −3 | 2.87 e −3 | 3.42 e −5 | 6.38 e −2 | 3.47 e −3 |
| MAX | 4.73 e −2 | 8.47 e −4 | 1.23 e −2 | 9.88 e −5 | 1.43 e −2 | 9.72 e −3 | 1.22 e −2 | 2.12 e −2 | 7.10 e −5 | 2.20 e −1 | 2.80 e −2 |
| Time (s) | 2.112 | 2.103 | 2.327 | 2.214 | 2.514 | 2.436 | 2.200 | 2.327 | 2.315 | 2.109 | 2.315 |

| Index | ABC | IABC | CABC | PABC | DABC | RABC | MABC | HBMO | IHBMO | HBO | SSO |
|---|---|---|---|---|---|---|---|---|---|---|---|
| MIN | 6.12 e −13 | 2.30 e −15 | 7.43 e −15 | 6.83 e −15 | 0.00 | 0.00 | 0.00 | 2.37 e −15 | 0.00 | 1.22 e −16 | 0.00 |
| MEAN | 2.12 e −1 | 7.53 e −3 | 5.12 e −3 | 1.42 e −2 | 1.67 e −3 | 6.74 e −4 | 5.12 e −2 | 5.73 e −3 | 4.84 e −5 | 4.21 e −3 | 5.23 e −6 |
| MAX | 1.142 | 2.90 e −2 | 2.13 e −2 | 9.87 e −2 | 6.40 e −3 | 4.63 e −3 | 9.84 e −2 | 2.03 e −2 | 7.40 e −5 | 3.26 e −2 | 9.03 e −6 |
| Time (s) | 2.176 | 2.172 | 2.198 | 2.201 | 2.163 | 2.176 | 2.175 | 2.532 | 2.112 | 2.301 | 2.031 |



DOI 10.1002/cplx

**FIGURE 5**




The shape of Griewangk function.

Table 2, it is seen that the MIN result of the proposed SSO is zero and only six other methods, including PSOTVAC, PSOIIW, FPSO, IHSA, IGSA, and RABC can reach this MIN result, while 26 other methods have higher MIN index. Moreover, MEAN and MAX indices of SSO are lower than those of all other methods of Table 2. Finally, SSO has the lowest computation time among all methods of Table 3. These results clearly illustrate high effectiveness of the proposed SSO for solving this benchmark function.

### 4.3. Test Case 3: Schaffer Benchmark Function (Reference of Data: [50])

This test function, represented in (14), is shown in Figure 6.

$$f(x_i)=0.5+\frac{\sin^2(\sqrt{x_1^2+x_2^2})-0.5}{(1+0.001(x_1^2+x_2^2))^2} \quad -100 \le x_i \le 100,\ i=1,2 \tag{14}$$

The results obtained for minimization of this test function are presented in Table 3. The MIN result of SSO and 11 alternative methods of Table 3 are zero, while the MIN index of 21 other methods is higher than zero. The MEAN result of SSO is better than that of 31 other methods of Table 3. Only one method, that is, HGAPSO, has slightly lower MEAN index than SSO (3.63 e −7 vs. 3.74 e −7). The MAX result of SSO is better than all other methods of Table 3. Also, SSO has the lowest computation time among all methods of this Table. These results again reveal high accuracy, robustness, and computational efficiency of the proposed SSO algorithm.

In addition to comparing MIN, MEAN, and MAX results in Tables (1–3), analysis of variance (ANOVA) is also performed on the 30 results (obtained from 30 runs) of the SSO and the methods that can reach to the global optimum in each test case. For instance, for the Rastrigin benchmark function, ANOVA test is run on the 30 results of SSO and the 10 methods of PSOTVAC, PSOIIW, FPSO, IHSA, BFA, IGSA, DABC, RABC, MABC, and IHBMO that have zero MIN in Table 1. The probability of $p$ value [51] obtained from the ANOVA for the four test cases 1–4 ranges from $5.81 \times 10^{-15}$ to $3.32 \times 10^{-18}$. These very small $p$ values indicate significant difference among the results of these algorithms, which means that the differences are unlikely to have occurred by chance.

The Figure 7 is illustrating graphical representations of histograms and normal Q–Q plots for the test cases 1. For the sake of conciseness, only the SSO results are analyzed in the histograms and normal Q–Q plots. A histogram represents a statistical variable by using bars, such that the height of each bar indicates the frequency of the represented values. On each histogram, a normal distribution is fitted, which is indicated by red color on it. A normal Q–Q plot represents the quantiles from the data observed and those from normal distribution against each other [52]. It can be used as an exploratory graphical device to check the validity of normal distribution assumption for a data set. In both the histograms and normal Q–Q plots of Figure 7, the results obtained from 51 runs of SSO, denoted by observed data in the figures, are normalized to be within the range [0,1].

The histogram in Figure 7(a) shows relatively high spread of the observed data and deviations from the normal distributions. From the normal Q–Q plot in Figure 7(b), some deviations from the straight line are seen, which again indicates some deviations of the data set observed from the assumed normal distribution. Also, the points in the Q–Q plots are relatively arced or "S"-shaped indicating that distribution of the observed data is more skewed and has heavier tails than the normal distributions, which can also be seen from the histograms.

### 4.4. Test Case 4:CEC-2013 Testbed

In this section, the proposed SSO is tested on the CEC-2013 testbed in 10- and 30-dimensional environments.-CEC-2013 consists of 28 numerical test functions with different characteristics, which are categorized into three groups as unimodal functions ($f_1$–$f_5$), multimodal functions ($f_6$–$f_{20}$) and composition functions ($f_{21}$–$f_{28}$). The complete description of this testbed is available in [53]. The results obtained from SSO for every function in this testbed in 10- and 30-dimensional environments are compared with the results obtained from IHSA, PSOIIW, and IGSA in Tables 4 and 5. As the number of test functions of this testbed is very large, only IHSA, PSOIIW, and IGSA, which overall have the best results among the 32 benchmark functions for this testbed, are considered for the comparison in Tables 4 and 5 for the sake of conciseness.


DOI 10.1002/cplx

**TABLE 2**

Obtained Results for Griewangk Benchmark Function

| Index | EA | GA | IGA | DE | PSO | IPSO | CPSO | APSO | PSOTVAC | PSOIIW | HGAPSO |
|---|---|---|---|---|---|---|---|---|---|---|---|
| MIN | 6.73 e −10 | 2.20 e −10 | 5.12 e −12 | 6.13 e −12 | 7.50 e −12 | 4.37 e −13 | 2.10 e −14 | 6.03 e −15 | 0.00 | 0.00 | 4.55 e −16 |
| MEAN | 7.44 e −1 | 7.12 e −1 | 5.63 e −1 | 6.03 e −1 | 4.52 e −1 | 7.64 e −3 | 6.57 e −3 | 6.02 e −3 | 3.65 e −4 | 2.31 e −4 | 2.53 e −3 |
| MAX | 2.897 | 3.136 | 2.114 | 2.563 | 9.87 e −1 | 8.91 e −2 | 8.54 e −2 | 7.32 e −2 | 8.37 e −4 | 7.90 e −4 | 6.54 e −3 |
| Time (s) | 3.452 | 3.401 | 3.204 | 3.121 | 3.014 | 2.903 | 2.803 | 2.913 | 2.705 | 2.601 | 3.103 |

| Index | DPSO | FPSO | HSA | IHSA | ACO | CACO | BFA | GSA | IGSA | SOA | ICA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| MIN | 7.17-16 | 0.00 | 1.01 e −16 | 0.00 | 6.66 e −15 | 5.50-17 | 3.04 e −17 | 7.82 e −16 | 0.00 | 9.45 e −16 | 7.31 e −16 |
| MEAN | 4.16 e −3 | 3.11 e −3 | 5.83 e −3 | 3.41 e −4 | 2.65 e −2 | 3.17 e −3 | 6.64 e −4 | 5.43 e −3 | 5.02 e −4 | 3.15 e −3 | 2.06 e −2 |
| MAX | 8.54 e −3 | 7.02 e −3 | 3.11 e −2 | 4.22 e −3 | 7.42 e −2 | 7.70 e −3 | 3.41 e −3 | 2.71 e −2 | 7.75 e −3 | 7.21 e −3 | 7.32 e −2 |
| Time (s) | 2.890 | 2.901 | 2.910 | 2.817 | 3.301 | 2.902 | 2.866 | 3.017 | 2.809 | 3.167 | 3.362 |

| Index | ABC | IABC | CABC | PABC | DABC | RABC | MABC | HBMO | IHBMO | HBO | SSO |
|---|---|---|---|---|---|---|---|---|---|---|---|
| MIN | 6.23 e −15 | 7.43 e −16 | 2.15 e −18 | 2.79 e −17 | 6.10 e −18 | 0.00 | 2.01 e −18 | 2.16 e −14 | 3.22 e −18 | 4.23 e −15 | 0.00 |
| MEAN | 6.04 e −3 | 5.32 e −3 | 7.14 e −3 | 7.57 e −4 | 7.02 e −4 | 4.67 e −4 | 6.27 e −4 | 4.53 e −3 | 2.88 e −4 | 3.98 e −3 | 2.14 e −4 |
| MAX | 5.81 e −2 | 7.34 e −3 | 4.17 e −2 | 9.20 e −3 | 8.11 e −3 | 9.46 e −4 | 9.83 e −4 | 7.02 e −3 | 7.80 e −4 | 7.14 e −3 | 6.93 e −4 |
| Time (s) | 3.044 | 2.855 | 2.783 | 2.714 | 2.836 | 2.812 | 2.943 | 3.163 | 2.892 | 3.145 | 2.573 |



DOI 10.1002/cplx

**TABLE 3**

Obtained Results for Schaffer Benchmark Function

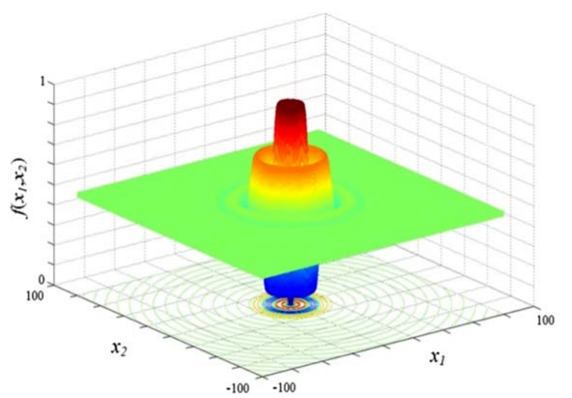
| Index | EA | GA | IGA | DE | PSO | IPSO | CPSO | APSO | PSOTVAC | PSOIIW | HGAPSO |
|---|---|---|---|---|---|---|---|---|---|---|---|
| MIN | 5.12 e −11 | 6.45 e −11 | 7.45 e −12 | 6.33-12 | 3.26 e −13 | 2.38 e −14 | 8.74 e −15 | 5.87 e −16 | 0.00 | 0.00 | 8.43 e −16 |
| MEAN | 5.21 e −4 | 5.13 e −5 | 5.08 e −5 | 5.43 e −5 | 4.98 e −6 | 6.33 e −6 | 4.96 e −6 | 7.30 e −5 | 3.17 e −5 | 2.87 e −6 | 3.63 e −7 |
| MAX | 6.72 e −3 | 8.94 e −4 | 7.02 e −4 | 6.54 e −4 | 3.25 e −5 | 6.74 e −5 | 5.73 e −5 | 3.12 e −4 | 3.84 e −4 | 4.35 e −5 | 4.65 e −6 |
| Time (s) | 3.324 | 3.201 | 2.918 | 2.551 | 2.109 | 2.110 | 2.111 | 2.108 | 2.119 | 2.102 | 2.098 |

| Index | DPSO | FPSO | HSA | IHSA | ACO | CACO | BFA | GSA | IGSA | SOA | ICA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| MIN | 0.00 | 0.00 | 2.31 e −15 | 0.00 | 4.48 e −14 | 2.33 e −16 | 0.00 | 3.42 e −15 | 0.00 | 2.03 e −15 | 6.73 e −16 |
| MEAN | 2.66 e −6 | 5.43 e −6 | 4.40 e −6 | 6.43 e −7 | 8.67 e −5 | 4.49 e −5 | 4.02 e −5 | 5.83 e −5 | 2.21 e −6 | 9.45 e −6 | 4.83 e −5 |
| MAX | 7.28 e −6 | 3.67 e −5 | 3.02 e −5 | 7.78 e −6 | 9.72 e −4 | 4.43 e −4 | 3.12 e −4 | 5.35 e −4 | 7.35 e −6 | 3.70 e −5 | 2.73 e −4 |
| Time (s) | 2.099 | 2.101 | 2.322 | 2.303 | 2.514 | 2.436 | 2.200 | 2.310 | 2.315 | 2.129 | 2.315 |

| Index | ABC | IABC | CABC | PABC | DABC | RABC | MABC | HBMO | IHBMO | HBO | SSO |
|---|---|---|---|---|---|---|---|---|---|---|---|
| MIN | 4.35 e −15 | 6.57 e −17 | 3.42 e −16 | 4.32 e −17 | 0.00 | 0.00 | 0.00 | 5.36 e −15 | 0.00 | 4.22 e −16 | 0.00 |
| MEAN | 3.17 e −5 | 6.03 e −6 | 4.57 e −7 | 5.33 e −6 | 4.72 e −6 | 2.13 e −6 | 3.02 e −6 | 5.11 e −6 | 1.78 e −6 | 2.20 e −6 | 3.74 e −7 |
| MAX | 5.63 e −5 | 5.37 e −5 | 4.34 e −6 | 9.72 e −5 | 4.19 e −5 | 9.34 e −6 | 7.35 e −6 | 7.41 e −5 | 9.26 e −6 | 8.93 e −6 | 9.33 e −7 |
| Time (s) | 2.176 | 2.172 | 2.138 | 2.201 | 2.132 | 2.126 | 2.125 | 2.497 | 2.480 | 2.301 | 2.055 |


DOI 10.1002/cplx

FIGURE 6




The shape of Schaffer function.

Also, only MIN, MEAN, and MAX results among 30 runs are reported for each method in these tables.

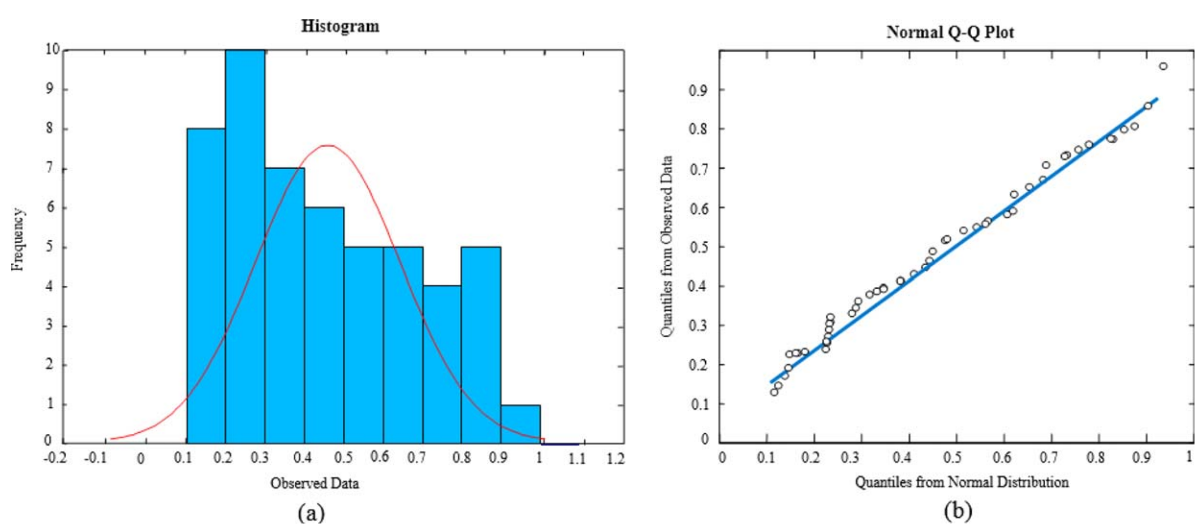
In Tables 4 and 5, if a benchmark method can reach the result of SSO, it is indicated by yellow color and if a benchmark method can obtain a result better than the result of SSO, it is indicated by red color. For instance, in Table 4, IHSA can reach the same MIN result of SSO for $f_5$ and obtain a better MAX index than SSO for $f_5$. In the other cases of Tables 4 and 5, indicated by black color, the results of SSO are better than the results of the benchmark methods. These tables clearly show that only in few cases, the benchmark methods can reach the result of SSO or obtain slightly better results, while in most of cases, SSO has better result than the benchmark methods. The comparisons of Tables 4 and 5 illustrate the effectiveness of SSO over a large number (totally 56) of complex benchmark functions.

It should be mentioned that SSO is a basic metaheuristic optimization algorithm inspired from the nature, similar to several other basic metaheuristic algorithms such as GA, PSO, DE, BF, and ABC. Many researchers have worked on these basic algorithms in the last years and developed enhanced versions of them, for example, by hybridizing these algorithms or adding extra search operators and adaptation mechanisms to them. While SSO is essentially in the level of the basic algorithms, it can outperform many other basic algorithms or even enhanced versions of them as shown in the previous numerical experiments. Additionally, SSO can be a good origin for developing a family of heuristic optimization methods such as those originated from GA, PSO, or DE. Also, as SSO has much higher search capability than the other basic algorithms, we expect that effective heuristic optimization methods can be derived from it as a good origin in the future works.

### 4.5. Test Case 5: Load Frequency Control

Electricity market is one of the biggest and most complicated trading systems in the world [54]. All of the trading issues in an electricity market are strictly dependent on the power system stability. Load frequency control (LFC) plays a fundamental role in providing better conditions for the system stability and power exchanges. Moreover, global analysis of electricity markets shows that

FIGURE 7




Test case 1 (Rastrigin function): (a) Histogram and (b) Normal Q-Q Plot.



DOI 10.1002/cplx

## TABLE 4

Obtained Results for CEC 2013 Testbed in 10-Dimensional Environments

| Method | Index | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ |
|---|---|---|---|---|---|---|---|---|
| SSO | MIN | 3.4 e −18 | 3.3 e −10 | 4.1 e −7 | 4.6 e −9 | 5.3 e −9 | 3.1 e −9 | 6.1 e −9 |
| | MEAN | 3.8 e −10 | 7.4 e −9 | 8.5 e −6 | 7.1 e −8 | 5.0 e −8 | 4.6 e −7 | 4.8 e −6 |
| | MAX | 4.3 e −9 | 6.8 e −8 | 5.7 e −5 | 6.3 e −7 | 5.2 e −7 | 5.5 e −6 | 6.1 e −5 |
| IHSA | MIN | 3.4 e −18 | 5.3 e −10 | 5.2 e −7 | 6.1 e −9 | 5.3 e −9 | 3.4 e −9 | 8.0 e −9 |
| | MEAN | 5.2 e −6 | 4.1 e −8 | 8.2 e −4 | 6.1 e −7 | 5.4 e −8 | 4.2 e −6 | 5.5 e −5 |
| | MAX | 6.3 e −5 | 4.8 e −7 | 6.7 e −3 | 7.4 e −6 | 4.2 e −7 | 4.7 e −5 | 4.8 e −4 |
| PSOIIW | MIN | 3.4 e −18 | 6.4 e −10 | 4.3 e −7 | 7.5 e −9 | 6.0 e −9 | 3.3 e −9 | 2.7 e −8 |
| | MEAN | 3.7 e −6 | 5.2 e −8 | 7.5 e −3 | 6.8 e −7 | 6.3 e −8 | 6.0 e −6 | 4.3 e −5 |
| | MAX | 4.5 e −5 | 4.8 e −7 | 7.5 e −2 | 7.2 e −6 | 7.3 e −7 | 3.9 e −5 | 4.4 e −4 |
| IGSA | MIN | 3.4 e −18 | 4.3 e −10 | 5.2 e −7 | 5.3 e −9 | 5.6 e −9 | 3.1 e −9 | 6.4 e −9 |
| | MEAN | 4.8 e −6 | 5.6 e −8 | 3.1 e −4 | 5.1 e −7 | 6.4 e −8 | 6.7 e −6 | 4.1 e −5 |
| | MAX | 5.3 e −5 | 6.5 e −7 | 3.6 e −3 | 5.4 e −6 | 6.5 e −7 | 7.4 e −5 | 5.2 e −4 |

| Method | Index | $f_8$ | $f_9$ | $f_{10}$ | $f_{11}$ | $f_{12}$ | $f_{13}$ | $f_{14}$ |
|---|---|---|---|---|---|---|---|---|
| SSO | MIN | 2.5 e −1 | 4.2 e −9 | 2.3 e −9 | 3.1 e −10 | 5.3 e −4 | 3.2 e −7 | 3.2 e −5 |
| | MEAN | 3.4e0 | 4.4 e −3 | 7.2 e −8 | 5.3 e −9 | 6.2 e −3 | 4.3 e −4 | 4.2 e −3 |
| | MAX | 8.5e0 | 3.9 e −2 | 7.3 e −7 | 5.4 e −8 | 5.8 e −2 | 3.8 e −3 | 5.2 e −2 |
| IHSA | MIN | 4.2e0 | 4.2 e −9 | 4.2 e −9 | 4.0 e −10 | 4.4 e −3 | 3.4 e −7 | 4.1 e −5 |
| | MEAN | 6.7 e +1 | 5.3 e −3 | 5.1 e −7 | 4.6 e −8 | 3.6 e −2 | 5.1 e −4 | 5.3 e −3 |
| | MAX | 5.4 e +2 | 4.7 e −2 | 4.3 e −6 | 5.1 e −7 | 4.0 e −1 | 4.8 e −3 | 5.2 e −2 |
| PSOIIW | MIN | 3.8e0 | 5.3 e −9 | 4.4 e −9 | 3.1 e −10 | 4.7 e −3 | 3.2 e −7 | 3.4 e −5 |
| | MEAN | 5.7 e +1 | 7.5 e −2 | 5.3 e −7 | 1.4 e −7 | 4.8 e −2 | 4.0 e −4 | 6.4 e −3 |
| | MAX | 4.6 e +2 | 6.2 e −1 | 4.8 e −6 | 5.0 e −7 | 4.5 e −1 | 4.3 e −3 | 5.6 e −2 |
| IGSA | MIN | 3.6e0 | 4.2 e −9 | 3.2 e −9 | 4.1 e −10 | 5.2 e −3 | 3.2 e −7 | 3.2 e −5 |
| | MEAN | 5.0 e +1 | 4.2 e −3 | 3.4 e −7 | 5.8 e −8 | 4.1 e −2 | 4.8 e −4 | 4.1 e −3 |
| | MAX | 4.0 e +2 | 5.1 e −2 | 2.5 e −6 | 6.3 e −7 | 5.0 e −1 | 5.1 e −3 | 5.4 e −2 |

| Method | Index | $f_{15}$ | $f_{16}$ | $f_{17}$ | $f_{18}$ | $f_{19}$ | $f_{20}$ | $f_{21}$ |
|---|---|---|---|---|---|---|---|---|
| SSO | MIN | 3.3 e −2 | 3.3 e −3 | 2.4 e −4 | 3.5 e −3 | 1.2 e −7 | 2.8 e −3 | 4.8 e −2 |
| | MEAN | 5.2e0 | 5.4 e −2 | 4.2 e −3 | 5.0 e −1 | 5.5 e −5 | 4.3 e −2 | 4.0e0 |
| | MAX | 5.3 e +1 | 4.3 e −1 | 3.6 e −2 | 5.8e0 | 4.8 e −4 | 5.4 e −1 | 4.5 e +1 |
| IHSA | MIN | 6.8 e −2 | 4.1 e −3 | 3.2 e −3 | 6.8 e −3 | 1.2 e −7 | 3.0 e −3 | 6.2 e −2 |
| | MEAN | 5.0 e +1 | 4.5 e −2 | 4.3 e −2 | 3.1e0 | 5.3 e −4 | 5.1 e −2 | 5.9e0 |
| | MAX | 6.1 e +2 | 4.7 e −1 | 3.5 e −1 | 4.2 e +1 | 6.6 e −3 | 5.8 e −1 | 7.2 e +1 |
| PSOIIW | MIN | 7.3 e −2 | 3.5 e −3 | 2.6 e −3 | 7.8 e −3 | 1.2 e −7 | 2.8 e −3 | 6.1 e −2 |
| | MEAN | 4.3 e +1 | 4.6 e −2 | 4.8 e −2 | 3.0e0 | 4.0 e −4 | 3.9 e −2 | 4.8e0 |
| | MAX | 5.1 e +2 | 4.8 e −1 | 5.8 e −1 | 5.1 e +1 | 5.6 e −3 | 5.5 e −1 | 6.2 e +1 |
| IGSA | MIN | 7.1 e −2 | 3.4 e −3 | 3.5 e −3 | 6.7 e −3 | 1.2 e −7 | 2.8 e −3 | 6.1 e −2 |
| | MEAN | 4.2 e +1 | 4.4 e −2 | 3.4 e −2 | 3.2e0 | 5.4 e −4 | 4.3 e −2 | 4.8e0 |
| | MAX | 5.8 e +2 | 5.0 e −1 | 4.8 e −1 | 4.2 e +1 | 4.8 e −3 | 5.6 e −1 | 5.3 e −+1 |

| Method | Index | $f_{22}$ | $f_{23}$ | $f_{24}$ | $f_{25}$ | $f_{26}$ | $f_{27}$ | $f_{28}$ |
|---|---|---|---|---|---|---|---|---|
| SSO | MIN | 2.2 e −3 | 4.2 e −3 | 6.2 e −2 | 1.3 e −3 | 1.8 e −2 | 2.8 e −3 | 2.7 e −1 |
| | MEAN | 5.1 e −1 | 5.0 e −1 | 3.3e0 | 4.6 e −1 | 6.4e0 | 5.0 e −1 | 4.4e0 |
| | MAX | 5.8e0 | 6.5e0 | 4.4 e +1 | 5.5 e −0 | 7.7 e +1 | 5.8e0 | 5.5 e +1 |
| IHSA | MIN | 1.4 e −2 | 4.2 e −3 | 2.9 e −1 | 3.6 e −3 | 2.0 e −2 | 2.1 e −2 | 3.6 e −1 |
| | MEAN | 3.1e0 | 5.3 e −1 | 3.6 e +1 | 4.8 e −1 | 6.3e0 | 4.1e0 | 6.8e0 |
| | MAX | 4.2 e +1 | 7.4e0 | 4.6 e +2 | 6.4 e −0 | 8.0 e +1 | 5.2 e +1 | 8.2 e +1 |
| PSOIIW | MIN | 2.3 e −2 | 4.6 e −3 | 2.5 e −1 | 1.3 e −3 | 2.2 e −2 | 1.9 e −2 | 2.7 e −1 |
| | MEAN | 4.2e0 | 5.0 e −1 | 3.8 e +1 | 5.2 e −1 | 7.8e0 | 4.3e0 | 5.0e0 |
| | MAX | 5.9 e +1 | 6.8e0 | 4.8 e +2 | 6.3 e −0 | 8.6 e +1 | 5.6 e +1 | 5.4 e +1 |
| IGSA | MIN | 1.9 e −2 | 4.8 e −3 | 3.7 e −1 | 1.6 e −3 | 3.2 e −2 | 2.1 e −2 | 2.7 e −1 |
| | MEAN | 5.0e0 | 5.9 e −1 | 3.9 e +1 | 4.3 e −1 | 7.0e0 | 5.4e0 | 4.2e0 |
| | MAX | 6.6 e +1 | 5.8e0 | 5.6 e +2 | 5.4 e −0 | 8.3 e +1 | 6.2 e +1 | 5.7 e +1 |

## TABLE 5

Obtained Results for CEC 2013 Testbed in 30-Dimensional Environments

| Method | Index | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ |
|---|---|---|---|---|---|---|---|---|
| SSO | MIN | 3.1 e −14 | 3.4 e −7 | 2.3 e −8 | 2.1 e −3 | 2.5 e −10 | 3.1 e −6 | 2.3 e −6 |
| | MEAN | 5.8 e −8 | 3.7 e −6 | 4.0 e −5 | 4.3 e −1 | 4.3 e −8 | 5.7 e −5 | 5.5 e −5 |
| | MAX | 6.9 e −7 | 5.4 e −5 | 5.9 e −4 | 5.2e0 | 5.2 e −7 | 6.8 e −4 | 4.2 e −4 |
| IHSA | MIN | 6.5 e −14 | 6.7 e −7 | 4.0 e −8 | 3.3 e −2 | 2.5 e −10 | 5.8 e −6 | 4.2 e −6 |
| | MEAN | 4.4 e −7 | 3.5 e −6 | 3.8 e −5 | 4.5e0 | 4.0 e −8 | 7.0 e −4 | 5.5 e −4 |
| | MAX | 3.4 e −6 | 5.4 e −5 | 6.3 e −4 | 5.2 e +1 | 5.8 e −7 | 6.4 e −3 | 4.4 e −3 |
| PSOIIW | MIN | 3.4 e −14 | 6.7 e −7 | 4.4 e −8 | 2.1 e −3 | 4.5 e −10 | 6.0 e −6 | 3.8 e −6 |
| | MEAN | 4.9 e −7 | 2.1 e −5 | 7.8 e −5 | 5.8 e −1 | 6.5 e −8 | 3.9 e −4 | 5.5 e −4 |
| | MAX | 6.2 e −6 | 4.8 e −4 | 9.3 e −4 | 5.0e0 | 8.0 e −7 | 3.3 e −3 | 6.3 e −3 |
| IGSA | MIN | 3.1 e −14 | 7.2 e −7 | 4.3 e −8 | 3.8 e −3 | 2.6 e −10 | 5.5 e −6 | 4.5 e −6 |
| | MEAN | 4.3 e −7 | 3.7 e −6 | 4.3 e −5 | 4.4 e −1 | 6.6 e −7 | 7.2 e −4 | 2.7 e −4 |
| | MAX | 5.6 e −6 | 4.4 e −5 | 5.9 e −4 | 5.6e0 | 8.0 e −6 | 6.3 e −3 | 3.9 e −3 |

| Method | Index | $f_8$ | $f_9$ | $f_{10}$ | $f_{11}$ | $f_{12}$ | $f_{13}$ | $f_{14}$ |
|---|---|---|---|---|---|---|---|---|
| SSO | MIN | 2.3 e −2 | 3.1 e −4 | 3.6 e −9 | 3.3 e −9 | 5.4 e −4 | 5.2 e −4 | 2.6 e −3 |
| | MEAN | 2.3e0 | 5.5 e −3 | 6.3 e −8 | 3.2 e −7 | 5.5 e −3 | 1.1 e −3 | 5.5 e −2 |
| | MAX | 3.8 e +1 | 6.8 e −2 | 7.5 e −7 | 4.6 e −6 | 6.8 e −2 | 1.8 e −2 | 6.8 e −1 |
| IHSA | MIN | 2.4 e −1 | 4.8 e −3 | 3.6 e −9 | 3.3 e −9 | 6.8 e −4 | 5.5 e −4 | 3.3 e −2 |
| | MEAN | 4.3e0 | 5.6 e −2 | 5.4 e −7 | 5.0 e −7 | 5.5 e −3 | 2.5 e −3 | 7.2 e −1 |
| | MAX | 5.6 e +1 | 6.5 e −1 | 6.6 e −6 | 6.3 e −6 | 6.9 e −2 | 3.4 e −2 | 7.7e0 |
| PSOIIW | MIN | 2.4 e −1 | 5.7 e −3 | 5.4 e −9 | 4.1 e −9 | 6.3 e −4 | 5.3 e −4 | 4.0 e −2 |
| | MEAN | 3.4e0 | 5.3 e −2 | 4.3 e −7 | 6.2 e −7 | 5.4 e −3 | 2.3 e −3 | 5.2 e −1 |
| | MAX | 4.7 e +1 | 6.5 e −1 | 5.2 e −6 | 7.3 e −6 | 6.9 e −2 | 3.4 e −2 | 6.7e0 |
| IGSA | MIN | 3.2 e −1 | 3.2 e −3 | 3.6 e −9 | 3.4 e −9 | 5.4 e −4 | 5.2 e −4 | 3.3 e −2 |
| | MEAN | 4.5 e +1 | 4.5 e −2 | 1.7 e −7 | 6.4 e −7 | 5.9 e −3 | 3.8 e −3 | 8.0 e −1 |
| | MAX | 5.5 e +2 | 5.4 e −1 | 2.5 e −6 | 7.4 e −6 | 7.1 e −2 | 5.4 e −2 | 7.5e0 |

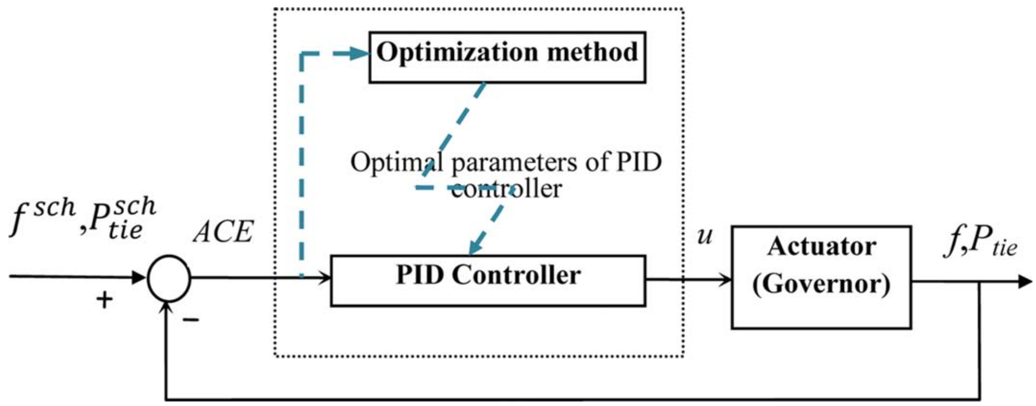
| Method | Index | $f_{15}$ | $f_{16}$ | $f_{17}$ | $f_{18}$ | $f_{19}$ | $f_{20}$ | $f_{21}$ |
|---|---|---|---|---|---|---|---|---|
| SSO | MIN | 1.5 e −2 | 3.2 e −3 | 1.6 e −3 | 3.7 e −2 | 4.7 e −5 | 3.5 e −2 | 3.8 e −1 |
| | MEAN | 7.2e0 | 2.2 e −2 | 2.6 e −2 | 4.5 e −1 | 6.5 e −4 | 3.3 e −1 | 3.1e0 |
| | MAX | 8.4 e +1 | 3.4 e −1 | 3.7 e −1 | 5.5e0 | 7.8 e −3 | 4.7e0 | 4.0 e +1 |
| IHSA | MIN | 3.2 e −2 | 3.5 e −3 | 2.0 e −3 | 5.0 e −2 | 3.1 e −4 | 3.1 e −1 | 4.0 e −1 |
| | MEAN | 7.4e0 | 4.5 e −2 | 3.3 e −2 | 4.4 e −1 | 7.0 e −3 | 3.5e0 | 4.5e0 |
| | MAX | 8.2 e +1 | 5.3 e −1 | 4.4 e −1 | 5.6e0 | 7.3 e −2 | 4.2 e +1 | 6.0 e +1 |
| PSOIIW | MIN | 1.5 e −2 | 4.6 e −3 | 3.4 e −3 | 6.0 e −2 | 4.6 e −4 | 4.0 e −1 | 4.1 e −1 |
| | MEAN | 7.4e0 | 4.0 e −2 | 5.1 e −2 | 3.6 e +1 | 5.4 e −3 | 3.4e0 | 5.3e0 |
| | MAX | 8.3 e +1 | 5.1 e −1 | 6.4 e −1 | 4.4 e +2 | 6.7 e −2 | 4.7+1 | 4.2 e +1 |
| IGSA | MIN | 1.5 e −2 | 4.4 e −3 | 4.0 e −3 | 5.2 e −2 | 4.3 e −4 | 1.9 e −1 | 4.0 e −1 |
| | MEAN | 5.2 e +1 | 6.5 e −2 | 4.3 e −2 | 8.3 e +1 | 6.6 e −3 | 5.0e0 | 4.8e0 |
| | MAX | 4.3 e +2 | 7.6 e −1 | 5.2 e −1 | 8.9 e +2 | 7.8 e −2 | 5.7 e +1 | 6.2 e +1 |

| Method | Index | $f_{22}$ | $f_{23}$ | $f_{24}$ | $f_{25}$ | $f_{26}$ | $f_{27}$ | $f_{28}$ |
|---|---|---|---|---|---|---|---|---|
| SSO | MIN | 3.7 e −1 | 3.4 e −1 | 2.3 e −1 | 1.2 e −1 | 1.2 e −1 | 3.3 e −2 | 1.8 e −1 |
| | MEAN | 4.2e0 | 4.3e0 | 3.8e0 | 4.2e0 | 1.5e0 | 4.5e0 | 3.4e0 |
| | MAX | 5.3 e +1 | 5.4 e +1 | 2.6 e +1 | 6.5 e +1 | 2.9 e +1 | 5.1 e +1 | 4.3 e +1 |
| IHSA | MIN | 3.7 e −1 | 4.3 e −1 | 2.4 e −1 | 2.7 e −1 | 2.4 e −1 | 3.6 e −1 | 1.8 e −1 |
| | MEAN | 5.3e0 | 6.0e0 | 4.4 e +1 | 4.2e0 | 2.7e0 | 5.4 e +1 | 4.9e0 |
| | MAX | 6.1 e +1 | 7.1 e +1 | 3.2 e +2 | 6.4 e +1 | 4.3 e +1 | 7.1 e +2 | 6.3 e +1 |
| PSOIIW | MIN | 3.7 e −1 | 3.4 e −1 | 4.0 e −1 | 4.4 e −1 | 3.4 e −1 | 4.1 e −1 | 1.8 e −1 |
| | MEAN | 5.7e0 | 5.5 e +1 | 4.4 e +1 | 5.5 e +1 | 4.0e0 | 4.3 e +1 | 4.5e0 |
| | MAX | 6.3 e +1 | 5.1 e +2 | 5.2 e +2 | 4.4 e +2 | 5.4 e +1 | 5.4 e +2 | 6.2 e +1 |
| IGSA | MIN | 3.7 e −1 | 4.7 e −1 | 2.3 e −1 | 3.8 e −1 | 1.4 e −1 | 4.4 e −1 | 1.8 e −1 |
| | MEAN | 5.4e0 | 6.6 e +1 | 4.3 e +1 | 3.5 e +1 | 3.4e0 | 4.2 e +1 | 3.3e0 |
| | MAX | 6.3 e +1 | 5.6 e +2 | 5.7 e +2 | 4.5 e +2 | 4.5 e +1 | 5.2 e +2 | 4.4 e +1 |



DOI 10.1002/cplx

FIGURE 8




Application of PID controller for LFC.

frequency control trade is one of the most profitable ancillary services in these markets [3]. Thus, LFC is a technically and economically important real-world optimization problem for power systems and electricity markets. Its model can be briefly described as follows.

Proportional-integrator-differentiator (PID) controller is usually used in LFC to provide the stability of power system. The gain parameters of the controller including $K_P$ $K_I$ and $K_D$ should be fine-tuned as decision variables by an optimization method for the effective performance of LFC. The PID controller in Laplace domain can be represented as follows:

$$\mathrm{PID}=k_P+\frac{k_I}{s}+k_D s \tag{15}$$

The constraints of the parameters of the PID controller can be described as below:

$$\begin{aligned} K_P^{\min} &\le K_P \le K_P^{\max} \\ K_I^{\min} &\le K_I \le K_I^{\max} \\ K_D^{\min} &\le K_D \le K_D^{\max} \end{aligned} \tag{16}$$

where the minimum and maximum values of the decision variables are usually taken as 0.001 and 10, respectively [11,12,55]. Schematic representation of the PID controller is shown in Figure 8. It is seen that the input of the controller is area control error (ACE), which is a linear combination of the interchange power deviation and frequency deviation of the area denoted by $\Delta P_{tie}$ and $\Delta f$, respectively:

$$\mathrm{ACE}=\Delta P_{tie}+B_f\times\Delta f=(P_{tie}^{sch}-P_{tie})+B_f(f^{sch}-f) \tag{17}$$

where the constant $B_f$, called frequency bias constant, is the coefficient of the linear combination. The superscript "sch" in (17) and Figure 8 indicates scheduled value.

By taking ACE as the input of PID controller, the output, that is, the control signal, becomes as follows based on (15):

$$u=K_P\mathrm{ACE}+K_I\int \mathrm{ACE}dt+K_D\frac{d\mathrm{ACE}}{dt} \tag{18}$$

The control signal $u$ is applied to the actuator, which is the governor of the slack generators in the area, for tuning the frequency $f$ and interchange power $P_{tie}$ of the area.

To evaluate the performance of a LFC mechanism, two objective functions, including integral of time multiplied absolute value of the error (ITAE) and figure of demerit (FD), are usually considered in the literature, which are as follows [11,12,55–58]:

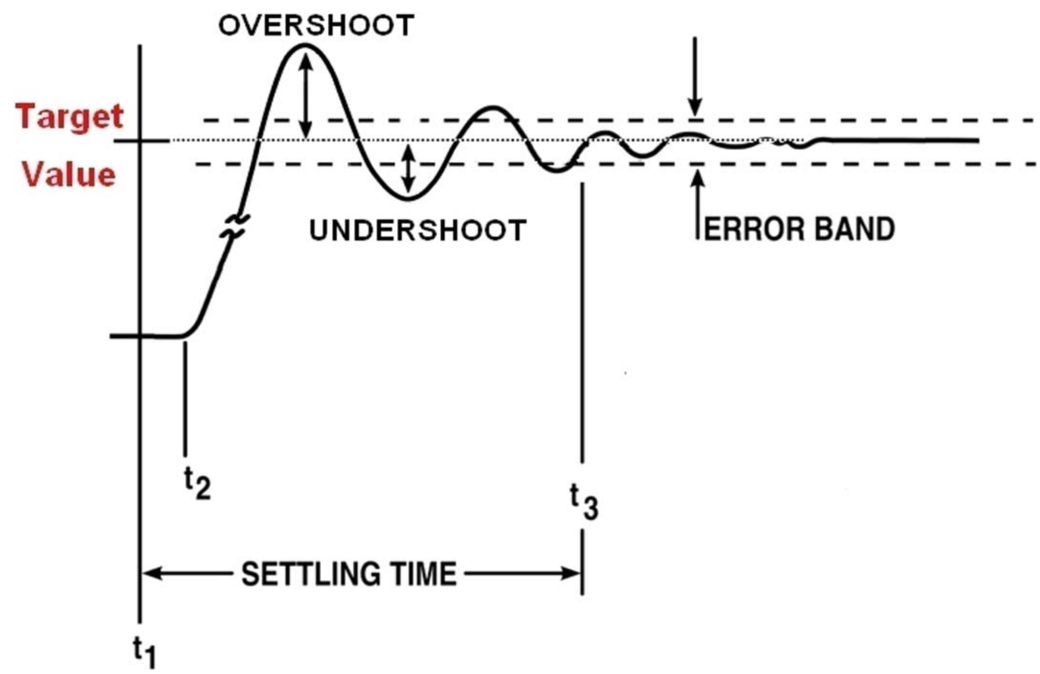
$$\mathrm{ITAE}=\sum_{i=1}^{N_{area}}\int_0^{t_{sim}}(|\mathrm{ACE}_i(t)|)dt \tag{19}$$

$$\mathrm{FD}=\sum_{i=1}^{N_{area}}[(\mathrm{OS}_i\times w)^2+(\mathrm{US}_i\times w)^2+(\mathrm{ST}_i)^2] \tag{20}$$

In (17), $t_{sim}$ indicates the simulation time and $N_{area}$ is the number of control areas (each control area has a LFC mechanism shown in Figure 8. By calculating the integral of the absolute value of ACE signals, ITAE can give a measure of the frequency deviations and interchange power deviations of all areas of the system in response to the disturbance. A lower value of ITAE means smaller deviations or a better system response indicating more effective performance of the controller. Conversely, FD evaluates effectiveness of the controller in terms of common control criteria including overshoot (OS), undershoot (US) and settling time (ST) as shown in (20). These criteria are graphically represented in Figure 9 and their summation over all areas is considered as FD. The allowable error


DOI 10.1002/cplx

**FIGURE 9**




Representation of overshoot (OS), undershoot (US), and settling time (ST).

band, shown in Figure 9, is usually taken as ±2% of the target value. In (20), $w$ is the weight parameter that weighs OS and US (in terms of signal magnitude) versus ST (in terms of time).

The well-known two-area power system, usually used as test case in LFC research works [11,12,55–58], is considered as test system here. Also, large variations in the load of the areas are applied as disturbance to the system. Detailed data of the test system and disturbance can be found in [11,12]. The results obtained from the PID controller with the proposed SSO as the optimization method (Figure 10) in two single-objective cases including ITAE and FD as the objective function, respectively, are shown in Table 6. In this table, the SSO results are compared with the results obtained from eight other optimization methods including numerical analysis (conventional controller) [58], GA [55], BF [58], fuzzy GA [56], fuzzy HPSO [57], fuzzy IABC [11], HBMO [55], and ABC [12]. HPSO stands for hybrid PSO and the other abbreviations have been previously defined. In both the cases, the proposed SSO leads to the best results with the lowest values of ITAE and FD among all methods of Table 6, which shows superiority of SSO for optimizing LFC control mechanism in 30 runs. To better illustrate this matter, system response including frequency deviation of the two areas (indicated by $\Delta f_1$ and $\Delta f_2$, respectively) and deviation in the interchange power between the two areas (indicated by $\Delta P_{12}$) for SSO as well as ABC and HMBO, which have the closest results to SSO among the eight other methods of Table 6, are shown in Figure 10. The results of this figure are related to the second case with FD as the objective function. It is seen that the system response with SSO optimization method has almost no overshoot despite the two other methods. Also, SSO leads to lower undershoot and settling time as well as better convergence behavior compared to the other methods of Figure 10, which reveals effectiveness of SSO for solving this real-world problem.

The convergence plots for this test case with the objective functions of ITAE and FD are shown in Figure 11. In these figures, the convergence curve of the proposed SSO is compared with the convergence curves of ABC, HBMO, and fuzzy IABC, which produce better results for the case study among the eight comparative methods, as shown in Table 6. These figures clearly illustrate better convergence trend as well as better final result of the proposed SSO compared to ABC, HBMO and fuzzy IABC in both the cases.

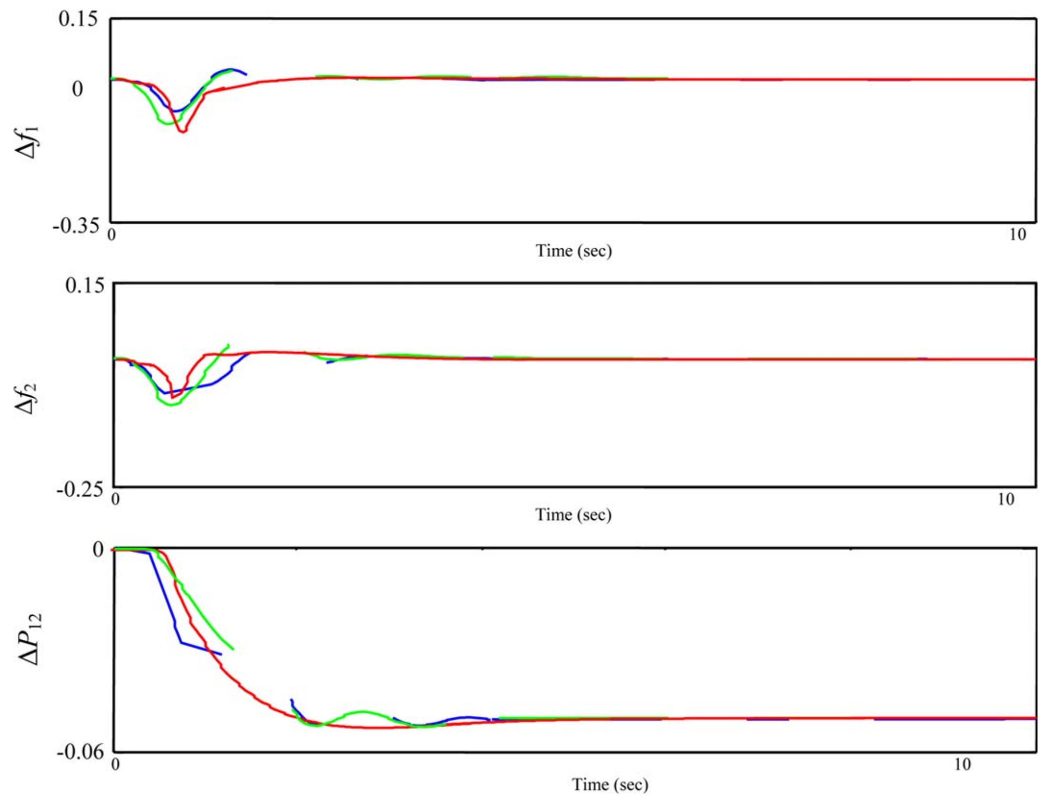
## 5. CONCLUSIONS

Optimization problems in the engineering world become more and more complex, due to, for example,



DOI 10.1002/cplx

FIGURE 10




Deviation of frequency and interchange power of the two areas of the test system: solid red (SSO), dashed green (HBMO), and dash-dotted blue (ABC).

higher nonlinearities and nonconvexities, challenging the numerical optimization methods. As another alternative to tackle with these problems, metaheuristic optimization approaches, usually inspired from natural phenomena (related to, e.g., micro-organisms and animals), have been proposed in the last decades. These approaches have high flexibility for modeling complex optimization problems and so their application in the engineering world increasingly grows.

Since about 400 million years ago that the first sharks appeared in the oceans, they have been known as superior hunters in the nature. An important reason for the success of sharks as a hunter is related to their high ability for finding prey. Powerful smell sense of a shark can locate the prey and effectively guide the shark toward it. This great natural phenomenon motivates the current research work. In this article, shark's search based on the smell sense for finding a prey is modeled as an optimization algorithm. Different behaviors of a shark in the search process such as following the odor, forward and rotational movements and attack are formulated and combined as a metaheuristic optimization approach. Effectiveness of the proposed approach compared to the many other metaheuristic methods is extensively illustrated on several benchmark functions and a real-world optimization problem. This effectiveness is due to the search operators of SSO. While momentum-incorporated gradient-based forward movement smoothly and adaptively tunes direction and velocity of the algorithm toward optimal solution,

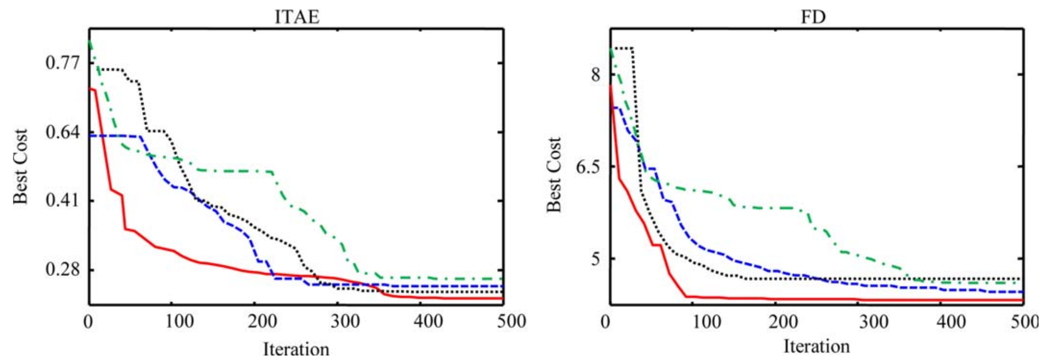
TABLE 6

Obtained Results for LFC of the Test System in Two Cases Including ITAE and FD as the Objective Function

| Method | ITAE | FD |
|---|---|---|
| Conventional controller | 3.5795 | 256.121 |
| GA | 2.554 | 194.568 |
| BF | 1.827 | 173.344 |
| Fuzzy GA | 0.647 | 162.919 |
| Fuzzy HPSO | 0.484 | 87.795 |
| Fuzzy IABC | 0.176 | 4.602 |
| HBMO | 0.145 | 4.763 |
| ABC | 0.154 | 4.425 |
| SSO | 0.141 | 4.102 |


DOI 10.1002/cplx

FIGURE 11




Convergence plot with ITAE and FD as the objective functions (Red solid line: SSO, Blue dashed line: ABC, Black dotted line: HBMO, green dash-dot line: fuzzy IABC).

rotational movements give the ability of local search around the found solutions to SSO. In this way, the proposed SSO can both discover different areas of the solution space (leading to its high exploration capability) and search within these areas with high resolution (resulting in its high exploitation capability).

Despite these capabilities, there is still much room for extending this research in the future works. In modeling the search process of shark some simplifying assumptions are considered in this article. By removing these assumptions and a more accurate modeling of the injected blood distribution in the sea water, a more realistic form of the shark's search process can be obtained. Moreover, various evolutionary and search operators can be added to the proposed SSO to construct enhanced versions of it similar to the other basic metaheuristic optimization methods.

## REFERENCES


1. López-Corona, O.; Padilla, P.; Escolero, O.; Armas, F.; García-Arrazola, R.; Esparza, R. Playing with models and optimization to overcome the tragedy of the commons in groundwater. Complexity 2013, 19, 9–21. doi:10.1002/cplx.21462.
2. Mohammadi, M.; Ghadimi, N. Optimal location and optimized parameters for robust power system stabilizer using honeybee mating optimization. Complexity 2014. doi:10.1002/cplx.21560.
3. Hays W.O. History of mathematical programming systems, Ann Hist Comput 1984, 6, 296–312.
4. Noruzi, A.; Banki, T.; Abedinia, O.; Ghadimi, N. A new method for probabilistic assessments in power systems, combining monte carlo and stochastic-algebraic methods. Complexity 2014. doi:10.1002/cplx.21582.
5. Eskandari Nasab, M.; Maleksaeedi, I.; Mohammadi, M.; Ghadimi, N. A new multiobjective allocator of capacitor banks and distributed generations using a new investigated differential evolution. Complexity 2014, 19, 40–54. doi:10.1002/cplx.21489.
6. Abedinia, O.; Amjady, N.; Ghasemi, A.; Hejrat, Z. Solution of economic load dispatch problem via hybrid particle swarm optimization with time-varying acceleration coefficients and bacteria foraging algorithm techniques, Eur Trans Electr Power 2012, 23, 1504–1522. doi:10.1002/etep.1674.
7. Parpinelli, R.S.; Lopes, H.S.; Freitas A.A., Data mining with an ant colony optimization algorithm, IEEE Trans Evolut Comput 2002, 6, 321–332.
8. Maleksaeedi, I.; Khiav, B.E.; Germi, M.B.; Ghadimi, N. A new two-stage algorithm for solving power flow tracing. Complexity 2014. doi:10.1002/cplx.21555.
9. Ahadi, A.; Ghadimi, N.; Mirabbasi, D. An analytical methodology for assessment of smart monitoring impact on future electric power distribution system reliability. Complexity 2014. doi:10.1002/cplx.21546.
10. Shayanfar, H.A.; Barazandeh, E.S.; SeyedShenava, S.J.; Ghasemi, A.; Abedinia, O. Solving optimal unit commitment by improved honey bee mating optimization, Int J Tech Phys Probl Eng 2012, 4, 38–45.
11. Javidan J.; Ghasemi A. A novel fuzzy RPID controller for multiarea AGC with IABC optimization, J Eng 2013, 2013, 1–13.
12. Abedinia, O.; Shayanfar, H.A.; Wyns, B.; Ghasemi, A. Design of robust PSS to improve stability of composed LFC and AVR using ABC in deregulated environment. In: Proceedings of the International Conference on Artificial Intelligence, Las Vegas, Nevada, USA, July 2011, pp. 82–88.
13. Passino, K.M. Biomimicry of bacterial foraging for distributed optimization and control, IEEE Control Syst Mag 2002, 22, 52–67.
14. de Castro, L.N.; Von Zuben, F.J. Learning and optimization using the clonal selection principle, IEEE Trans Evolut Comput, 2002, 6, 239–251.
15. Karimkashi, S.; Kishk A.A. Invasive weed optimization and its features in electromagnetics, IEEE Trans Antennas Propag 2010, 58, 1269–1278.




DOI 10.1002/cplx

16. Eusuff, M.; Lansey, K.; Pasha, F. Shuffled frog-leaping algorithm: A memetic meta-heuristic for discrete optimization, Eng Optim 2006, 38, 129–154.
17. Hornby, G.S.; Pollack, J.B. Creating high-level components with a generative representation for body-brain evolution, Artif Life 2002, 8, 223–246.
18. Hagh, M.T.; Ghadimi, N. Multisignal histogram-based islanding detection using neuro-fuzzy algorithm. Complexity 2014. doi:10.1002/cplx.21556.
19. Černý V. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm, J Optim Theory Appl 1985, 45, 41–51.
20. Ghadimi, N. An adaptive neuro-fuzzy inference system for islanding detection in wind turbine as distributed generation. Complexity 2014. doi:10.1002/cplx.21537.
21. Khajehzadeh, M.; Tahaa, M.R.; El-Shafiea, A.; Eslamib, M. A modified gravitational search algorithm for slope stability analysis, Eng Appl Artif Intell 2012, 1589–1597.
22. Morsali, R.; Ghadimi, N.; Karimi, M.; Mohajeryami, S. Solving a novel multiobjective placement problem of recloser and distributed generation sources in simultaneous mode by improved harmony search algorithm. Complexity 2014. doi:10.1002/cplx.21567.
23. Costa, D.P.; Sinervo, B. Field physiology: Physiological insights from animals in nature, Annu Rev Physiol 2004, 66, 209–238.
24. Eilperin, J. Sharks are scary but not that dangerous, The Daily Beast. Newsweek/Daily Beast. Available at: http://www.the-dailybeast.com/articles/2011/06/16/sharks-are-scary-but-, not-that-dangerous.html. Accessed on November, 2012.
25. Zappler G. Learn about Texas freshwater fishes—Texas parks & wildlife, Texas Parks and Wildlife Press, USA, 2001.
26. Magnuson, J.J. 4 Locomotion by scombrid fishes: Hydromechanics, morphology and behavior, Fish Physiol 1979, 7, 239–313.
27. Sfakiotakis, M.; Lane, D.M.; Davies, J.B.C. Review of fish swimming modes for aquatic locomotion, IEEE J Oceanic Eng 1999, 24, 237–252.
28. Wooldridge, M.A. Shark World Teacher Guide, Marine World Foundation, 1990, pp. 1–24.
29. Yao-Tsu Wu, T. Hydromechanics of swimming propulsion. Part 1. Swimming of a two-dimensional flexible plate at variable forward speeds in an inviscid fluid, J Fluid Mech 1971, 46, 337–355.
30. Gardiner, J.M.; Atema, J. The function of bilateral odor arrival time differences in olfactory orientation of sharks, Curr Biol 2010, 20, 1187–1191.
31. Hush, D.R.; Horne, B.G. Progress in supervised neural networks, IEEE Signal Process Mag, 1993, 10, 8–39.
32. Hongwu, L. An adaptive chaotic particle swarm optimization, Int Colloquium Comput Commun Control Manage 2009, 2, 254–257.
33. Zkaya, U.O.; Unes, F.G. A modified particle swarm optimization algorithm and its application to the multi-objective FET modeling problem, Turk J Electr Eng Comput Sci 2012, 20, 263–271.
34. Chen, L.F.; Su, C.T.; Chen, K.H. An improved particle swarm optimization for feature selection, Intell Anal J 2012, 16, 167–182.
35. Wang, H.; Wu, Z.; Rahnamayan, S.; Liu Y.; Ventresca M. Enhancing particle swarm optimization using generalized opposition-based learning, Inf Sci 2011, 181, 4699–4714,
36. Narasimhan, H. Parallel artificial bee colony (PABC) algorithm, In: the Second World Congress on Nature and Biologically Inspired Computing (NaBIC 2010) will be held at the Kitakyushu International Conference Center, Kitakyushu, Japan, December 2009, pp. 306–311.
37. Zeng, X.J.; Tao, J.; Zhang, P.; Pan, H.; Wang, Y.Y. Reactive power optimization of wind farm based on improved genetic algorithm,. In: 2nd International Conference on Advances in Energy Engineering, Proceedings of a meeting held 27–28 December 2011, Bangkok, Thailand, pp. 1362–1367.
38. Safari, A.; Shayeghi, H. Iteration particle swarm optimization procedure for economic load dispatch with generator constraints, Expert Syst Appl 2011, 38, 6043–6048.
39. Chen, C.L.; Jan, R.M.; Lee, T.Y.; Chen, C.H. A novel particle swarm optimization algorithm solution of economic dispatch with valve point loading, J Mar Sci Technol 2011, 19, 43–51.
40. Sarikhani, A.; Osania, A. Hybrid GA-PSO multi-objective design optimization of coupled PM synchronous motor-drive using physics-based modeling approach. In: 14th Biennial IEEE Conference on Electromagnetic Field Computation (CEFC), Hyatt Regency Hotel O'Hare Chicago, IL, USA 2010, pp. 1–6.
41. Venayagamoorthy, G.K. Adaptive critics for dynamic particle swarm optimization. In: Proceedings of the 2004 IEEE International Symposium on Intelligent Control, September 2004, Sendai, pp. 380–384.
42. Tian, D.; Li, N. Fuzzy particle swarm optimization algorithm. In: 9th International Joint Conference on Artificial Intelligence, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA ©1985, April 2009, pp. 263–267.
43. Wang, L.; Li, L. An effective differential harmony search algorithm for the solving non-convex economic load dispatch problems, Electr Power Energy Syst 2013, 44, 832–843.
44. Dai, C.; Zhu, Y.; Chen, W. Seeker optimization algorithm, Comput Intell Secur, LNCS 2007, 4456, 167–176.
45. Atashpaz-Gargari, E. Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. In: IEEE Congress on Evolutionary Computation, IEEE Congress on Evolutionary Computation (CEC), Singapore, September 2007, pp. 4661–4667.
46. Hong, W.C. Electric load forecasting by seasonal recurrent SVR (support vector regression) with chaotic artificial bee colony algorithm, Energy 2011, 36, 5568–5578.


DOI 10.1002/cplx

47. FatihTasgetiren, M.; Pan, Q.K.; Suganthan, P.N.; Chen, A.H.-L. A discrete artificial bee colony algorithm for the total flow time minimization in permutation flow shops, Inf Sci 2011, 181, 3459–3475.
48. Bahriye, A.; Karaboga, D. A modified artificial bee colony algorithm for real-parameter optimization, Inf Sci 2012, 192, 120–142.
49. Khanmirzaei, Z.; Teshnehlab, M.; Sharifi, A.; Modified honey bee optimization for recurrent neuro-fuzzy system model,. In: The 2nd International Conference on Computer and Automation Engineering (ICCAE), Vol. 5, NEC of Nanyang Technological University Singapore, Singapore, February 2010, pp. 780-785.
50. Karaboga, D.; Akay, B. A comparative study of artificial bee colony algorithm, Appl Math Comput 2009, 214, 108–132.
51. Garcia, S.; Molina, D.; Lozano, M.; Herrera, F. A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 special session on real parameter optimization, J Heuristics 2009, 15, 617–644.
52. Thode, C.H. Testing for Normality, Marcel Dekker: New York, 2002.
53. Liang, J.J.; Qu, B.-Y.; Suganthan, P.N.; Hernández-Díaz, A.G. Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report 201212, 2013.
54. Koritarov, V.S. Real-world market representation with agents, IEEE Power Energy Mag 2004, 2, 39–46.
55. Abedinia, O.; Naderi, M.S.; Ghasemi, A. Robust LFC in deregulated environment: Fuzzy PID using HBMO. In: 10th International Conference on Environment and Electrical Engineering (EEEIC), SGM Conference Center, Rome, Italy, May 2011, pp. 74–77.
56. Chadli, M.; Guerra, T.M. LMI solution for robust static output feedback control of Takagi-Sugeno fuzzy models, IEEE Trans Fuzzy Syst 2012, 20, 1160–1165.
57. Taher, S.; Hematti, R.; Abdolalipour, A.; Tabei, S.H. Optimal decentralized load frequency control using HPSO algorithms in deregulated power systems, Am J Appl Sci 2008, 5, 1167–1174.
58. Ali, E.S.; Abd-Elazim, S.M. Bacteria foraging optimization algorithm based load frequency controller for interconnected power system, Electr Power Energy Syst 2011, 33, 633–638.




DOI 10.1002/cplx