

Research Article

An Improved Squirrel Search Algorithm for Optimization

Tongyi Zheng and Weili Luo 

School of Civil Engineering, Guangzhou University, Guangzhou, China

Correspondence should be addressed to Weili Luo; wlluo@gzhu.edu.cn

Received 15 February 2019; Revised 5 May 2019; Accepted 28 May 2019; Published 1 July 2019

Academic Editor: Alex Alexandridis

Copyright © 2019 Tongyi Zheng and Weili Luo. 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.

Squirrel search algorithm (SSA) is a new biological-inspired optimization algorithm, which has been proved to be more effective for solving unimodal, multimodal, and multidimensional optimization problems. However, similar to other swarm intelligence-based algorithms, SSA also has its own disadvantages. In order to get better global convergence ability, an improved version of SSA called ISSA is proposed in this paper. Firstly, an adaptive strategy of predator presence probability is proposed to balance the exploration and exploitation capabilities of the algorithm. Secondly, a normal cloud model is introduced to describe the randomness and fuzziness of the foraging behavior of flying squirrels. Thirdly, a selection strategy between successive positions is incorporated to preserve the best position of flying squirrel individuals. Finally, in order to enhance the local search ability of the algorithm, a dimensional search enhancement strategy is utilized. 32 benchmark functions including unimodal, multimodal, and CEC 2014 functions are used to test the global search ability of the proposed ISSA. Experimental test results indicate that ISSA provides competitive performance compared with the basic SSA and other four well-known state-of-the-art optimization algorithms.

1. Introduction

Optimization is the process of determining decision variables of a function in a way that the function is in its maximal or minimal value. Many real-life engineering problems belong to optimization problems [1–3] in which decision variables are determined in a way that the systems operate in their best optimal point. Usually, these problems are discontinuous, nondifferentiable, multimodal, and nonconvex, and thus the classical gradient-based deterministic algorithms [4–6] are not applicable.

To overcome the drawbacks of the classical algorithms, a considerable number of stochastic optimization algorithms known as metaheuristic algorithms [7–9] have been developed in recent decades. These algorithms are mainly inspired by biological behaviors or physical phenomena and can be roughly classified into three categories: evolutionary algorithms, swarm intelligence, and physical-based algorithms. The evolutionary algorithms simulate the evolution of the nature such as reproduction, mutation, recombination, and selection, in which a population tries to survive based on

the evaluation of fitness value in a given environment. Genetic algorithm (GA) [10] and evolution strategy (ES) [11] are the most popular evolutionary algorithms. Swarm intelligence algorithms are the second category, which are based on movement of a swarm of creatures and imitate the interaction of swarm and their environment in order to enhance their knowledge of a goal (e.g., food source). The most well-known swarm intelligence algorithms are particle swarm optimization (PSO) [12], artificial bee colony (ABC) [13] algorithm, ant colony optimization (ACO) [14], and grey wolf optimization algorithm (GWO) [15], to name a few. The physical-based algorithms are inspired by the basic physical laws in universe such as gravitational force, electromagnetic force, and inertia force. Several representatives of this category are simulated annealing (SA) [16], big-bang big-crunch (BB-BC) [17], and gravitational search algorithm (GSA) algorithm [18]. Some of the recent nature-inspired algorithms are lightning attachment procedure optimization (LAPO) [19], spotted hyena optimizer (SHO) [20], weighted superposition attraction (WSA) [21], and many more [22, 23].

Unfortunately, most of the abovementioned basic metaheuristic algorithms fail to balance exploration and exploitation, thereby yielding unsatisfactory performance for real-life complicated optimization problems. Exploration stands for global search ability and ensures the algorithm to reach all over the search space and then to find promising regions, whereas exploitation represents local search ability and ensures the searching of optimum within the identified promising regions. Emphasizing the exploration capability only results in a waste of computational resources on searching all over the interior regions of search space and thus reduces the convergence rate; emphasizing the exploitation capability only, by contrast, causes loss of population diversity early and thus probably leads to premature convergence or to be stuck in local optimal. This fact motivates the introduction of various strategies for improving the convergence rate and precision of the basic metaheuristic algorithms. As an illustration, premature convergence of PSO was prevented in CLPSO by proposing a comprehensive learning strategy to maintain the population diversity [24]; a social learning component called fitness-distance-ratio was employed to enhance local search capability of PSO [25]; a self-organizing hierarchical PSO with time-varying acceleration coefficients (HPSO-TVAC) was introduced to efficiently control the local search and convergence to the global optimal solution [26]; a distance-based locally informed PSO (LIPS) enabled the algorithm to quickly converge to global optimal solution with high accuracy [27]. Likewise, many modified versions have been proposed to enhance the global search ability of the basic ABC: an improved-global-best-guide term with a nonlinear adjusting factor was employed to balance the exploration and exploitation [28]; a multiobjective covariance guided artificial bee colony algorithm (M-CABC) was proposed to obtain higher precision with quick convergence speed when solving portfolio problems [29]; the slow convergence and unsatisfactory solution accuracy were improved in the variant IABC [30]. As for fruit fly optimization algorithm (FOA) [31], an escape parameter was introduced in MFOA to escape from the local solution [32]; and modified versions for balancing between exploration and exploitation abilities include, for example, IFOA [33], MSFOA [34], IFFO [35], and CMFOA [36].

Squirrel search algorithm (SSA), proposed by Mohit et al. in 2018, is a new and powerful global optimization algorithm inspired by the natural dynamic foraging behavior of flying squirrels [37]. In comparison with other swarm intelligence optimization algorithms, SSA has the advantages of better and efficient search space exploration because a seasonal monitoring condition is incorporated. Moreover, three types of trees (normal tree, oak tree, and hickory tree) are available in the forest region, preserving the population diversity and thus enhancing the exploration of the algorithm. Test results of 33 benchmark functions and a real-time controller design problem confirm the superiority of SSA in comparison with other well-known algorithms such as GA [10], PSO [25], BA (bat algorithm) [38], and FF (firefly algorithm) [39].

However, SSA still suffers from premature convergence and easily gets trapped in a local optimal solution, especially when solving highly complex problems. The convergence rate

of SSA, like other swarm intelligence algorithms, depends on the balance between exploration and exploitation capabilities. In other words, an excellent performance in dealing with optimization problems requires fine-tuning of the exploration and exploitation problem. According to “no free lunch” (NFL) theorem [40], no single optimization algorithm is able to achieve the best performance for all problems, and SSA is not an exception. Therefore, there still exists room for improving the accuracy and convergence rates of SSA.

Based on the discussion above, this study proposes an improved variant of SSA (ISSA) which employs four strategies to enhance the global search ability of SSA. In brief, the main contributions of this research can be summarized as follows:

(i) An adaptive strategy of predator presence probability is proposed, which dynamically adjusts with the iteration process. This strategy discourages premature convergence and improves the intensive search ability of the algorithm, especially at the latter stages of search. In this way, a balance between the exploration and exploitation capabilities can be properly managed.

(ii) The proposed ISSA employs a normal cloud generator [41] to generate new locations for flying squirrels during the course of gliding, which improves the exploration capability of SSA. This is motivated by the fact that the gliding behaviors of flying squirrels have characteristics of randomness and fuzziness, which can be simultaneously described by the normal cloud model [42].

(iii) A selection strategy between successive positions is proposed to maintain the best position of a flying squirrel individual throughout the optimization process, which enhances the exploitation ability of the algorithm.

(iv) A dimensional search enhancement strategy is originally put forward and results in a better quality of the best solution in each iteration, thereby strengthening the local search ability of the algorithm.

The general properties of ISSA are evaluated against 32 benchmark function including unimodal, multimodal, and CEC 2014 functions [43]. Meanwhile, its performance is compared with the basic SSA and other four well-known state-of-the-art optimization algorithms.

The rest of this paper is organized as follows: Section 2 briefly recapitulates the basic SSA. Next, the proposed ISSA is presented in detail in Section 3. Experimental comparisons are illustrated in Section 4. Finally, Section 5 gives the concluding remarks.

2. The Basic Squirrel Search Optimization Algorithm

SSA mimics the dynamic foraging behavior of southern flying squirrels via gliding, an effective mechanism used by small mammals for travelling long distance, in deciduous forest of Europe and Asia [37]. During warm weather, the squirrels change their locations by gliding from one tree to another in the forest and explore for food resources. They can easily find acorn nuts for meeting daily energy needs. After that, they begin searching hickory nuts (the optimal food source) that are stored for winter. During cold weather, they become

less active and maintain their energy requirements with the storage of hickory nuts. When the weather gets warm, flying squirrels become active again. The abovementioned process is repeated and continues throughout the life space of the squirrels, which serves as a foundation of the SSA. According to the food foraging strategy of flying squirrels, the optimization SSA can be modeled by the following phases mathematically.

2.1. Initialize the Algorithm Parameters. The main parameters of the SSA are the maximum number of iteration $Iter_{max}$, the population size NP , the number of decision variables n , the predator presence probability P_{dp} , the scaling factor sf , the gliding constant G_c , and the upper and lower bounds for decision variable FS_U and FS_L . These parameters are set in the beginning of the SSA procedure.

2.2. Initialize Flying Squirrels' Locations and Their Sorting. The flying squirrels' locations are randomly initialized in the search space as follows:

$$FS_{i,j} = FS_L + \text{rand}() * (FS_U - FS_L), \quad (1)$$

$$i = 1, 2, \dots, NP, \quad j = 1, 2, \dots, n$$

where $\text{rand}()$ is a uniformly distributed random number in the range $[0, 1]$.

The fitness value $f = (f_1, f_2, \dots, f_{NP})$ of an individual flying squirrel's location is calculated by substituting the value of decision variables into a fitness function:

$$f_i = f_i(FS_{i,1}, FS_{i,2}, \dots, FS_{i,n}), \quad i = 1, 2, \dots, NP \quad (2)$$

Then the quality of food sources defined by the fitness value of the flying squirrels' locations is sorted in an ascending order:

$$[\text{sorted_f}, \text{sorte_index}] = \text{sort}(f) \quad (3)$$

After sorting the food sources of each flying squirrel's location, three types of trees are categorized: hickory tree (hickory nuts food source), oak tree (acorn nuts food source), and normal tree. The location of the best food source (i.e., minimal fitness value) is regarded as the hickory nut tree (FS_{ht}), the locations of the following three food sources are supposed to be the acorn nuts trees (FS_{at}), and the rest are considered as normal trees (FS_{nt}):

$$FS_{ht} = FS(\text{sorte_index}(1)) \quad (4)$$

$$FS_{at}(1:3) = FS(\text{sorte_index}(2:4)) \quad (5)$$

$$FS_{nt}(1:NP-4) = FS(\text{sorte_index}(5:NP)) \quad (6)$$

2.3. Generate New Locations through Gliding. Three scenarios may appear after the dynamic gliding process of flying squirrels.

Scenario 1. Flying squirrels on acorn nut trees tend to move towards hickory nut tree. The new locations can be generated as follows:

$$FS_{at}^{new} = \begin{cases} FS_{at}^{old} + d_g G_c (FS_{ht}^{old} - FS_{at}^{old}), & \text{if } R_1 \geq P_{dp} \\ \text{random location}, & \text{otherwise} \end{cases} \quad (7)$$

where d_g is random gliding distance, R_1 is a function which returns a value from the uniform distribution on the interval $[0, 1]$, and G_c is a gliding constant.

Scenario 2. Some squirrels which are on normal trees may move towards acorn nut tree to fulfill their daily energy needs. The new locations can be generated as follows:

$$FS_{nt}^{new} = \begin{cases} FS_{nt}^{old} + d_g G_c (FS_{at}^{old} - FS_{nt}^{old}), & \text{if } R_2 \geq P_{dp} \\ \text{random location}, & \text{otherwise} \end{cases} \quad (8)$$

where R_2 is a function which returns a value from the uniform distribution on the interval $[0, 1]$.

Scenario 3. Some flying squirrels on normal trees may move towards hickory nut tree if they have already fulfilled their daily energy requirements. In this scenario, the new location of squirrels can be generated as follows:

$$FS_{nt}^{new} = \begin{cases} FS_{nt}^{old} + d_g G_c (FS_{ht}^{old} - FS_{nt}^{old}), & \text{if } R_3 \geq P_{dp} \\ \text{random location}, & \text{otherwise} \end{cases} \quad (9)$$

where R_3 is a function which returns a value from the uniform distribution on the interval $[0, 1]$.

In all scenarios, gliding distance d_g is considered to be in the interval between 9 and 20 m [37]. However, this value is quite large and may introduce large perturbations in (7)-(9) and hence may cause unsatisfactory performance of the algorithm. In order to achieve acceptable performance of the algorithm, a scaling factor (sf) is introduced as a divisor of d_g and its value is chosen to be 18 [37].

2.4. Check Seasonal Monitoring Condition. The foraging behavior of flying squirrels is significantly affected by season variations [43]. Therefore, a seasonal monitoring condition is introduced in the algorithm to prevent the algorithm from being trapped in local optimal solutions.

A seasonal constant Sc and its minimum value are calculated firstly:

$$S_c^t = \sqrt{\sum_{k=1}^n (FS_{at,k}^t - FS_{ht,k})^2}, \quad t = 1, 2, 3 \quad (10)$$

$$S_{cmin} = \frac{10E - 6}{365^{Iter/(Iter_{max})/2.5}} \quad (11)$$

Then the seasonal monitoring condition is checked. Under the condition of $S_c^t < S_{cmin}$, the winter is over, and the flying squirrels which lose their abilities to explore the forest will

randomly relocate their searching positions for food source again:

$$FS_{nt}^{new} = FS_L + \text{Lévy}(n) \times (FS_U - FS_L) \quad (12)$$

where Lévy distribution is a powerful mathematical tool to enhance the global exploration capability of most optimization algorithms [44]:

$$\text{Lévy}(x) = 0.01 \times \frac{r_a \times \sigma}{|r_b|^{1/\beta}} \quad (13)$$

where r_a and r_b are two functions which return a value from the uniform distribution on the interval $[0, 1]$, β is a constant ($\beta = 1.5$ in this paper), and σ is calculated as follows:

$$\sigma = \left(\frac{\Gamma(1 + \beta) \times \sin(\pi\beta/2)}{\Gamma((1 + \beta)/2) \times \beta \times 2^{((\beta-1)/2)}} \right)^{1/\beta} \quad (14)$$

where $\Gamma(x) = (x-1)!$.

2.5. Stopping Criterion. The algorithm terminates if the maximum number of iterations is satisfied. Otherwise, the behaviors of generating new locations and checking seasonal monitoring condition are repeated.

2.6. Procedure of the Basic SSA. The pseudocode of SSA is provided in Algorithm 1.

3. The Improved Squirrel Search Optimization Algorithm

This section presents an improved squirrel search optimization algorithm by introducing four strategies to enhance the searching capability of the algorithm. In the following, the four strategies will be presented in detail.

3.1. An Adaptive Strategy of Predator Presence Probability. When flying squirrels generate new locations, their natural behaviors are affected by the presence of predators and this character is controlled by predator presence probability P_{dp} . In the early search stage, flying squirrels' population is often far away from the food source and its distribution range is large; thus it faces a great threat from predators. With the evolution going on, flying squirrels' locations are close to the food source (an optimal solution). In this case, the distribution range of flying squirrels' population is increasingly smaller and less threats from predators are expected. Thus, to enhance the exploitation capacity of the SSA, an adaptive P_{dp} , which dynamically varies as a function of iteration number, is adopted as follows:

$$P_{dp} = (P_{dpmax} - P_{dpmin}) \times (1 - \text{Iter}/\text{Iter}_{max})^{10} + P_{dpmin} \quad (15)$$

where P_{dpmax} and P_{dpmin} are the maximum and minimum predator presence probability, respectively.

3.2. Flying Squirrels' Random Position Generation Based on Cloud Generator. Under the condition of $R_1, R_2, R_3 < P_{dp}$, the flying squirrels randomly proceed gliding to the next potential food locations, different individuals generally have different judgments and their gliding directions and routines vary. In other words, the foraging behavior of flying squirrels has the characteristics of randomness and fuzziness. These characteristics can be synthetically described and integrated by a normal cloud model. In the model, a normal cloud model generator instead of uniformly distributed random functions is used to reproduce new location for each flying squirrel. Thus, (7)-(9) are replaced by the following equations:

$$FS_{at}^{new} = \begin{cases} FS_{at}^{old} + d_g G_c (FS_{nt}^{old} - FS_{at}^{old}), & \text{if } R_1 \geq P_{dp} \\ Cx (FS_{at}^{old}, En, He), & \text{otherwise} \end{cases} \quad (16)$$

$$FS_{nt}^{new} = \begin{cases} FS_{nt}^{old} + d_g G_c (FS_{at}^{old} - FS_{nt}^{old}), & \text{if } R_2 \geq P_{dp} \\ Cx (FS_{nt}^{old}, En, He), & \text{otherwise} \end{cases} \quad (17)$$

$$FS_{nt}^{new} = \begin{cases} FS_{nt}^{old} + d_g G_c (FS_{nt}^{old} - FS_{nt}^{old}), & \text{if } R_3 \geq P_{dp} \\ Cx (FS_{nt}^{old}, En, He), & \text{otherwise} \end{cases} \quad (18)$$

where En (Entropy) represents the uncertainty measurement of a qualitative concept and He (Hyper Entropy) is the uncertain degree of entropy En [42]. Specifically, in (16)-(18), En stands for the search radius, and $He = 0.1En$ is used to represent the stability of the search. In the early iterations, a large En is requested because the flying squirrels' location is often far away from an optimal solution. Under the condition of final generations where the population location is close to an optimal solution, a smaller En is appropriate for the fine-tuning of solutions. Therefore, the search radius En dynamically changes with iteration number:

$$En = En_{max} \times (1 - \text{Iter}/\text{Iter}_{max})^{10} \quad (19)$$

where $En_{max} = (FS_U - FS_L)/4$ is the maximum search radius.

3.3. A Selection Strategy between Successive Positions. When new positions of flying squirrels are generated, it is possible that the new position is worse than the old one. This suggests that the fitness value of each individual needs to be checked after the generation of new positions by comparing with the old one in each iteration. If the fitness value of the new position is better than the old one, the position of the corresponding flying squirrel is updated by the new position. Otherwise, the old position is reserved. This strategy can be mathematically described by

$$FS_i = \begin{cases} FS_i^{new}, & \text{if } f_i^{new} < f_i^{old} \\ FS_i^{old}, & \text{otherwise} \end{cases} \quad (20)$$

```

Set  $Iter_{max}$ ,  $NP$ ,  $n$ ,  $P_{dp}$ ,  $sf$ ,  $G_c$ ,  $FS_U$  and  $FS_L$ 
Randomly initialize the flying squirrels locations
 $FS_{i,j} = FS_L + \text{rand}() * (FS_U - FS_L)$ ,  $i = 1, 2, \dots, NP$ ,  $j = 1, 2, \dots, n$ 
Calculate fitness value
 $f_i = f_i(FS_{i,1}, FS_{i,2}, \dots, FS_{i,n})$ ,  $i = 1, 2, \dots, NP$ 
while  $Iter < Iter_{max}$ 
     $[sorted\_f, sorte\_index] = \text{sort}(f)$ 
     $FS_{nt} = FS(\text{sorte\_index}(1))$ 
     $FS_{at}(1:3) = FS(\text{sorte\_index}(2:4))$ 
     $FS_{nt}(1:NP-4) = FS(\text{sorte\_index}(5:NP))$ 
    Generate new locations
    for  $t = 1: n1$  ( $n1$  = total number of squirrels on acorn trees)
        if  $R_1 \geq P_{dp}$ 
             $FS_{at}^{new} = FS_{at}^{old} + d_g G_c (FS_{nt}^{old} - FS_{at}^{old})$ 
        else
             $FS_{at}^{new} = \text{random location}$ 
        end
    end
    for  $t = 1: n2$  ( $n2$  = total number of squirrels on normal trees moving towards acorn trees)
        if  $R_2 \geq P_{dp}$ 
             $FS_{nt}^{new} = FS_{nt}^{old} + d_g G_c (FS_{at}^{old} - FS_{nt}^{old})$ 
        else
             $FS_{nt}^{new} = \text{random location}$ 
        end
    end
    for  $t = 1: n3$  ( $n3$  = total number of squirrels on normal trees moving towards hickory trees)
        if  $R_3 \geq P_{dp}$ 
             $FS_{nt}^{new} = FS_{nt}^{old} + d_g G_c (FS_{nt}^{old} - FS_{nt}^{old})$ 
        else
             $FS_{nt}^{new} = \text{random location}$ 
        end
    end
    end
     $S_c^t = \sqrt{\sum_{k=1}^n (FS_{at,k}^t - FS_{nt,k}^t)^2}$ ,  $S_{cmin} = \frac{10E-6}{365^{Iter/(Iter_{max})/2.5}}$ 
    if  $S_c^t < S_{cmin}$ 
         $FS_{nt}^{new} = FS_L + \text{Lévy}(n) \times (FS_U - FS_L)$ 
    end
    Calculate fitness value of new locations
     $f_i = f_i(FS_{i,1}^{new}, FS_{i,2}^{new}, \dots, FS_{i,n}^{new})$ ,  $i = 1, 2, \dots, NP$ 
     $Iter = Iter + 1$ 
end

```

ALGORITHM 1: Pseudocode of basic SSA.

3.4. Enhance the Intensive Dimensional Search. In the basic SSA, all dimensions of one individual flying squirrel are updated simultaneously. The main drawback of this procedure is that different dimensions are dependent and the change of one dimension may have negative effects on others, preventing them from finding the optimal variables in their own dimensions. To further enhance the intensive search of each dimension, the following steps are taken for each iteration: (i) find the best flying squirrel location; (ii) generate one more solution based on the best flying squirrel location by changing the value of one dimension while maintaining the rest dimensions; (iii) compare fitness values of the new-generated solution with the original one, and reserve the

better one; (iv) repeat steps (ii) and (iii) in other dimensions individually. The new-generated solution is produced by

$$FS_{best,j}^{new} = Cx(FS_{best,j}^{old}, En, He), \quad j = 1, 2, \dots, n \quad (21)$$

3.5. Procedure of ISSA. The pseudocode of SSA is provided in Algorithm 2.

4. Experimental Results and Analysis

The performance of proposed ISSA is verified and compared with five nature-inspired optimization algorithms, including the basic SSA, PSO [12], fruit fly optimization algorithm


```

Set  $Iter_{max}$ ,  $NP$ ,  $n$ ,  $P_{dpmax}$ ,  $P_{dpmmin}$ ,  $sf$ ,  $G_c$ ,  $FS_U$  and  $FS_L$ 
Randomly initialize the flying squirrels locations
 $FS_{i,j} = FS_L + \text{rand}() * (FS_U - FS_L)$ ,  $i = 1, 2, \dots, NP$ ,  $j = 1, 2, \dots, n$ 
Calculate fitness value
 $f_i = f_i(FS_{i,1}, FS_{i,2}, \dots, FS_{i,n})$ ,  $i = 1, 2, \dots, NP$ 
while  $Iter < Iter_{max}$ 
     $[sorted\_f, sorte\_index] = \text{sort}(f)$ 
     $FS_{ht} = FS(\text{sorte\_index}(1))$ 
     $FS_{at}(1:3) = FS(\text{sorte\_index}(2:4))$ 
     $FS_{nt}(1:NP-4) = FS(\text{sorte\_index}(5:NP))$ 
    Generate new locations
     $P_{dp} = (P_{dpmax} - P_{dpmmin}) \times (1 - \frac{Iter}{Iter_{max}})^{10} + P_{dpmmin}$ 
    for  $t = 1: n1$  ( $n1$  = total number of squirrels on acorn trees)
        if  $R_1 \geq P_{dp}$ 
             $FS_{at}^{new} = FS_{at}^{old} + d_g G_c (FS_{ht}^{old} - FS_{at}^{old})$ 
        else
             $FS_{at}^{new} = Cx(FS_{at}^{old}, En, He)$ 
        end
    end
    for  $t = 1: n2$  ( $n2$  = total number of squirrels on normal trees moving towards acorn trees)
        if  $R_2 \geq P_{dp}$ 
             $FS_{nt}^{new} = FS_{nt}^{old} + d_g G_c (FS_{at}^{old} - FS_{nt}^{old})$ 
        else
             $FS_{nt}^{new} = Cx(FS_{nt}^{old}, En, He)$ 
        end
    end
    for  $t = 1: n3$  ( $n3$  = total number of squirrels on normal trees moving towards hickory trees)
        if  $R_3 \geq P_{dp}$ 
             $FS_{nt}^{new} = FS_{nt}^{old} + d_g G_c (FS_{ht}^{old} - FS_{nt}^{old})$ 
        else
             $FS_{nt}^{new} = Cx(FS_{nt}^{old}, En, He)$ 
        end
    end
    end
     $S_c^t = \sqrt{\sum_{k=1}^n (FS_{at,k}^t - FS_{ht,k}^t)^2}$ ,  $S_{cmin} = \frac{10E-6}{365^{Iter/(Iter_{max})/2.5}}$ 
    if  $S_c^t < S_{cmin}$ 
         $FS_{nt}^{new} = FS_L + \text{Lévy}(n) \times (FS_U - FS_L)$ 
    end
    Calculate fitness value of new locations
     $f_i^{new} = f_i(FS_{i,1}^{new}, FS_{i,2}^{new}, \dots, FS_{i,n}^{new})$ ,  $i = 1, 2, \dots, NP$ 
    if  $f_i^{new} < f_i$ 
         $FS_i = FS_i^{new}$ 
         $f_i = f_i^{new}$ 
    end
    Enhance intensive dimensional search
    Find  $FS_{best}$ ,  $f_{best}$ 
    for  $j = 1:n$ 
         $FS_{best,j}^{new} = Cx(FS_{best,j}, En, He)$ 
    end
    Calculate fitness value of the new solution
     $f_{best}^{new} = f(FS_{best,1}, FS_{best,2}, \dots, FS_{best,j}^{new}, \dots, FS_{best,n})$ 
    if  $f_{best}^{new} < f_{best}$ 
         $FS_{best,j} = FS_{best,j}^{new}$ 
         $f_{best} = f_{best}^{new}$ 
    end
    end
     $Iter = Iter + 1$ 
end

```

ALGORITHM 2: Pseudocode of basic ISSA.

TABLE 1: Parametric settings of algorithms.

Parameter	ISSA	SSA	PSO	CMFOA	IFFO	FOA
$Iter_{max}$	10000	10000	10000	10000	10000	10000
NP	50	50	50	50	50	50
G_c	1.9	1.9	-	-	-	-
sf	18	18	-	-	-	-
P_{dpmax}	0.1	-	-	-	-	-
P_{dpmmin}	0.001	-	-	-	-	-
P_{dp}	-	0.1	-	-	-	-
C_1 and C_2	-	-	2	-	-	-
w	-	-	0.9	-	-	-
En_{max}	-	-	-	$(UB - UL)/4$	-	-
λ_{max}	-	-	-	-	$(UB - UL)/2$	-
λ_{min}	-	-	-	-	0.00001	-
$randValue$	-	-	-	-	-	1

TABLE 2: Unimodal benchmark functions.

Function	Range	Fmin
$F1(x) = \sum_{i=1}^n ix_i^2$	$[-10, 10]$	0
$F2(x) = \sum_{i=2}^n i(2x_i^2 - x_{i-1})^2 + (x_1 - 1)^2$	$[-10, 10]$	0
$F3(x) = -\exp\left(-0.5 \sum_{i=1}^n x_i^2\right)$	$[-1, 1]$	-1
$F4(x) = \sum_{i=1}^n (10^6)^{(i-1)/(n-1)} x_i^2$	$[-100, 100]$	0
$F5(x) = \sum_{i=1}^n ix_i^4 + \text{rand}()$	$[-1.28, 1.28]$	0
$F6(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$	$[-30, 30]$	0
$F7(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j^2 \right)$	$[-100, 100]$	0
$F8(x) = \max\{ x_i , 1 \leq i \leq n\}$	$[-100, 100]$	0
$F9(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-10, 10]$	0
$F10(x) = \sum_{i=1}^n x_i^2$	$[-100, 100]$	0
$F11(x) = \sum_{i=1}^n x_i ^{i+1}$	$[-1, 1]$	0

(FOA) [31], and its two variations: improved fruit fly optimization algorithm (IFFO) [35] and cloud model based fly optimization algorithm (CMFOA) [36]. 32 benchmark functions are tested with a dimension being equal to 30, 50, or 100. These functions are frequently adopted for validating global optimization algorithms, among which F1-F11 are unimodal, F13-F25 belong to multimodal, and F26-F32 are composite functions in the IEEE CEC 2014 special section [43]. Each function is calculated for ten independent runs in order to better compare the results of different algorithms.

Common parameters are set the same for all algorithms, such as population size $NP = 50$; maximal iteration number $Iter_{max} = 10,000$. Meanwhile, the same set of initial random populations is used. The algorithm-specific parameters are chosen the same as those used in the literature that introduces the algorithm at the first time. The parameters of PSO, FOA, IFFO, CMFOA, and SSA are chosen according to [12], [31], [35], [36], and [37], respectively. Table 1 summarizes both common and algorithm-specific parameters for ISSA and other five algorithms. The error value, defined as $(f(x) -$

F_{min}), is recorded for the solution x , where $f(x)$ is the optimal fitness value of the function calculated by the algorithms, and F_{min} is the true minimal value of the function. The average and standard deviation of the error values over all independent runs are calculated.

4.1. Test 1: Unimodal Functions. Unimodal benchmark functions (Table 2) have one global optimum only and they are commonly used for evaluating the exploitation capacity of optimization algorithms. Tables 3–5 list the mean error and standard deviation of the results obtained from each algorithm after ten runs at dimension $n = 30, 50$, and 100 , respectively. The best values are highlighted and marked in italic. It is noted that difficulty in optimization arises with the increase in the dimension of a function because its search space increases exponentially [45]. It is clear from the results that on most of unimodal functions ISSA has better accuracy and convergence precision than other five counterpart algorithms, which confirms that the proposed ISSA has good exploitation ability. As for F2 and F5, ISSA can obtain the same level of accurate mean error as IFFO, while the former outperforms the latter under the condition of $n = 100$. It is also found that both ISSA and CMFOA can achieve the true minimal value of F3 at $n = 30$ and 50 , while ISSA is superior at $n = 100$.

Figures 1–3 show several representative convergence graphs of ISSA and its competitors at $n = 30, 50$, and 100 , respectively. It can be observed that ISSA is able to converge to the true value for most unimodal functions with the fastest convergence speed and highest accuracy, while the convergence results of PSO and FOA are far from satisfactory. The IFFO and CMFOA, with the improvements of search radius though, yield better convergence rates and accuracy in comparison with FOA, but still cannot outperform the proposed ISSA. It is also found that ISSA greatly improves the global convergence ability of SSA mainly because of the introduction of an adaptive strategy of P_{dp} , a selection strategy between successive positions, and enhancement in dimensional search. In addition, the accuracy of all algorithms tends to decrease as the dimension increases, particularly on F6 and F11.

4.2. Test 2: Multimodal Functions. Different from the unimodal functions, multimodal functions have one global optimal solution and multiple local optimal solutions, and the number of local optimal solutions exponentially increases with the increase of dimension. This feature makes them suitable for testing the exploration ability of an algorithm. Details of these multimodal functions are listed in Table 6. The recorded results of statistical analysis over 10 independent runs are presented in Tables 7–9 for $n = 30, 50$, and 100 , respectively. It is revealed from these tables that ISSA is superior on F12, F13, F14, F16, F19, and F24, regardless of dimension number. On other functions, ISSA tends to have comparable level of accuracy with some of its competitors. For example, both ISSA and CMFOA are able to obtain the exact optimal solution of F21 and F22; both ISSA and SSA have the same level of accuracy on F15, F18, and F23. It is noticeable that ISSA tends to get

better performance in accuracy on more functions as the dimension number increases. This is mainly contributed by normal cloud model based flying squirrels' random position generation and dimensionally enhanced search. These two strategies can help the flying squirrels to escape from local optimal.

Figures 4–6 show the recorded convergence characteristics of algorithms for several multimodal benchmark functions at $n = 30, 50$, and 100 , respectively. It is evident that ISSA offers better global convergence rate and precision in comparison with other five algorithms among which both PSO and FOA are easy to be trapped to the local optimal and the rest three algorithms (IFFO, CMFOA, and SSA) produce fair convergence rates. It is interesting to note that SSA becomes much poorer as the dimension number increases, while ISSA still has excellent exploration ability and its convergence curve ranks No. 1 at all iterations in the case of $n = 100$. This is due to the incorporation of attributes regarding normal cloud model generators and search enhancement on each dimension.

4.3. Test 3: CEC 2014 Benchmark Functions. Next, the benchmark functions used in IEEE CEC 2014 are considered for investigating the balance between exploration and exploitation of optimization algorithms. These functions include several novel basic problems (e.g., with shifting and rotation) and hybrid and composite test problems. In the present test, seven CEC 2014 functions are selected with at least one function in each group and the details are provided in Table 10. Statistical results obtained by different algorithms through 10 independent runs are recorded in Tables 11–13. It is worth mentioning that CEC 2014 functions are specifically designed to have complicated features, and thus it is difficult to reach the global optimal for all algorithms under consideration. Nevertheless, in contrast to other five algorithms, ISSA is able to get highly competitive results for most CEC 2014 functions in Table 10, especially at higher dimension number. As a matter of fact, ISSA always has the best solution at $n = 100$ although the solution is still far away from optimal. The results of convergence studies (Figures 7–9) show that ISSA has promising convergence performance with the comparison of other five algorithms. The superior performance of the proposed ISSA is mainly benefited from an equilibrium between global and local search abilities because of the use of the four strategies described in Section 3.

4.4. Statistical Analysis. In order to analyze the performance of any two algorithms, the most frequently used nonparametric statistical test, Wilcoxon's test [46], is considered for the present work and results are summarized in Tables 14–16 for $n = 30, 50$, and 100 , respectively. The test is carried out by considering the best solution of each algorithm on each benchmark function with 10 independent runs and a significance level of $\alpha = 0.05$. In Tables 14–16, '+' sign indicates that the reference algorithm outperforms the compared one,

TABLE 3: Statistical results obtained by ISSA, SSA, PSO, CMFOA, IFFO, and FOA through 10 independent runs on unimodal benchmark functions with $n = 30$.

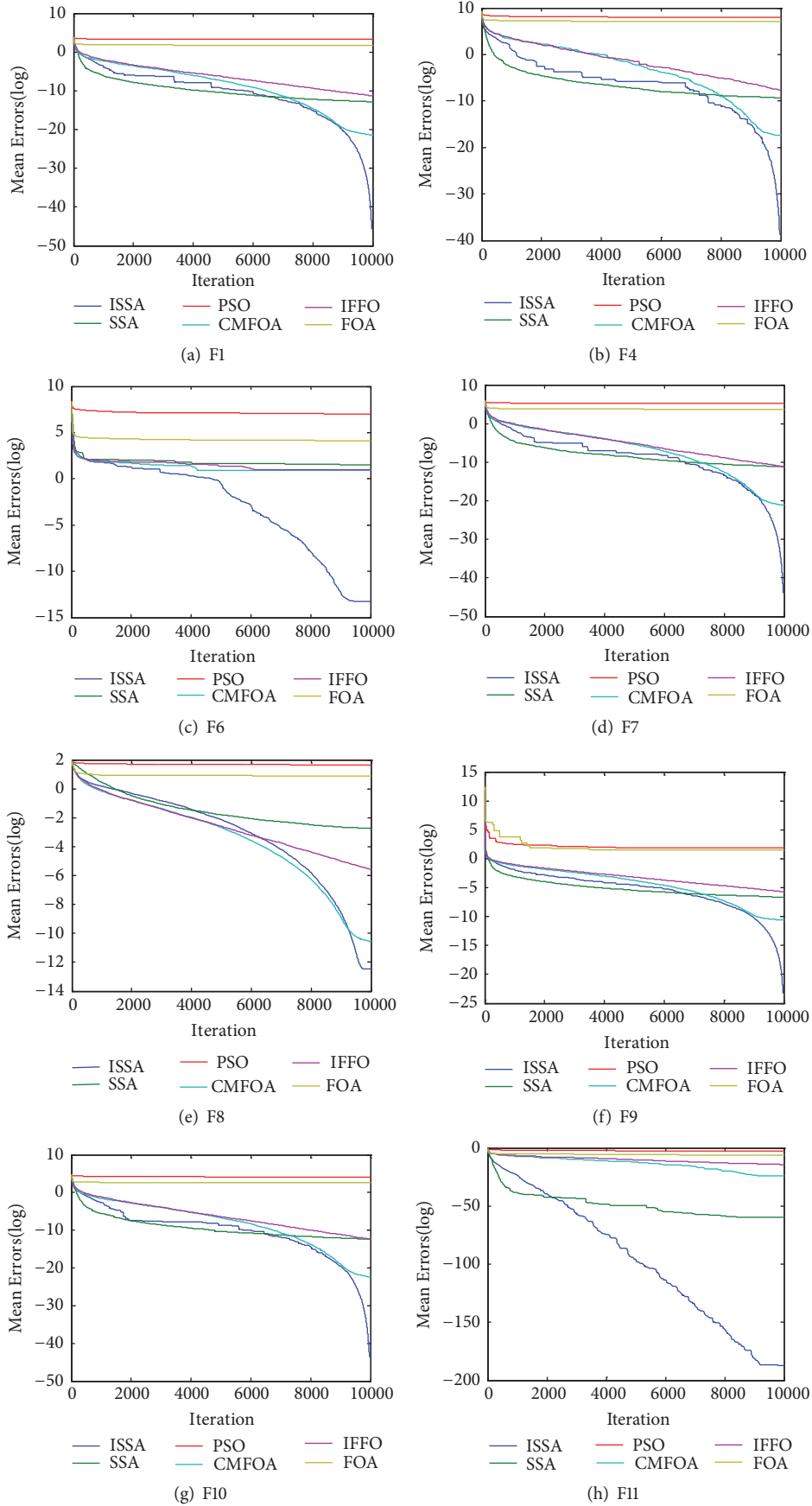
Fun		ISSA	SSA	PSO	CMFOA	IFFO	FOA
F1	Mean	2.2612E-46	1.4374E-13	1.8031E+03	3.8546E-22	5.3419E-12	5.8887E+01
	Std	3.9697E-46	1.0187E-13	1.6038E+02	1.1464E-22	2.6506E-12	1.0717E+01
F2	Mean	5.3333E-01	6.0000E-01	7.0475E+04	6.6667E-01	6.6667E-01	1.2221E+02
	Std	2.8109E-01	2.1082E-01	1.8027E+04	1.1102E-16	3.6414E-11	3.5452E+01
F3	Mean	0.0000E+00	0.0000E+00	4.7300E-01	0.0000E+00	8.6042E-14	1.8586E-02
	Std	1.8504E-16	2.6168E-16	4.2225E-02	9.0649E-17	2.7669E-14	1.9010E-03
F4	Mean	2.1268E-39	3.9943E-10	9.2617E+07	3.7704E-18	2.0345E-08	1.1112E+07
	Std	5.6486E-39	3.2855E-10	1.8784E+07	2.4165E-18	1.4277E-08	3.0272E+06
F5	Mean	4.3164E-03	9.3054E-02	5.5376E+00	3.2231E-03	2.2754E-03	1.7965E-02
	Std	1.9931E-03	2.3058E-02	1.0210E+00	1.3321E-03	1.1046E-03	3.1904E-03
F6	Mean	5.5447E-14	2.9695E+01	1.0669E+07	7.6347E+00	8.9052E+00	1.2862E+04
	Std	5.4894E-14	3.4074E+01	1.8313E+06	5.9003E+00	7.1150E+00	4.4338E+03
F7	Mean	1.1996E-44	6.5616E-12	1.6688E+05	7.1055E-22	6.0744E-12	5.4708E+03
	Std	3.0449E-44	5.4085E-12	1.7265E+04	1.6506E-22	2.9515E-12	4.9070E+02
F8	Mean	3.5080E-13	1.7850E-03	4.5639E+01	2.7020E-11	2.6440E-06	7.8561E+00
	Std	7.0894E-13	4.3281E-04	2.0578E+00	6.0413E-12	2.4822E-07	1.7334E+00
F9	Mean	5.5772E-24	2.2148E-07	7.1792E+01	2.3880E-11	2.0415E-06	3.0453E+01
	Std	7.9227E-24	6.7302E-08	3.0539E+01	4.1360E-12	3.6636E-07	4.6515E+01
F10	Mean	2.6748E-44	4.4745E-13	1.3098E+04	4.2105E-23	4.4042E-13	3.6501E+02
	Std	7.8461E-44	3.7055E-13	1.3116E+03	1.0825E-23	1.5887E-13	4.3608E+01
F11	Mean	6.1803E-188	1.7833E-60	1.8391E-03	7.8265E-25	5.1176E-15	6.7271E-07
	Std	0.0000E+00	3.7202E-60	1.1147E-03	6.5934E-25	7.6076E-15	4.6836E-07

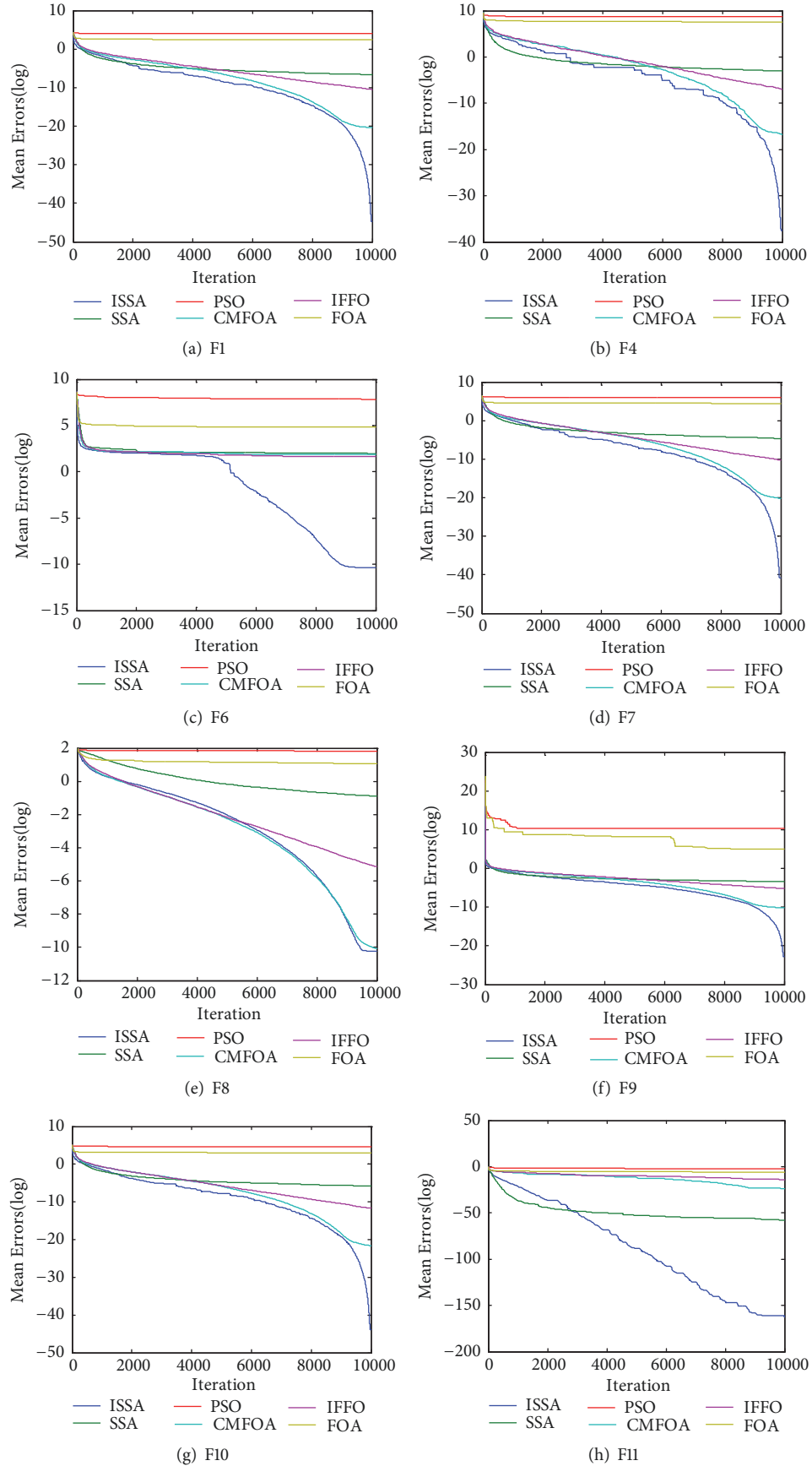
TABLE 4: Statistical results obtained by ISSA, SSA, PSO, CMFOA, IFFO, and FOA through 10 independent runs on unimodal benchmark functions with $n = 50$.

Fun		ISSA	SSA	PSO	CMFOA	IFFO	FOA
F1	Mean	1.9373E-45	2.4408E-07	8.9004E+03	3.6571E-21	3.2632E-11	2.8288E+02
	Std	4.0611E-45	7.2161E-08	1.0186E+03	9.0310E-22	8.2802E-12	1.8277E+01
F2	Mean	6.6667E-01	2.1716E+00	8.0535E+05	1.2514E+00	6.6667E-01	8.2661E+02
	Std	8.2003E-16	2.2142E+00	1.1670E+05	1.2485E+00	2.5423E-10	7.7814E+01
F3	Mean	0.0000E+00	7.6318E-11	8.6602E-01	0.0000E+00	4.1100E-13	5.1246E-02
	Std	2.2204E-16	2.2120E-11	1.8360E-02	2.3984E-16	1.4800E-13	4.0430E-03
F4	Mean	3.0671E-38	9.7033E-04	4.6756E+08	1.9432E-17	1.0621E-07	3.8893E+07
	Std	6.9186E-38	3.4464E-04	6.0135E+07	1.1627E-17	7.6083E-08	7.3301E+06
F5	Mean	7.1557E-03	2.9566E-01	5.3164E+01	1.0084E-02	7.3580E-03	1.0458E-01
	Std	2.3021E-03	3.3200E-02	6.4915E+00	2.6523E-03	1.8100E-03	2.6586E-02
F6	Mean	4.3706E-11	9.5471E+01	7.0118E+07	7.7331E+01	4.7194E+01	6.5331E+04
	Std	9.5151E-11	3.5358E+01	5.3302E+06	3.4463E+01	4.0976E+01	1.3721E+04
F7	Mean	1.2947E-41	2.3079E-05	9.1659E+05	6.9771E-21	6.4025E-11	2.8862E+04
	Std	3.7876E-41	5.8814E-06	9.3287E+04	3.1808E-21	2.6901E-11	2.8375E+03
F8	Mean	6.0872E-11	1.2706E-01	6.7093E+01	8.4930E-11	7.1576E-06	1.1919E+01
	Std	2.5158E-11	3.3391E-02	2.5011E+00	1.1107E-11	6.9032E-07	1.0404E+00
F9	Mean	1.8289E-23	3.8822E-04	2.0565E+10	6.0338E-11	6.3442E-06	1.1434E+05
	Std	2.8884E-23	7.2525E-05	6.3864E+10	6.4165E-12	9.0797E-07	2.4588E+05
F10	Mean	1.2924E-44	1.4594E-06	4.1629E+04	2.2846E-22	2.0175E-12	1.0536E+03
	Std	2.5807E-44	6.0662E-07	3.7125E+03	4.1424E-23	5.2761E-13	5.9785E+01
F11	Mean	4.4745E-163	1.9208E-58	9.2852E-03	1.1169E-24	5.1458E-15	1.6917E-06
	Std	0.0000E+00	2.2612E-58	3.5776E-03	1.4674E-24	8.2130E-15	9.4517E-07

TABLE 5: Statistical results obtained by ISSA, SSA, PSO, CMFOA, IFFO, and FOA through 10 independent runs on unimodal benchmark functions with $n = 100$.

Fun		ISSA	SSA	PSO	CMFOA	IFFO	FOA
F1	Mean	5.2002E-44	1.3601E-01	6.4559E+04	6.7630E-20	5.4042E-10	2.3688E+03
	Std	8.9565E-44	2.8398E-02	2.7675E+03	1.8636E-20	1.0843E-10	1.1782E+02
F2	Mean	2.5837E+00	2.8507E+01	8.9058E+06	1.0018E+01	1.1850E+01	1.3921E+04
	Std	4.0923E+00	1.2482E+01	6.4125E+05	9.8260E+00	6.7378E+00	2.1479E+03
F3	Mean	1.9984E-15	1.7506E-05	9.9937E-01	1.9984E-15	3.4570E-12	1.9460E-01
	Std	2.7940E-16	3.3607E-06	1.9874E-04	3.9686E-16	5.7323E-13	1.1259E-02
F4	Mean	1.4073E-38	3.0139E+02	2.6151E+09	1.4261E-16	1.0212E-06	2.0499E+08
	Std	1.5382E-38	9.0551E+01	2.6783E+08	7.5737E-17	3.3853E-07	3.5388E+07
F5	Mean	1.7615E-02	1.4345E+00	5.9603E+02	3.7550E-02	2.9349E-02	1.2443E+00
	Std	4.2239E-03	1.3985E-01	5.8412E+01	1.2602E-02	4.5825E-03	1.8471E-01
F6	Mean	1.1417E+01	5.8422E+02	4.2299E+08	1.7988E+02	1.6578E+02	5.6078E+05
	Std	3.0258E+01	9.7884E+02	4.8581E+07	3.9022E+01	4.6685E+01	6.5477E+04
F7	Mean	1.6881E-41	1.2984E+01	6.4707E+06	1.1852E-19	7.4831E-10	2.2865E+05
	Std	3.4134E-41	2.2729E+00	3.3435E+05	3.1718E-20	9.5033E-11	2.0650E+04
F8	Mean	4.5259E-08	3.9819E+00	8.5137E+01	1.7042E-04	2.9244E-05	3.3956E+01
	Std	2.9104E-08	4.1522E-01	1.2566E+00	7.8665E-05	2.7018E-06	4.7713E+00
F9	Mean	1.2222E-22	3.6814E-01	7.2469E+32	2.5070E-10	2.3795E-05	1.4112E+27
	Std	8.4369E-23	4.1963E-02	2.0633E+33	2.3874E-11	1.3879E-06	4.4627E+27
F10	Mean	3.4254E-42	3.7261E-01	1.4940E+05	2.0714E-21	1.8486E-11	4.3041E+03
	Std	9.6608E-42	9.2561E-02	4.4643E+03	4.6407E-22	2.3956E-12	2.4348E+02
F11	Mean	1.6400E-123	1.5040E-52	3.7278E-02	6.5780E-24	3.1174E-14	1.0720E-05
	Std	5.1861E-123	1.6670E-52	1.1165E-02	7.2960E-24	3.6245E-14	5.9475E-06

FIGURE 1: Convergence rate comparison for representative unimodal functions ($n = 30$).

FIGURE 2: Convergence rate comparison for representative unimodal functions ($n = 50$).

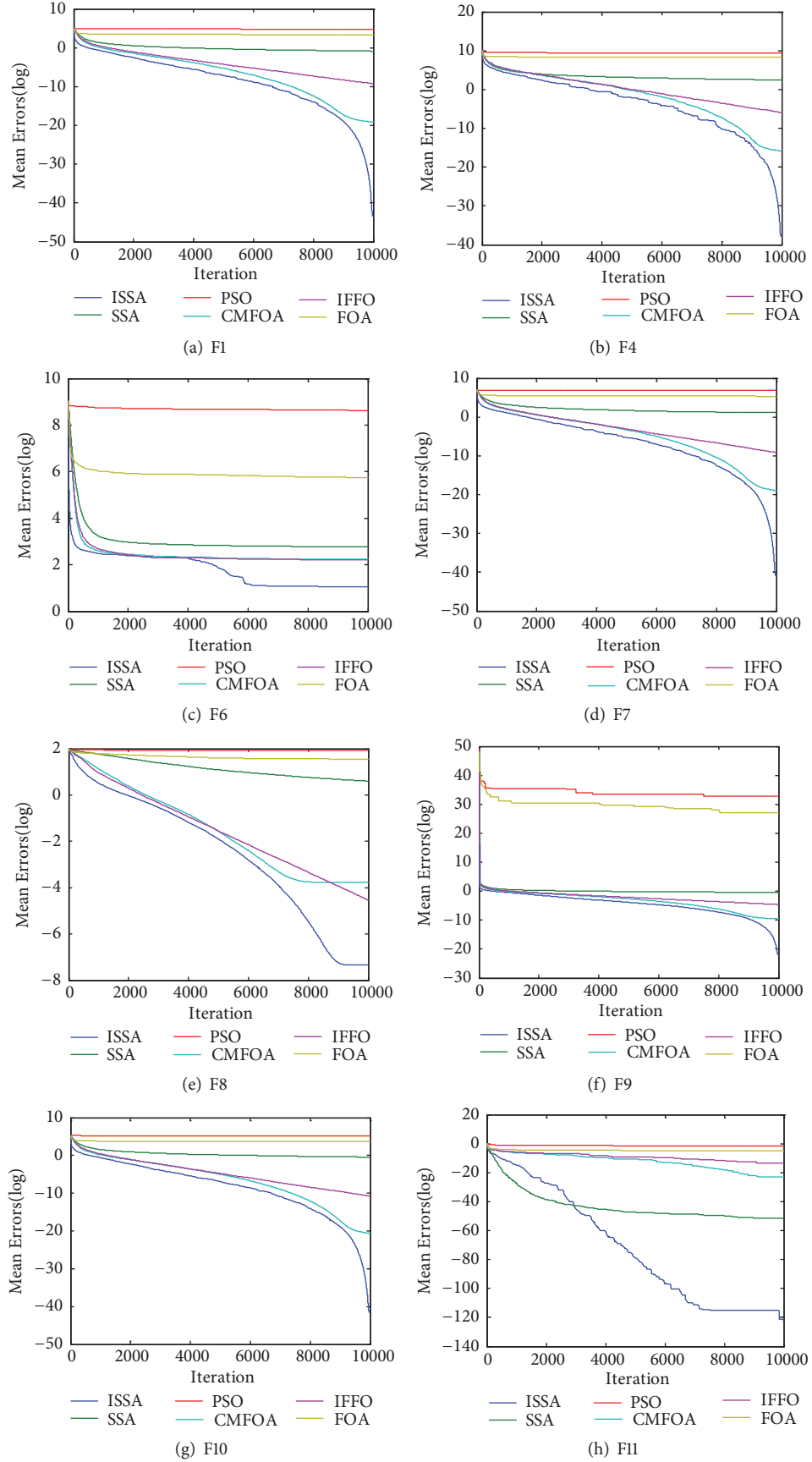
FIGURE 3: Convergence rate comparison for representative unimodal functions ($n = 100$).

TABLE 6: Multimodal benchmark functions.

Function	Range	Fmin
$F12(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + \exp(1)$	$[-32, 32]$	0
$F13(x) = \sum_{i=1}^n x_i \sin(x_i) + 0.1 x_i $	$[-10, 10]$	0
$F14(x) = f_s(x_1, x_2) + \dots + f_s(x_n, x_1)$, $f_s(x, y) = (x^2 + y^2)^{0.25} \left[\sin^2(50(x^2 + y^2)^{0.1}) + 1 \right]$	$[-100, 100]$	0
$F15(x) = f_s(x_1, x_2) + \dots + f_s(x_n, x_1)$, $f_s(x, y) = \frac{0.5 \left(\sin^2(\sqrt{x^2 + y^2}) - 0.5 \right)}{(1 + 0.001(x^2 + y^2))^2}$	$[-100, 100]$	0
$F16(x) = \frac{\pi}{n} \left\{ 10 \sin^2(\pi y_i) + \sum_{i=1}^{n-1} (y_i - 1)^2 \left[1 + 10 \sin^2(\pi y_{i+1}) \right] + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$, $y_i = 1 + \frac{1}{4}(x_i + 1)$, $u(x_i, a, k, m) = \begin{cases} 0, & -a \leq x_i \leq a \\ k(x_i - a)^m, & x_i > a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	$[-50, 50]$	0
$F17(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-100, 100]$	0
$F18(x) = -\sum_{i=1}^{n-1} \left(\exp\left(-\frac{(x_i^2 + x_{i+1}^2 + 0.5x_i x_{i+1})}{8}\right) * \cos\left(4\sqrt{x_i^2 + x_{i+1}^2 + 0.5x_i x_{i+1}}\right) \right)$	$[-5, 5]$	1-n
$F19(x) = \sum_{i=1}^n (x_i - 1)^2 - \sum_{i=2}^n x_i x_{i-1}$	$[-n^2, n^2]$	$\frac{n(n+4)(n-1)}{-6}$
$F20(x) = \frac{\sum_{i=2}^{n-1} (0.5 + (\sin^2(\sqrt{100x_i^2 + x_{i+1}^2}) - 0.5))}{(1 + 0.001(x_i^2 - 2x_i x_{i-1} + x_{i-1}^2))^2}$	$[-100, 100]$	0
$F21(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]$	0
$F22(x) = \sum_{i=1}^n [y_i^2 - 10 \cos(2\pi y_i) + 10]$, $y_i = \begin{cases} x_i, & x_i < 0.5 \\ \frac{\text{round}(2x_i)}{2}, & x_i \geq 0.5 \end{cases}$	$[-5.12, 5.12]$	0
$F23(x) = 1 - \cos\left(2\pi \sqrt{\sum_{i=1}^n x_i^2}\right) + 0.1 \sqrt{\sum_{i=1}^n x_i^2}$	$[-100, 100]$	0
$F24(x) = \sum_{i=1}^n \left\{ \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (x_i + 0.5))] \right\} - n \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k 0.5)]$	$[-0.5, 0.5]$	0
$F25(x) = \sum_{k=1}^n \sum_{j=1}^n \left(\left(\frac{y_{jk}^2}{4000} \right) - \cos(y_{jk}) + 1 \right)$, $y_{jk} = 100(x_k - x_j)^2 + (1 - x_j^2)^2$	$[-100, 100]$	0

TABLE 7: Statistical results obtained by ISSA, SSA, PSO, CMFOA, IFFO, and FOA through 10 independent runs on multimodal benchmark functions with $n = 30$.

Fun		ISSA	SSA	PSO	CMFOA	IFFO	FOA
F12	Mean	5.1692E-14	2.1708E-07	1.6343E+01	4.2641E-12	4.3970E-07	7.0983E+00
	Std	9.4813E-15	1.1785E-07	4.5830E-01	5.1275E-13	5.3024E-08	4.5755E+00
F13	Mean	1.9651E-15	1.7670E-07	3.0865E+01	3.8781E-12	1.2507E-06	2.9660E+01
	Std	1.7016E-15	1.0899E-07	2.8749E+00	2.0014E-12	1.9125E-06	5.7790E+00
F14	Mean	2.8586E-11	4.7414E-02	2.1576E+02	3.5954E-05	1.7290E-02	1.8705E+02
	Std	1.7874E-11	1.8105E-02	5.0836E+00	1.9343E-06	1.0857E-03	5.0868E+01
F15	Mean	9.9552E-01	4.6150E-01	1.2596E+01	9.4983E-01	1.0032E+00	1.2147E+01
	Std	3.8926E-01	3.3522E-01	2.1495E-01	4.2966E-01	3.5690E-01	1.7388E-01
F16	Mean	1.5705E-32	1.3069E-15	5.6725E+06	5.0290E-25	9.9726E-15	3.1482E+00
	Std	2.8850E-48	5.7169E-16	1.7168E+06	4.7027E-25	8.5374E-15	5.8054E-01
F17	Mean	1.3781E-02	1.0332E-02	4.3352E+00	4.4332E-03	1.2793E-02	1.0971E+00
	Std	1.4865E-02	1.2632E-02	4.2518E-01	7.9408E-03	1.0155E-02	1.0766E-02
F18	Mean	5.0849E+00	3.8253E+00	2.0946E+01	4.9225E+00	4.9048E+00	2.1497E+01
	Std	1.6014E+00	1.4627E+00	7.6856E-01	2.1737E+00	2.0411E+00	1.3669E+00
F19	Mean	2.6841E-07	1.9292E+02	4.9808E+05	1.9677E+02	2.4098E+02	3.0226E+04
	Std	3.2619E-08	1.5971E+02	1.4706E+05	1.6572E+02	2.3149E+02	6.0289E+03
F20	Mean	2.5989E-07	4.7006E-06	3.3592E-02	4.4469E-08	1.8865E-07	1.5406E-01
	Std	5.9383E-07	7.3387E-06	2.2456E-02	1.0350E-07	3.1612E-07	5.6719E-02
F21	Mean	0.0000E+00	7.0841E-13	2.5769E+02	0.0000E+00	4.5409E-11	3.0881E+02
	Std	0.0000E+00	4.5361E-13	9.0973E+00	0.0000E+00	1.9882E-11	2.7305E+01
F22	Mean	0.0000E+00	7.7467E-13	2.3335E+02	0.0000E+00	6.4403E-11	2.5509E+02
	Std	0.0000E+00	3.6979E-13	1.5942E+01	0.0000E+00	3.3820E-11	2.6992E+01
F23	Mean	9.3987E-01	5.2987E-01	1.2199E+01	1.3599E+00	1.4399E+00	2.1878E+00
	Std	2.1705E-01	1.2517E-01	4.9304E-01	3.6576E-01	2.1705E-01	6.2731E-02
F24	Mean	1.4921E-14	3.7233E-04	3.2412E+01	4.7458E-09	4.2553E-03	2.6924E+01
	Std	1.7226E-14	9.8846E-05	1.1649E+00	2.8242E-09	4.2975E-04	3.5559E+00
F25	Mean	2.9494E+01	1.0724E+02	1.1372E+07	4.0462E+01	9.3530E+00	9.2421E+03
	Std	2.9743E+01	5.1800E+01	3.1606E+06	3.9685E+01	9.0392E+00	1.8838E+03

TABLE 8: Statistical results obtained by ISSA, SSA, PSO, CMFOA, IFFO, and FOA through 10 independent runs on multi-modal benchmark functions with $n = 50$.

Fun		ISSA	SSA	PSO	CMFOA	IFFO	FOA
F12	Mean	8.5798E-14	2.4174E-04	1.8459E+01	7.4044E-12	7.3673E-07	8.2226E+00
	Std	1.7360E-14	5.5274E-05	1.9447E-01	8.8139E-13	8.0222E-08	4.2517E+00
F13	Mean	2.2538E-15	2.9492E-04	7.1594E+01	2.1041E-11	3.2004E-06	6.0959E+01
	Std	1.8688E-15	1.0372E-04	4.5394E+00	1.5865E-11	1.5334E-06	4.4766E+00
F14	Mean	7.1759E-11	2.0261E+00	4.3430E+02	7.7682E-05	3.3324E-02	4.2669E+02
	Std	2.4650E-11	5.0770E-01	1.4055E+01	5.4975E-06	1.0537E-03	8.0127E+01
F15	Mean	1.6716E+00	1.2749E+00	2.2241E+01	1.6927E+00	1.4937E+00	2.1617E+01
	Std	7.6572E-01	4.3985E-01	3.3014E-01	4.7677E-01	6.3574E-01	5.4534E-01
F16	Mean	9.4233E-33	1.3057E-09	7.6995E+07	1.7755E-24	8.4648E-14	6.9921E+00
	Std	1.4425E-48	3.7533E-10	2.1712E+07	1.9092E-24	1.7429E-13	8.9129E-01
F17	Mean	7.6377E-03	1.4219E-02	1.1606E+01	6.4039E-03	1.0080E-02	1.2641E+00
	Std	5.7418E-03	2.1089E-02	4.6282E-01	7.0807E-03	1.3952E-02	1.6555E-02
F18	Mean	8.3103E+00	7.9047E+00	3.9689E+01	8.9467E+00	9.6041E+00	3.8726E+01
	Std	2.6072E+00	2.5432E+00	7.7616E-01	7.8506E-01	2.1029E+00	1.3015E+00
F19	Mean	4.5562E+01	2.6833E+04	8.0668E+06	1.6118E+04	1.3155E+04	7.0015E+05
	Std	3.8094E+01	2.1743E+04	2.1709E+06	1.2498E+04	1.3009E+04	9.7174E+04
F20	Mean	4.3064E-08	2.5702E-04	1.1519E-01	5.2365E-08	1.6998E-06	5.0047E-01
	Std	4.4294E-08	2.7576E-04	3.9417E-02	9.5247E-08	4.9881E-06	2.6305E-01
F21	Mean	0.0000E+00	1.1310E-06	5.3146E+02	0.0000E+00	2.3711E-10	5.8748E+02
	Std	0.0000E+00	3.3614E-07	3.2117E+01	0.0000E+00	4.5437E-11	2.9507E+01
F22	Mean	0.0000E+00	1.6167E-06	4.8729E+02	0.0000E+00	2.4407E-10	5.2060E+02
	Std	0.0000E+00	6.3216E-07	2.4382E+01	0.0000E+00	7.5889E-11	4.2230E+01
F23	Mean	1.3699E+00	8.9987E-01	2.1237E+01	2.2699E+00	2.5899E+00	3.5955E+00
	Std	2.3594E-01	6.6667E-02	5.8033E-01	4.1913E-01	6.2973E-01	1.2247E-01
F24	Mean	7.1054E-14	2.6826E-02	6.3090E+01	1.9033E-08	9.6037E-03	4.7263E+01
	Std	2.7621E-14	4.7780E-03	2.2392E+00	6.1075E-09	9.7071E-04	5.2689E+00
F25	Mean	6.6563E+01	8.4722E+02	1.1275E+08	6.5780E+01	3.9992E+01	8.8242E+04
	Std	1.0992E+02	2.1138E+02	2.1091E+07	9.4954E+01	4.3819E+01	1.6832E+04

TABLE 9: Statistical results obtained by ISSA, SSA, PSO, CMFOA, IFFO, and FOA through 10 independent runs on multi-modal benchmark functions with $n = 100$.

Fun		ISSA	SSA	PSO	CMFOA	IFFO	FOA
F12	Mean	1.8989E-13	1.6584E-01	1.9996E+01	1.7809E-11	1.4744E-06	1.3554E+01
	Std	2.0566E-14	5.3720E-02	9.0319E-02	1.9159E-12	1.8930E-07	6.0821E+00
F13	Mean	2.2871E-15	1.7736E-01	1.9447E+02	1.3452E-10	1.3291E-05	1.6379E+02
	Std	2.6741E-15	5.3611E-02	6.2653E+00	3.8592E-11	5.5001E-06	1.3313E+01
F14	Mean	1.8736E-10	7.4259E+01	1.0132E+03	2.2866E-04	8.3534E-02	9.5534E+02
	Std	3.7223E-11	1.9144E+01	1.8986E+01	1.4283E-05	1.0592E-02	5.3523E+01
F15	Mean	2.6814E+00	1.0178E+01	4.7083E+01	2.8083E+00	3.4325E+00	4.5859E+01
	Std	7.3851E-01	1.6238E+00	2.2513E-01	4.6148E-01	6.0283E-01	6.9914E-01
F16	Mean	4.7116E-33	2.4454E-04	9.0382E+08	8.1890E-24	6.2347E-14	2.7647E+03
	Std	7.2124E-49	5.9650E-05	6.4985E+07	6.7958E-24	5.5604E-14	4.4231E+03
F17	Mean	3.4494E-03	1.1896E-02	3.7816E+01	3.4509E-03	4.1885E-03	2.1280E+00
	Std	6.0565E-03	6.5363E-03	1.5922E+00	4.6765E-03	8.6153E-03	5.4359E-02
F18	Mean	1.8033E+01	1.7806E+01	8.6826E+01	1.8319E+01	1.8828E+01	8.2458E+01
	Std	1.9652E+00	3.8319E+00	9.3222E-01	2.9296E+00	2.5377E+00	1.5159E+00
F19	Mean	8.2462E+04	2.7944E+06	4.8046E+08	2.8415E+05	6.0265E+05	4.9201E+07
	Std	5.5732E+04	8.9703E+05	9.6715E+07	2.4572E+05	2.7137E+05	7.2772E+06
F20	Mean	5.7130E-07	8.1688E-03	9.6848E-01	1.3631E-06	2.7143E-05	2.1656E+00
	Std	6.1122E-07	5.3195E-03	4.4542E-01	2.5155E-06	5.8766E-05	8.0368E-01
F21	Mean	0.0000E+00	5.1414E+00	1.3305E+03	0.0000E+00	2.0026E-09	1.3623E+03
	Std	0.0000E+00	1.7825E+00	2.2890E+01	0.0000E+00	3.2815E-10	6.0996E+01
F22	Mean	0.0000E+00	7.7848E+00	1.2609E+03	0.0000E+00	2.0383E-09	1.2745E+03
	Std	0.0000E+00	2.3732E+00	2.9100E+01	0.0000E+00	2.9753E-10	5.9708E+01
F23	Mean	2.5599E+00	2.0499E+00	3.9804E+01	4.7099E+00	4.3699E+00	7.3691E+00
	Std	3.6878E-01	1.5092E-01	6.9296E-01	5.9151E-01	5.6184E-01	1.7989E-01
F24	Mean	4.0927E-13	2.6229E+00	1.5145E+02	1.8874E-07	2.9476E-02	1.0478E+02
	Std	8.8061E-14	6.3367E-01	4.2830E+00	3.7074E-08	1.7697E-03	1.1873E-01
F25	Mean	4.2987E+02	8.1178E+03	1.3524E+09	5.6790E+02	4.4982E+02	1.8038E+06
	Std	4.3423E+02	3.3128E+02	7.8399E+07	5.4327E+02	4.6926E+02	2.1315E+05

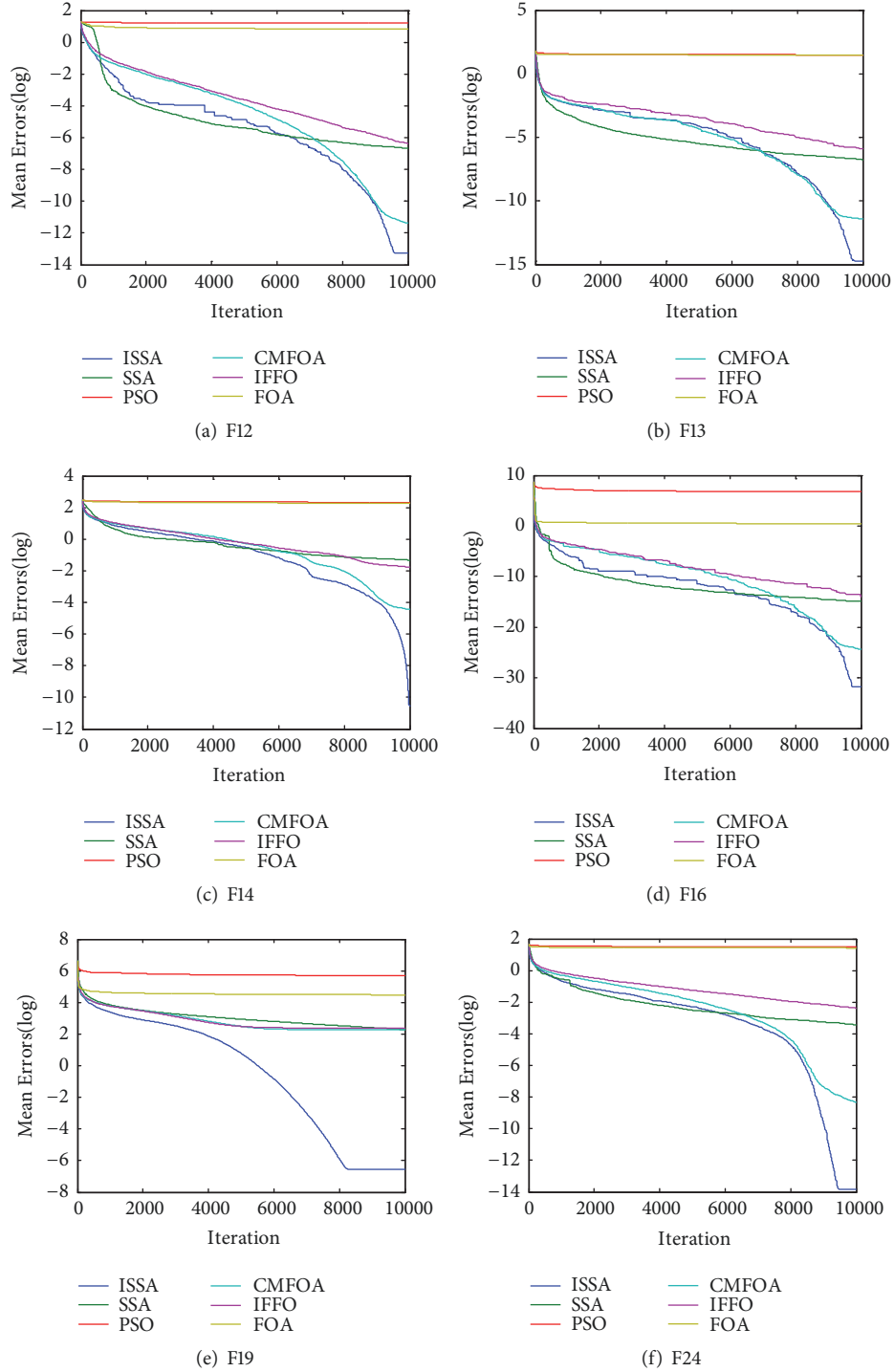


FIGURE 4: Convergence rate comparison for representative multimodal functions ($n = 30$).

‘-’ sign indicates that the reference algorithm is inferior to the compared one, and ‘=’ sign indicates that both algorithms have comparable performances. The results of last row of the tables show that the proposed ISSA has a larger number of ‘+’ counts in comparison to other algorithms, confirming that ISSA is better than the other five compared algorithms under 95% level of significance.

The quantitative analysis is also carried out for six algorithms with an index of mean absolute error (MAE) which is an effective performance index for ranking the optimization algorithms and is defined by [47]

$$\text{MAE} = \frac{\sum_{j=1}^N |m_j - k_j|}{N} \quad (22)$$

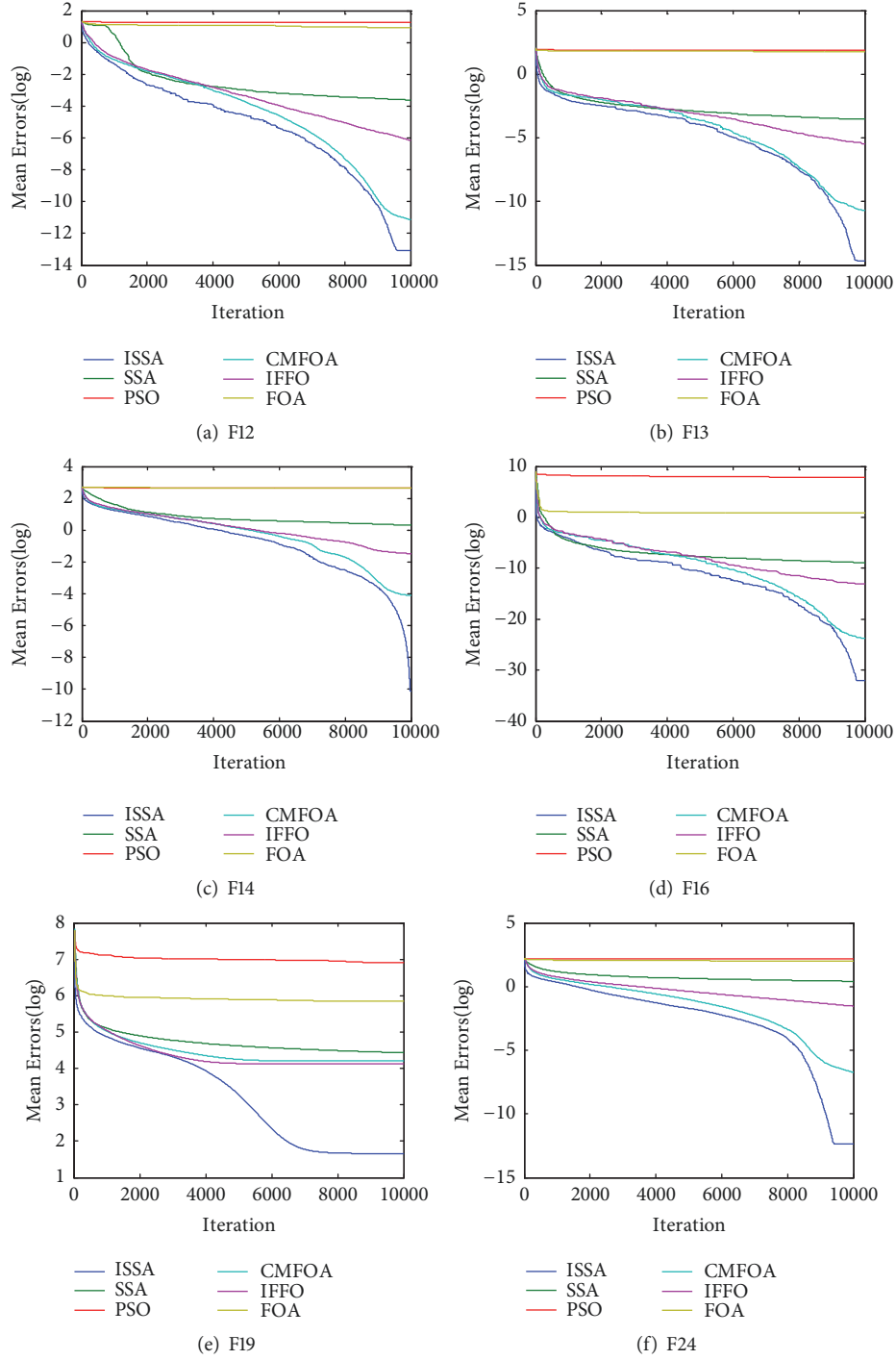
FIGURE 5: Convergence rate comparison for representative multimodal functions ($n = 50$).

TABLE 10: CEC 2014 benchmark functions.

Function	Range	Fmin
F26 (CEC1: Rotated High Conditioned Elliptic Function)	$[-100, 100]$	100
F27 (CEC2: Rotated Bent Cigar Function)	$[-100, 100]$	200
F28 (CEC4: Shifted and Rotated Rosenbrock's Function)	$[-100, 100]$	400
F29 (CEC17: Hybrid Function 1)	$[-100, 100]$	1700
F30 (CEC23: Composition Function 1)	$[-100, 100]$	2300
F31 (CEC24: Composition Function 2)	$[-100, 100]$	2400
F32 (CEC25: Composition Function 3)	$[-100, 100]$	2500

TABLE II: Statistical results obtained by ISSA, SSA, PSO, CMFOA, IFFO, and FOA through 10 independent runs on CEC 2014 benchmark functions with $n = 30$.

Fun		ISSA	SSA	PSO	CMFOA	IFFO	FOA
F26	Mean	3.1365E+04	1.3133E+06	1.1295E+08	1.0017E+06	8.6475E+05	1.3449E+07
	Std	1.8602E+04	5.2974E+05	3.1387E+07	4.8689E+05	4.8607E+05	2.8405E+06
F27	Mean	3.0400E-10	1.8446E+04	8.4500E+09	1.0512E+04	1.2359E+04	5.8535E+08
	Std	6.1535E-10	1.4049E+04	1.0125E+09	1.2485E+04	1.1922E+04	3.8771E+07
F28	Mean	4.2105E-01	4.6710E+01	7.3819E+02	4.1147E+01	3.7814E+01	1.4226E+02
	Std	1.2624E+00	3.1490E+01	9.9455E+01	4.7336E+01	3.4110E+01	2.9201E+01
F29	Mean	7.5177E+03	1.4891E+05	2.9286E+06	4.7277E+05	3.1099E+05	3.9826E+05
	Std	3.3119E+03	6.8316E+04	9.1904E+05	2.1021E+05	2.2686E+05	1.5511E+05
F30	Mean	3.1524E+02	3.1524E+02	3.8129E+02	3.1524E+02	3.1524E+02	3.2568E+02
	Std	8.5708E-12	1.9710E-07	1.4082E+01	1.1524E-11	4.5680E-11	5.8955E+00
F31	Mean	2.3483E+02	2.3172E+02	3.0117E+02	2.3811E+02	2.3858E+02	2.4179E+02
	Std	4.1748E+00	7.2461E+00	4.8903E+00	5.6097E+00	5.0249E+00	9.0228E+00
F32	Mean	2.0790E+02	2.0603E+02	2.1884E+02	2.1485E+02	2.0975E+02	2.0633E+02
	Std	4.1618E+00	3.2456E+00	3.0353E+00	8.7909E+00	5.7719E+00	1.6880E+00

TABLE 12: Statistical results obtained by ISSA, SSA, PSO, CMFOA, IFFO, and FOA through 10 independent runs on CEC 2014 benchmark functions with $n = 50$.

Fun		ISSA	SSA	PSO	CMFOA	IFFO	FOA
F26	Mean	2.6771E+05	2.6458E+06	4.1139E+08	2.8678E+06	2.1673E+06	4.5383E+07
	Std	1.0247E+05	1.1716E+06	8.5387E+07	8.0425E+05	4.5535E+05	1.1975E+07
F27	Mean	6.3168E+03	1.0319E+04	2.4705E+10	1.1223E+04	1.1413E+04	1.7003E+09
	Std	1.0293E+04	1.1213E+04	1.5153E+09	9.7927E+03	1.0930E+04	2.1837E+08
F28	Mean	6.4225E+01	8.9987E+01	2.2396E+03	1.0089E+02	8.5303E+01	2.2261E+02
	Std	5.0934E+01	1.1705E+01	3.0013E+02	4.0299E+01	4.1667E+01	5.7160E+01
F29	Mean	3.3693E+04	5.2699E+05	2.1158E+07	4.7974E+05	6.0921E+05	2.4066E+06
	Std	1.8553E+04	3.1305E+05	3.5783E+06	2.3522E+05	4.3922E+05	8.7454E+05
F30	Mean	3.4400E+02	3.4400E+02	5.3872E+02	3.4400E+02	3.4400E+02	3.8544E+02
	Std	2.6860E-12	6.5963E-07	3.8691E+01	2.6516E-12	3.3520E-12	1.0309E+01
F31	Mean	2.6752E+02	2.6538E+02	4.6079E+02	2.6825E+02	2.6586E+02	3.1213E+02
	Std	5.0026E+00	7.0454E+00	6.8300E+00	4.4449E+00	3.9383E+00	3.6751E+00
F32	Mean	2.1061E+02	2.1388E+02	2.7124E+02	2.1691E+02	2.1542E+02	2.2054E+02
	Std	5.5300E+00	5.9914E+00	1.1291E+01	6.2484E+00	5.2166E+00	5.2494E+00

TABLE 13: Statistical results obtained by ISSA, SSA, PSO, CMFOA, IFFO, and FOA through 10 independent runs on CEC 2014 benchmark functions with $n = 100$.

Fun		ISSA	SSA	PSO	CMFOA	IFFO	FOA
F26	Mean	1.0395E+06	4.9662E+07	1.9596E+09	1.0516E+07	1.5208E+07	2.8282E+08
	Std	3.6972E+05	5.6939E+06	2.1605E+08	3.5784E+06	5.0169E+06	4.4860E+07
F27	Mean	1.4837E+04	5.8871E+05	1.0093E+11	2.6410E+04	3.7388E+04	7.1189E+09
	Std	1.5318E+04	1.0255E+05	1.0099E+10	2.8473E+04	4.1209E+04	3.2998E+08
F28	Mean	1.3263E+02	2.4979E+02	1.1962E+04	2.2607E+02	2.3713E+02	8.4991E+02
	Std	4.3021E+01	7.0814E+01	1.4132E+03	4.5595E+01	2.4642E+01	1.0057E+02
F29	Mean	1.6986E+05	4.2648E+06	1.7618E+08	3.1738E+06	2.8874E+06	1.8248E+07
	Std	6.2432E+04	1.1220E+06	2.7101E+07	4.2353E+05	1.3296E+06	6.2005E+06
F30	Mean	3.4823E+02	3.4875E+02	1.4344E+03	3.4910E+02	3.4901E+02	5.7172E+02
	Std	6.2960E-11	4.3294E-01	1.5590E+02	9.1883E-01	9.3000E-01	2.8371E+01
F31	Mean	3.4722E+02	3.5878E+02	9.2092E+02	3.5108E+02	3.4814E+02	5.0149E+02
	Std	1.0958E+01	3.7623E+00	2.4898E+01	1.0734E+01	1.0706E+01	1.0838E+01
F32	Mean	2.4544E+02	2.5216E+02	5.2841E+02	2.6036E+02	2.6337E+02	2.9287E+02
	Std	1.5945E+01	1.3749E+01	2.4285E+01	1.2685E+01	1.5913E+01	1.1210E+01

TABLE 14: Results of Wilcoxon's test for ISSA against other six algorithms for each benchmark function with 10 independent runs and $n = 30$ ($\alpha = 0.05$).

Fun	SSA vs ISSA		PSO vs ISSA		CMFOA vs ISSA		IFFO vs ISSA		FOA vs ISSA	
	p-value	win	p-value	win	p-value	win	p-value	win	p-value	win
F1	1.5723E-03	+	5.4503E-11	+	2.1431E-06	+	1.2930E-04	+	3.1274E-08	+
F2	5.9105E-01	-	5.9726E-07	+	1.6785E-01	-	1.6785E-01	-	1.7438E-06	+
F3	1.8034E-01	-	5.6302E-11	+	6.6374E-01	-	4.413E-06	+	1.8978E-10	+
F4	3.9391E-03	+	8.0559E-08	+	8.0897E-04	+	1.4754E-03	+	1.0215E-06	+
F5	7.5194E-07	+	3.5327E-08	+	2.2706E-01	-	4.2611E-02	+	1.5497E-06	+
F6	2.2263E-02	+	1.8702E-08	+	2.7096E-03	+	3.3147E-03	+	7.3030E-06	+
F7	3.9878E-03	+	2.1023E-10	+	2.6126E-07	+	1.1038E-04	+	5.8740E-11	+
F8	3.7778E-07	+	1.2311E-13	+	2.2556E-07	+	8.8317E-11	+	1.6744E-07	+
F9	2.5658E-06	+	3.9583E-05	+	2.0251E-08	+	2.7652E-08	+	6.8325E-02	-
F10	4.0986E-03	+	1.5715E-10	+	6.2372E-07	+	1.0581E-05	+	7.5777E-10	+
F11	1.6385E-01	-	5.5101E-04	+	4.5288E-03	+	6.2300E-02	-	1.4019E-03	+
F12	2.5148E-04	+	1.7221E-15	+	8.8689E-10	+	8.2337E-10	+	8.4091E-04	+
F13	6.2223E-04	+	8.2292E-11	+	1.7434E-04	+	6.8585E-02	-	5.6801E-08	+
F14	1.6770E-05	+	3.5961E-16	+	6.0168E-13	+	2.4086E-12	+	1.0063E-06	+
F15	9.1211E-03	+	4.2859E-14	+	7.9924E-01	-	9.6191E-01	-	1.2100E-14	+
F16	4.9253E-05	+	2.4808E-06	+	8.1048E-03	+	4.9672E-03	+	3.5094E-08	+
F17	5.2276E-01	-	1.1956E-10	+	1.6338E-01	-	8.7704E-01	-	1.2329E-18	+
F18	5.9605E-02	-	7.3103E-10	+	7.5245E-01	-	8.3423E-01	-	1.4080E-08	+
F19	4.0911E-03	+	2.0151E-06	+	4.5217E-03	+	9.3504E-03	+	6.9674E-08	+
F20	8.9857E-02	-	1.0735E-03	+	2.9254E-01	-	7.6513E-01	-	1.2493E-05	+
F21	8.0383E-04	+	1.3653E-14	+	-	=	4.9618E-05	+	5.1686E-11	+
F22	9.6507E-05	+	5.1321E-12	+	-	=	1.9712E-04	+	2.5703E-10	+
F23	1.0362E-03	+	3.7568E-14	+	1.6044E-02	+	1.9660E-04	+	7.4376E-08	+
F24	8.2001E-07	+	1.6038E-14	+	4.8491E-04	+	1.6951E-10	+	1.8472E-09	+
F25	1.4795E-03	+	1.2097E-06	+	1.9763E-01	-	4.3929E-02	-	8.2364E-08	+
F26	2.9892E-05	+	1.2127E-06	+	1.3438E-04	+	3.8826E-04	+	1.1510E-07	+
F27	2.4771E-03	+	7.7797E-10	+	2.5931E-02	+	9.5563E-03	-	3.8874E-12	+
F28	1.1525E-03	+	2.1817E-09	+	2.3075E-02	+	7.6652E-03	+	1.0245E-07	+
F29	9.9588E-05	+	3.4016E-06	+	6.1373E-05	+	2.1918E-03	+	2.3509E-05	+
F30	9.0190E-02	-	1.2454E-07	+	7.1059E-05	+	1.6503E-06	+	3.3480E-04	+
F31	2.5587E-01	-	9.8592E-11	+	2.2578E-01	-	1.3543E-01	-	7.9203E-02	-
F32	3.1415E-01	-	5.5580E-06	+	7.1757E-02	-	2.0510E-01	-	3.4882E-01	-
+/-		29/3		32/0		20/10		23/9		29/3

TABLE 15: Results of Wilcoxon's test for ISSA against other six algorithms for each benchmark function with 10 independent runs and $n = 50$ ($\alpha = 0.05$).

Fun	SSA vs ISSA		PSO vs ISSA		CMFOA vs ISSA		IFFO vs ISSA		FOA vs ISSA	
	p-value	win	p-value	win	p-value	win	p-value	win	p-value	win
F1	2.0377E-06	+	5.1683E-10	+	4.4186E-07	+	5.5764E-07	+	3.1113E-12	+
F2	6.0105E-02	-	4.2014E-09	+	1.7277E-01	-	2.4422E-02	+	9.1132E-11	+
F3	1.7250E-06	+	1.3907E-16	+	9.8022E-02	-	1.0738E-05	+	1.8638E-11	+
F4	9.3262E-06	+	1.4595E-09	+	5.0379E-04	+	1.6848E-03	+	4.2472E-08	+
F5	5.7607E-10	+	9.2006E-10	+	2.3798E-02	+	8.1251E-01	-	1.0642E-06	+
F6	1.3107E-05	+	1.3362E-11	+	5.6932E-05	+	5.3828E-03	+	1.0919E-07	+
F7	5.7850E-07	+	1.8163E-10	+	6.7859E-05	+	3.5922E-05	+	1.3335E-10	+
F8	75219E-07	+	2.2270E-14	+	3.3394E-02	+	1.1235E-10	+	4.6085E-11	+
F9	3.9321E-08	+	3.3513E-01	-	2.6869E-10	+	3.7640E-09	+	1.7549E-01	-
F10	3.2994E-05	+	5.5796E-11	+	3.0272E-08	+	7.2141E-07	+	9.7090E-13	+
F11	2.4950E-02	+	1.8032E-05	+	3.9453E-02	+	7.8893E-02	-	3.0964E-04	+
F12	2.2790E-07	+	2.5730E-19	+	8.2015E-10	+	3.3180E-10	+	1.7587E-04	+
F13	8.6055E-06	+	2.6273E-12	+	2.3293E-03	+	9.9266E-05	+	9.8054E-12	+
F14	5.0086E-07	+	6.2475E-15	+	7.0383E-12	+	5.0688E-15	+	4.1146E-08	+
F15	1.7136E-01	-	1.3728E-13	+	9.4200E-01	-	5.9423E-01	-	3.3136E-15	+
F16	1.6083E-06	+	1.3679E-06	+	1.6464E-02	+	1.5895E-01	-	1.3483E-09	+
F17	2.9046E-01	-	3.9668E-14	+	6.8720E-01	-	6.2215E-01	-	2.9446E-18	+
F18	6.6743E-01	-	1.1386E-10	+	4.3569E-01	-	2.0341E-01	-	4.5540E-11	+
F19	3.6286E-03	+	9.2080E-07	+	2.7891E-03	+	1.0982E-02	+	2.8723E-09	+
F20	1.6305E-02	+	6.8713E-06	+	8.0834E-01	-	3.1893E-01	-	1.9845E-04	+
F21	2.1300E-06	+	1.7078E-12	+	-	=	4.9113E-08	+	3.2451E-13	+
F22	2.0294E-05	+	3.1368E-13	+	-	=	3.1089E-06	+	2.3903E-11	+
F23	1.2107E-04	+	6.0776E-15	+	7.7875E-06	+	7.0901E-05	+	1.7113E-09	+
F24	2.5888E-08	+	1.4322E-14	+	4.0414E-06	+	1.7080E-10	+	4.0917E-10	+
F25	3.1276E-06	+	3.9758E-08	+	9.8360E-01	-	4.9413E-01	-	4.5773E-08	+
F26	1.3214E-04	+	9.9102E-08	+	4.1042E-06	+	1.7402E-07	+	7.9545E-07	+
F27	1.6043E-01	-	1.9505E-12	+	3.4341E-01	-	3.9881E-01	-	1.4412E-09	+
F28	1.2130E-01	-	5.8692E-09	+	1.3887E-01	-	4.2578E-01	-	2.6464E-04	+
F29	8.4658E-04	+	1.6521E-08	+	2.0073E-04	+	2.7477E-03	+	1.1585E-05	+
F30	9.4213E-04	+	6.7411E-08	+	5.3101E-04	+	5.4640E-04	+	4.7099E-07	+
F31	4.6697E-01	-	4.2833E-14	+	7.9775E-01	-	4.0133E-01	-	1.1364E-10	+
F32	2.7813E-01	-	2.4129E-07	+	6.1643E-02	-	8.3535E-02	-	6.3552E-03	+
+/-		24/8		31/1		19/11		20/12		31/1

TABLE 16: Results of Wilcoxon's test for ISSA against other six algorithms for each benchmark function with 10 independent runs and $n = 100$ ($\alpha = 0.05$).

Fun	SSA vs ISSA		PSO vs ISSA		CMFOA vs ISSA		IFFO vs ISSA		FOA vs ISSA	
	p-value	win	p-value	win	p-value	win	p-value	win	p-value	win
F1	1.0378E-07	+	7.8176E-14	+	1.1254E-06	+	7.3355E-08	+	2.9716E-13	+
F2	4.2836E-05	+	8.2177E-12	+	4.9949E-02	+	2.6382E-03	+	7.2835E-09	+
F3	4.9896E-08	+	7.8338E-35	+	3.5536E-02	+	1.3895E-08	+	1.1550E-12	+
F4	2.3331E-06	+	1.9205E-10	+	2.1416E-04	+	5.2932E-06	+	1.9678E-08	+
F5	1.2600E-10	+	1.2963E-10	+	1.7828E-03	+	1.0868E-05	+	5.0309E-09	+
F6	9.8970E-02	-	5.3354E-10	+	4.7015E-06	+	1.6844E-05	+	6.1888E-10	+
F7	2.2243E-08	+	4.1865E-13	+	8.7771E-07	+	1.3044E-09	+	6.2464E-11	+
F8	2.2556E-10	+	5.3495E-18	+	7.4894E-05	+	7.9906E-11	+	3.1999E-09	+
F9	4.9870E-10	+	2.9549E-01	-	1.0030E-10	+	1.2423E-12	+	3.4344E-01	+
F10	4.6494E-07	+	3.0486E-15	+	1.9111E-07	+	1.5614E-09	+	9.4423E-13	+
F11	1.8990E-02	+	2.2724E-06	+	1.9056E-02	+	2.3614E-02	+	2.9444E-04	+
F12	4.3699E-06	+	1.2600E-22	+	3.2460E-10	+	1.4367E-09	+	6.0050E-05	+
F13	2.4541E-06	+	5.9980E-15	+	1.5823E-06	+	3.1849E-05	+	2.4334E-11	+
F14	6.3858E-07	+	4.5807E-17	+	2.2981E-12	+	1.2864E-09	+	8.6555E-13	+
F15	1.7146E-07	+	2.2593E-17	+	7.0366E-01	-	9.9469E-02	-	5.1238E-16	+
F16	3.9761E-07	+	8.1135E-12	+	4.1494E-03	+	6.2574E-03	+	7.9491E-02	+
F17	1.0397E-02	+	6.7363E-14	+	9.9961E-01	-	8.3209E-01	-	7.9210E-16	+
F18	8.6191E-01	-	1.7179E-15	+	7.9452E-01	-	4.3052E-01	-	1.7688E-13	+
F19	5.9040E-06	+	7.5177E-08	+	3.3686E-03	+	4.6936E-05	+	4.7998E-09	+
F20	9.0127E-04	+	7.2610E-05	+	3.7345E-01	-	1.8813E-01	-	1.3324E-05	+
F21	7.6534E-06	+	2.1239E-17	+	-	=	1.2438E-08	+	1.1562E-13	+
F22	2.6358E-06	+	2.9856E-16	+	-	=	4.4818E-09	+	1.7365E-13	+
F23	3.4130E-03	+	4.6644E-17	+	2.8070E-06	+	7.8756E-06	+	5.9044E-11	+
F24	3.6618E-07	+	1.8577E-15	+	6.0981E-08	+	1.6105E-12	+	4.7301E-10	+
F25	6.4937E-12	+	1.1756E-12	+	5.1565E-01	-	9.2513E-01	-	6.9216E-10	+
F26	5.6291E-10	+	3.6946E-10	+	1.3740E-05	+	1.2241E-05	+	9.4839E-09	+
F27	5.2495E-08	+	1.5615E-10	+	1.8874E-01	-	1.2714E-01	-	1.5781E-13	+
F28	2.6066E-03	+	8.6946E-10	+	7.5687E-04	+	4.3007E-05	+	3.6968E-09	+
F29	8.4514E-07	+	7.1725E-09	+	2.6646E-09	+	8.7814E-05	+	7.1732E-06	+
F30	4.4636E-03	+	3.8618E-09	+	1.5805E-02	+	2.7858E-02	+	1.2999E-09	+
F31	1.3273E-02	+	1.8782E-13	+	5.2897E-01	-	7.8331E-01	-	6.0488E-11	+
F32	3.7345E-01	-	8.6751E-10	+	9.3177E-02	-	6.1812E-03	+	2.0169E-06	+
+/-		29/3		31/1		22/8		25/7		31/1

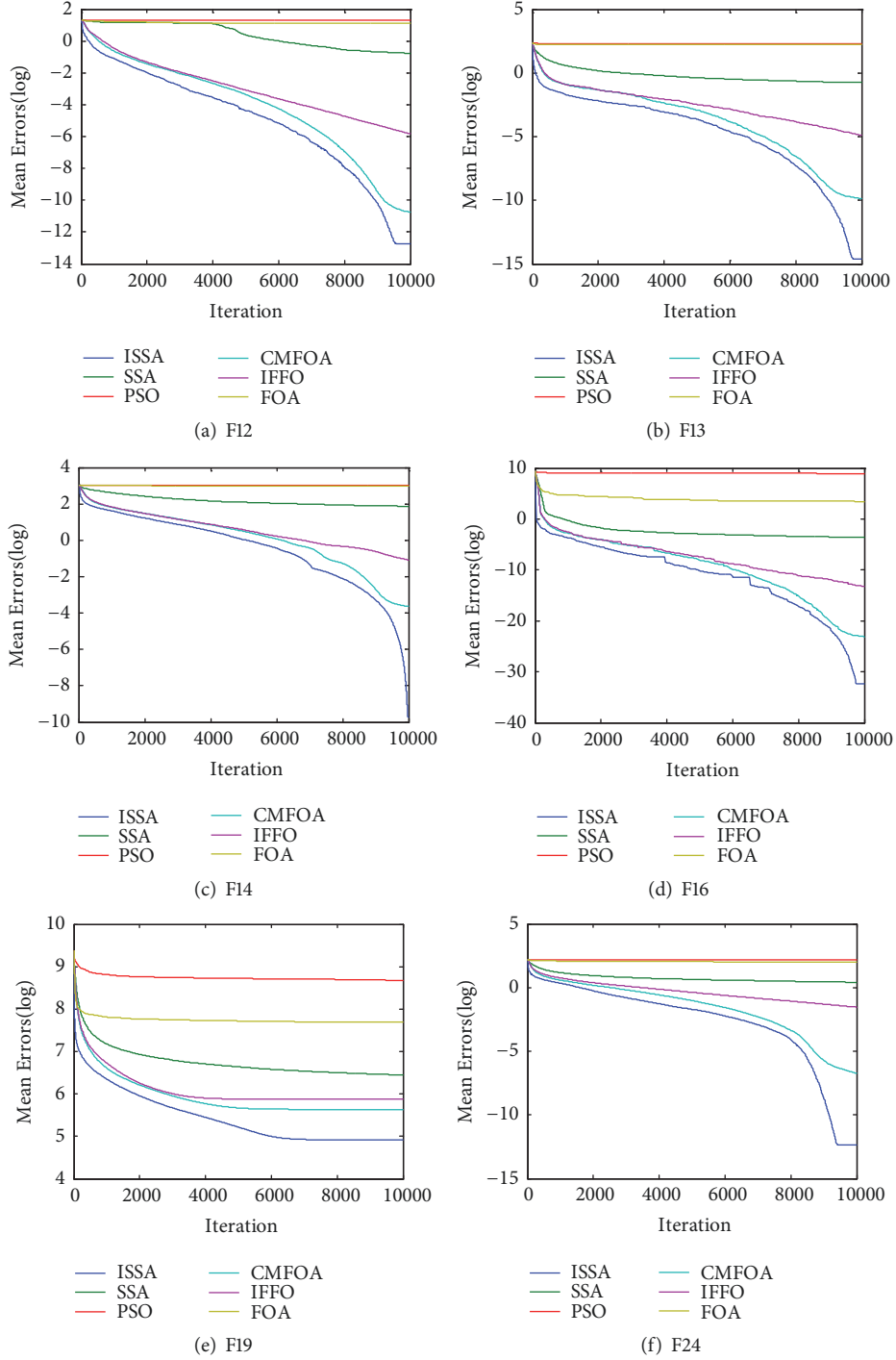


FIGURE 6: Convergence rate comparison for representative multimodal functions ($n = 100$).

where m_j is the mean of optimal values, k_j is the actual global optimal value, and N is the number of samples. In the present work, N is the number of benchmark functions. The MAE of all algorithms and their ranking for all functions are given in Table 17. It is clear to find that ISSA ranks No. 1 and provides the minimum MAE in all cases. ISSA reaches the optimum solution 653 times out of 960 runs (10 runs for each test function for $n = 30, 50$, and 100 , respectively) and comes in

the first rank as shown in Figure 10. It is concluded that ISSA provides the best performance in comparison to other five optimization algorithms.

5. Conclusions

The SSA is a new algorithm for searching global optimization based on the food foraging behavior of the flying squirrels.

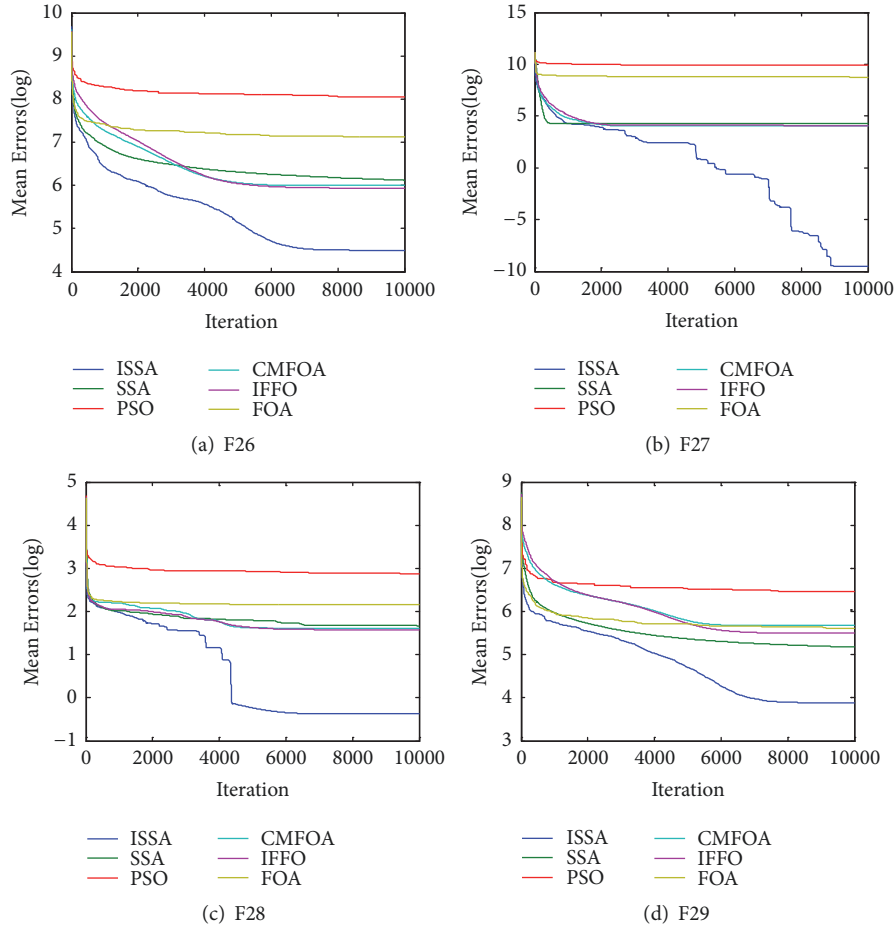


FIGURE 7: Convergence rate comparison for representative CEC 2014 functions ($n = 30$).

TABLE 17: Ranking of algorithms using MAE.

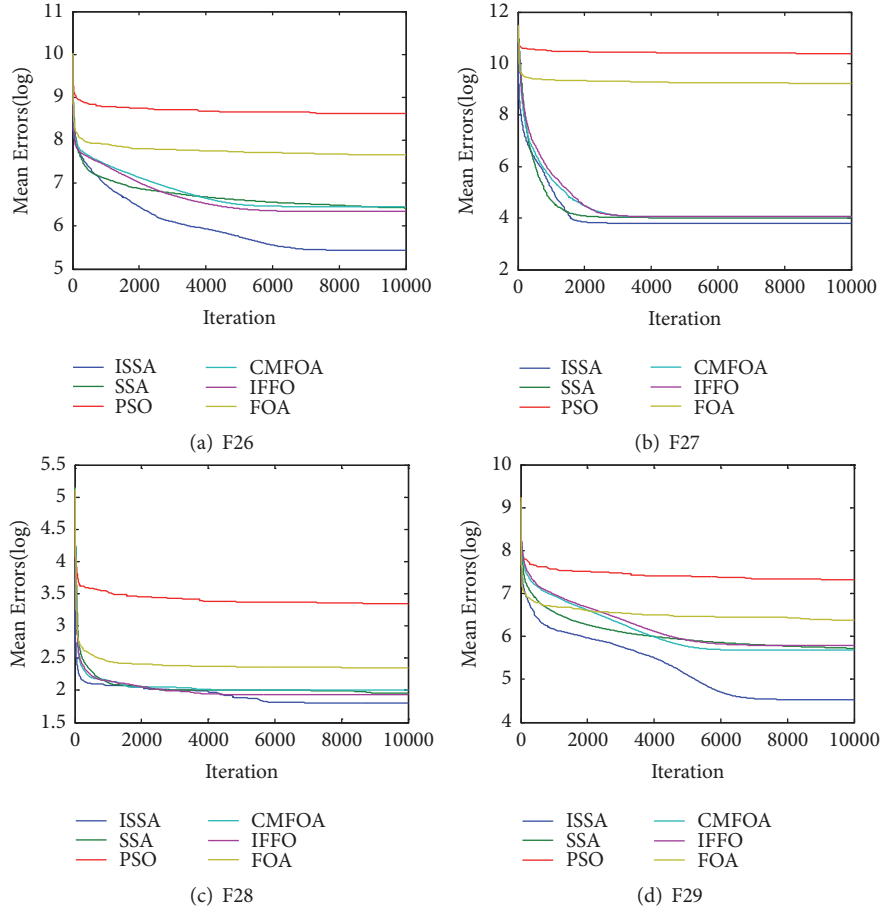
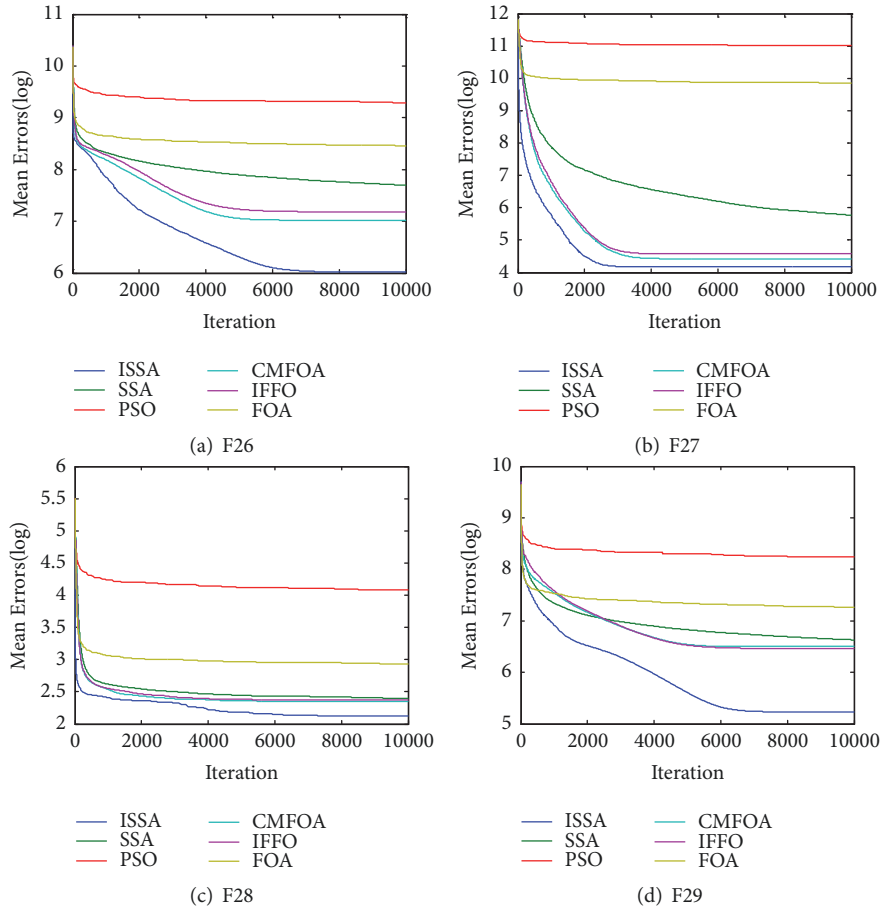
Algorithm	MAE (n=30)	MAE (n=50)	MAE (n=100)	Rank
ISSA	1.2399E+03	9.6479E+03	4.0880E+04	1
CMFOA	3.7162E+04	8.7565E+04	4.3758E+05	2
IFFO	4.6305E+04	1.0037E+05	5.8554E+05	3
SSA	4.6440E+04	1.0550E+05	1.7913E+06	4
FOA	1.9074E+07	5.5874E+07	4.4101E+25	5
PSO	2.7147E+08	1.4513E+09	2.2646E+31	6

To balance the exploration and exploitation capacities of the SSA, an improved SSA (ISSA) is proposed in this paper by introducing four strategies, namely, an adaptive predator presence probability, a normal cloud model generator, a selection strategy between successive positions, and enhancement in dimensional search. The adaptive strategy of predator presence probability assists to reach the potential search region at the early stage and then to implement the local search at the later stage. The normal cloud model is utilized to describe the randomness and fuzziness of the gliding behavior of the flying squirrel population. The selection strategy enhances the local search ability of an individual flying squirrel. In addition, enhancing dimensional search

results in a better quality of the best solution in each iteration. Extensive comparative studies are conducted for 32 benchmark functions (including 11 unimodal functions, 14 multimodal functions, and 7 CEC 2014 functions). Results confirm that the proposed ISSA is a powerful optimization algorithm and in general significantly outperforms five state-of-the-art algorithms (PSO, FOA, IFFO, CMFOA, and SSA).

Data Availability

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

FIGURE 8: Convergence rate comparison for representative CEC 2014 functions ($n = 50$).FIGURE 9: Convergence rate comparison for representative CEC 2014 functions ($n = 100$).

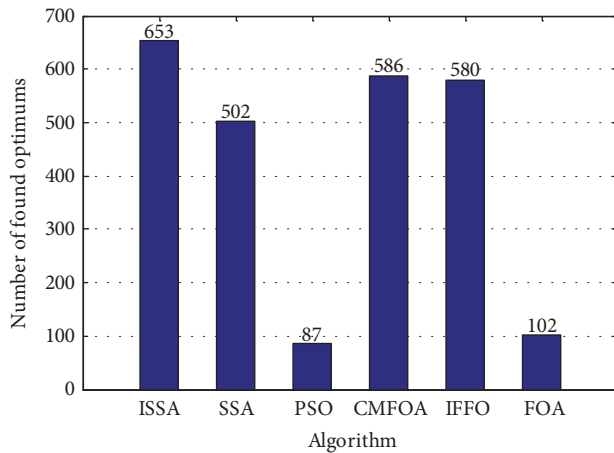


FIGURE 10: Comparison of algorithms in finding the global optimal solution out of 960 runs.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by a research grant from National Key Research and Development Program of China (project no. 2017YFC1500400) and the National Natural Science Foundation of China (51808147).

References

- [1] X. S. Yang, *Engineering Optimization: An Introduction with Metaheuristic Applications*, Wiley Publishing, 2010.
- [2] X.-S. Yang and A. H. Gandomi, "Bat algorithm: A novel approach for global engineering optimization," *Engineering Computations*, vol. 29, no. 5, pp. 464–483, 2012.
- [3] A. López-Jaimes and C. A. Coello Coello, "Including preferences into a multiobjective evolutionary algorithm to deal with many-objective engineering optimization problems," *Information Sciences*, vol. 277, pp. 1–20, 2014.
- [4] R. A. Monzingo, R. L. Haupt, and T. W. Miller, "Gradient-based algorithms, introduction to adaptive arrays," *IET Digital Library*, pp. 153–237, 2011.
- [5] R. J. Williams and D. Zipser, *Gradient-based learning algorithms for recurrent networks and their computational complexity*, Backpropagation, L. Erlbaum Associates Inc, 1995.
- [6] F. Ding and T. Chen, "Gradient based iterative algorithms for solving a class of matrix equations," *IEEE Transactions on Automatic Control*, vol. 50, no. 8, pp. 1216–1221, 2005.
- [7] M. Mohammadi and N. Ghadimi, "Optimal location and optimized parameters for robust power system stabilizer using honeybee mating optimization," *Complexity*, vol. 21, no. 1, pp. 242–258, 2015.
- [8] O. Abedinia, N. Amjadi, and A. Ghasemi, "A new metaheuristic algorithm based on shark smell optimization," *Complexity*, vol. 21, no. 5, pp. 97–116, 2016.
- [9] D. E. Goldberg, K. Sastry, and X. Llorca, "Toward routine billion-variable optimization using genetic algorithms," *Complexity*, vol. 12, no. 3, pp. 27–29, 2007.
- [10] J. H. Holland, "Genetic algorithms and the optimal allocation of trials," *SIAM Journal on Computing*, vol. 2, no. 2, pp. 88–105, 1973.
- [11] H. Beyer and H. Schwefel, *Evolution Strategies – A Comprehensive Introduction*, Kluwer Academic Publishers, 2002.
- [12] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, IEEE, 2002.
- [13] D. Karaboga and B. Basturk, *A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm*, Kluwer Academic Publishers, 2007.
- [14] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.
- [15] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [16] D. Bertsimas and J. Tsitsiklis, "Simulated annealing," *Statistical Science*, vol. 8, no. 1, pp. 10–15, 1993.
- [17] Z. Yin, J. Liu, W. Luo, and Z. Lu, "An improved Big Bang–Big Crunch algorithm for structural damage detection," *Structural Engineering and Mechanics*, vol. 68, no. 6, pp. 735–745, 2018.
- [18] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information Sciences*, vol. 213, pp. 267–289, 2010.
- [19] A. F. Nematollahi, A. Rahiminejad, and B. Vahidi, "A novel physical based meta-heuristic optimization method known as Lightning Attachment Procedure Optimization," *Applied Soft Computing*, vol. 59, pp. 596–621, 2017.
- [20] G. Dhiman and V. Kumar, "Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications," *Advances in Engineering Software*, vol. 114, pp. 48–70, 2017.
- [21] A. Baykasoğlu and Ş. Akpinar, "Weighted Superposition Attraction (WSA): A swarm intelligence algorithm for optimization problems – Part 1: Unconstrained optimization," *Applied Soft Computing*, vol. 56, pp. 520–540, 2017.
- [22] A. Kaveh and A. Dadras, "A novel meta-heuristic optimization algorithm: Thermal exchange optimization," *Advances in Engineering Software*, vol. 110, pp. 69–84, 2017.
- [23] A. Tabari and A. Ahmad, "A new optimization method: electrosearch algorithm," in *Computers & Chemical Engineering*, vol. 10, pp. 1–11, Chem, UK, 2017.
- [24] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.
- [25] T. Peram, K. Veeramachaneni, and C. K. Mohan, "Fitness-distance-ratio based particle swarm optimization," in *Proceedings of the Swarm Intelligence Symposium, 2003, SIS '03. Proceedings of the IEEE*, pp. 174–181, 2012.
- [26] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, 2004.
- [27] B. Y. Qu, P. N. Suganthan, and S. Das, "A distance-based locally informed particle swarm model for multimodal optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 3, pp. 387–402, 2013.
- [28] F. Zhong, H. Li, and S. Zhong, "A modified ABC algorithm based on improved-global-best-guided approach and adaptive-limit strategy for global optimization," *Applied Soft Computing*, vol. 46, pp. 469–486, 2016.

- [29] D. Kumar and K. Mishra, "Portfolio optimization using novel co-variance guided Artificial Bee Colony algorithm," *Swarm and Evolutionary Computation*, vol. 33, pp. 119–130, 2017.
- [30] S. Ghambari and A. Rahati, "An improved artificial bee colony algorithm and its application to reliability optimization problems," *Applied Soft Computing*, vol. 62, pp. 736–767, 2018.
- [31] W. T. Pan, "A new evolutionary computation approach: fruit fly optimization algorithm," in *Proceedings of the Conference of Digital Technology and Innovation Management*, Taipei, Taiwan, 2011.
- [32] W.-T. Pan, "Using modified fruit fly optimisation algorithm to perform the function test and case studies," *Connection Science*, vol. 25, no. 2-3, pp. 151–160, 2013.
- [33] L. Wang, Y. Shi, and S. Liu, "An improved fruit fly optimization algorithm and its application to joint replenishment problems," *Expert Systems with Applications*, vol. 42, no. 9, pp. 4310–4323, 2015.
- [34] X. F. Yuan, X. S. Dai, J. Y. Zhao, and Q. He, "On a novel multi-swarm fruit fly optimization algorithm and its application," *Applied Mathematics and Computation*, vol. 233, pp. 260–271, 2014.
- [35] Q.-K. Pan, H.-Y. Sang, J.-H. Duan, and L. Gao, "An improved fruit fly optimization algorithm for continuous function optimization problems," *Knowledge-Based Systems*, vol. 62, pp. 69–83, 2014.
- [36] T. Zheng, J. Liu, W. Luo, and Z. Lu, "Structural damage identification using cloud model based fruit fly optimization algorithm," *Structural Engineering and Mechanics*, vol. 67, no. 3, pp. 245–254, 2018.
- [37] M. Jain, V. Singh, and A. Rani, "A novel nature-inspired algorithm for optimization: Squirrel search algorithm," *Swarm and Evolutionary Computation*, 2018.
- [38] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Proceedings of the Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, vol. 284, pp. 65–74, Springer Berlin Heidelberg, 2010.
- [39] I. Fister, I. Fister Jr., X.-S. Yang, and J. Brest, "A comprehensive review of firefly algorithms," *Swarm and Evolutionary Computation*, vol. 13, no. 1, pp. 34–46, 2013.
- [40] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.
- [41] D. Li, H. Meng, and X. Shi, "Membership clouds and membership clouds generator," *International Journal of Computational Research and Development*, vol. 32, pp. 15–20, 1995.
- [42] D. Li, C. Liu, and W. Gan, "A new cognitive model: cloud model," *International Journal of Intelligent Systems*, vol. 24, no. 3, pp. 357–375, 2009.
- [43] J. Liang, B. Qu, and P. Suganthan, "Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization," Zhengzhou China and Technical Report, Computational Intelligence Laboratory, Zhengzhou University, Nanyang Technological University, Singapore, 2014.
- [44] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proceedings of the World Congress on Nature and Biologically Inspired Computing (NABIC '09)*, pp. 210–214, 2009.
- [45] M. Jamil and X.-S. Yang, "A literature survey of benchmark functions for global optimisation problems," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 4, no. 2, pp. 150–194, 2013.
- [46] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.
- [47] E. Nabil, "A modified flower pollination algorithm for global optimization," *Expert Systems with Applications*, vol. 57, pp. 192–203, 2016.

