An Improved Squirrel Search Algorithm With Reproductive Behavior

XUNCAI ZHANG1, (Member, IEEE), KAI ZHAO1, LINGFEI WANG1, YANFENG WANG1, AND YING NIU2

1College of Electrical and Information Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China
2College of Architecture Environment Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China

Corresponding author: Xuncai Zhang (zhangxuncai@pku.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61572446, Grant 61602424, and Grant U1804262, and in part by the Key Research and Development Program of Henan Province under Grant 202102210177 and Grant 192102210134.

ABSTRACT The squirrel search algorithm (SSA) is a recently proposed nature-inspired algorithm based on the dynamic foraging and gliding behavior of squirrels. Because of its simplicity and stability, the squirrel algorithm has attracted increasing research interest. However, the lack of exploration ability of the SSA may lead to premature convergence to the local optimum. To overcome this disadvantage, an improved SSA with reproductive behavior (RSSA) is proposed to solve the numerical optimization problem. First, the reproductive behavior of the invasive weed algorithm (IWO) is introduced to the conventional SSA to generate offspring individuals, and these offspring individuals are scattered into the search space by Gaussian distribution to complete the location update. This method makes it possible for individuals with poor fitness to enter the next generation search, improving the exploration ability of the SSA. Second, an adaptive step strategy is proposed to adaptively adjust the search step of squirrels according to the distance between each squirrel and other family members. This strategy effectively balances the exploration and exploitation of the algorithm. Finally, the performance of the proposed RSSA algorithm is evaluated using Wilcoxon’s test on unimodal, multimodal, fixed-dimensional multimodal and CEC 2014 benchmark functions. Experimental results and statistical tests show that RSSA has better performance in terms of convergence, accuracy, and search capability compared with other state-of-the-art algorithms.

INDEX TERMS Nature-inspired, squirrel search algorithm, exploration and exploitation, Wilcoxon’s test, CEC 2014 benchmark functions.

I. INTRODUCTION

In the field of computer science, optimization problems pose the challenge of finding the optimal solution among all feasible solutions. Traditional methods to solve such problems include the conjugate gradient method, branch and bound algorithm, and dynamic programming method [1], [2]. However, with the increasing scale of optimization problems, it is becoming increasingly difficult for traditional optimization algorithms to meet the requirements of time complexity. As a result, the need for more efficient algorithms is increasing. Over the past few decades, swarm intelligence (SI) has shown efficient and robust performance in solving modern nonlinear numerical global optimization problems [3], [4].

Swarm intelligence imitates the social behavior of animal groups. Because of its simplicity, flexibility, non-derivation mechanism, and avoidance of local optima, SI is widely applied in feature selection [5], hardware/software co-design [6], scheduling [7], agriculture [8], metallurgy [9], and military [10]. The most representative SI algorithms are particle swarm optimization (PSO) [11], ant colony optimization (ACO) [12], and artificial bee colony (ABC) [13]. The social behavior of birds inspires PSO. It uses individual cognitive items to guide the birds towards the best position that has been found. Additionally, social cognitive items are used to guide all birds towards the global optimum. Because of its simplicity and stability, PSO is continuously improved to solve more complicated optimization problems [14]–[16]. Ant colony optimization (ACO) is inspired by the social behavior of ant colonies [17]. The pheromone matrix of the ant colony...
evolves continuously during the iteration of the candidate solution and eventually finds the optimal solution. ACO has the advantages of simple structure, ease of use, and excellent performance. However, its convergence rate is slow in complex optimization problems. Recently, researchers combined the high-speed parallel computing ability of GPUs to improve the convergence speed of the ABC algorithm and achieved excellent results [18]–[20]. The artificial bee colony algorithm simulates the intelligent foraging behavior of a honeybee colony [21]. Owing to its outstanding global search capability, ABC has been successfully utilized in many applications, i.e., parameter estimation [22] and wireless sensor networks [23]. It has been proven that there is no algorithm generic enough to solve all optimization problems [24]. Therefore, many new SI algorithms emerge every year, such as the grey wolf optimization (GWO) [25], firefly algorithm (FF) [26], cuckoo search (CS) [27], dragonfly algorithm (DA) [28], and squirrel search algorithm (SSA) [29].

The squirrel search algorithm (SSA) is a new swarm intelligent optimization algorithm initially developed by Jain in 2018 [29]. It is inspired by the dynamic foraging and gliding behavior of squirrels. SSA searches the global optimum by gliding among different kinds of trees (normal tree, acorn tree, and hickory tree), searching for food sources, and avoiding predators. The performance of SSA is tested on classical and modern CEC 2014 benchmark functions. The results demonstrate the effectiveness of SSA in comparison with some state-of-the-art optimization algorithms, such as PSO, ABC, bat algorithm (BA), and firefly algorithm (FF). Since the development of SSA, it has been successfully applied to various optimization problems such as structural design [30], image analysis [31], and production scheduling [32], [33].

In the conventional SSA, however, the exploration ability of SSA is regarded as weak, and it has had problems with premature convergence [31]. Moreover, the convergence velocity of SSA is too low, and its performance decreases as the search space dimensionality increases. Therefore, a considerable number of research studies have been conducted to improve the global performance of conventional SSA. For example, Wang and Du proposed an improved squirrel search algorithm (ISSA) [34], which utilizes the jumping search method and the progressive search method to strengthen the diversity of global searching. The experimental results on 22 benchmark functions demonstrated that ISSA could improve the performance of the squirrel search algorithm. Walaa et al. proposed the combination of the best-fit heuristic (BF) [35] and squirrel swarm algorithm to improve the global search ability of the algorithm named (MSBPP) [36]. In MSBPP, the best-fit heuristic is utilized to generate a set of initial solutions. Besides, the MSBPP introduced an operator strategy to solve the one-dimensional bin-pacing problem. However, it neglects the running time of the algorithm and has a weak effect in high-dimension problems. To increase population diversity, Wang et al. proposed an improved squirrel search algorithm (ISSA) with spatial variation and diffusion behavior to update the positions of squirrels [30]. The results indicated that the ISSA outperformed conventional SSA. In addition to the above improvement measures, other recent techniques to improve the SSA and their advantages, disadvantages, and applications are listed in TABLE 1. However, their performance is still deficient in solving problems.

### TABLE 1. Recent techniques to improve the SSA.

| Algorithm | Improvement measures | Advantages | Disadvantages | Application |
|-----------|----------------------|------------|---------------|-------------|
| ISSA [37] | 1. An adaptive strategy of predator presence probability is proposed, which dynamically adjusts with the iteration process. 2. A normal cloud generator is used to generate new locations for squirrels, which improves the exploration capability of SSA. | Premature convergence is avoided, and the intensive search ability is improved. | The algorithm is not stable enough and has a weak effect on some benchmark functions. | – |
| ACS-SSA [31] | The random flight operator of an adaptive cuckoo algorithm (ACS) is introduced into SSA to strengthen the exploration ability of the SSA. | The accuracy of the solution is improved, and the performance of the algorithm is stable. | Convergence speed is not improved, and the performance is poor in high-dimensional benchmark functions. | Brain Image Analysis |
| SSAWO [38] | Combine the SSA with a whale optimization algorithm (WOA). Then modify the results of the SSA using the crossover operator and mutation function in WOA. | The global search ability and the robustness are improved. | Slow convergence | Optimal power flow management |
| CSSA [32] | The chaotic squirrel search algorithm (CSSA) introduced a jump search measure to increase the ability of exploration in the early stage of evolution. | Fast convergence and high robustness | The accuracy is not high in high-dimensional problems. | Multi-objective task scheduling |
with difficult mathematical properties, and the abilities of exploration and exploitation are not adequately balanced, resulting in poor robustness of the algorithm. To overcome these problems, we improve a squirrel search algorithm with reproductive behavior (RSSA) is proposed in this paper. The main contributions of this paper are summarized as follows:

1) A reproduction mechanism is proposed to increase the number of candidate solutions in each iteration to improve the population diversity of the SSA. This mechanism improves the exploratory ability of the algorithm to avoid local optimization.

2) The position updating formulas are combined with the adaptive step size strategy. In this way, the exploitation and exploration abilities of the algorithm can be properly balanced. It also avoids the complex parameter adjustment of the algorithm.

The remainder of the paper is organized as follows. Section II briefly recapitulates the basic theory of the squirrel search algorithm. Next, the proposed RSSA with reproduction mechanism and adaptive step size strategy is presented in detail in Section III. Experimental results and analysis are illustrated in Section IV. Finally, Section V gives the conclusion.

II. AN OVERVIEW OF THE SQUIRREL SEARCH ALGORITHM
The squirrel search algorithm imitates the dynamic foraging behavior of squirrels and their efficient method of locomotion, known as gliding. It is a new swarm intelligence algorithm proposed by Jain et al. in 2018 [29]. A squirrel is a kind of arboreal rodent that moves mainly by gliding. Flying squirrels are an arboreal and nocturnal type of rodent that are exceptionally adapted for gliding locomotion. Squirrel gliding is considered the most complex form of aerodynamic motion and the most energy efficient [39]. Its dynamic foraging behavior also makes the most efficient use of food resources [40]. In the SSA, there are four necessary assumptions:

1) In a deciduous forest, there are $n$ squirrels and $n$ trees, one squirrel in one tree.

2) These $n$ trees contain a hickory tree and $N_{fs} \ (1 < N_{fs} < n)$ acorn trees, and the rest are normal trees.

3) There are only three types of trees in the forest. The hickory tree has the best food source (hickory nuts), acorn trees have a general food source (acorn nuts) and normal trees have no food.

4) Each squirrel seeks food individually and utilizes the available food resources by showing dynamic foraging behavior.

A. INITIALIZE THE POPULATION
There are $n$ squirrels in a forest. $FS_U$ and $FS_L$ are the lower and upper bounds in the search space. According to formula (1), $n$ individual squirrels are randomly generated:

$$FS_i = FS_L + rand(1, d) \times (FS_U - FS_L), \quad (i = 1, 2, \ldots, n)$$

where $FS_i$ is the $i^{th}$ squirrel, $rand(1, d)$ is a $1 \times d$ random matrix in the range $[0,1]$, and $d$ is the dimension of the problem.

B. CLASSIFY THE POPULATION
The positions of the squirrels are evaluated by the fitness function $f$. The decision variable (position vector of the squirrels) is input into the fitness function, and the corresponding result $f(FT_i)$ is the fitness value of the $i^{th}$ squirrel. The fitness value represents the quality of the food source sought by the $i^{th}$ squirrel. Then, the fitness values of all the squirrels are sorted in ascending order. The squirrel in the hickory tree ($FS_{h}$) represents the individual with the minimum fitness value. The squirrels in the acorn trees ($FS_{a}$) represent the individuals with fitness values ranging from 2 to $N_{fs} + 1$. The remaining individuals are denoted as squirrels in the normal tree ($FS_{n}$).

C. UPDATE THE POSITIONS
The positions of the individuals are updated by gliding to trees of different species. The $n_1$ squirrels that are in normal trees may move towards the acorn trees, whereas the remaining $n_2$ squirrels may move towards the hickory tree. The $n_3$ squirrels that are in acorn trees may move towards the hickory tree. In these cases, the new positions of the squirrels can be obtained as follows:

$$FS_{i}^{t+1} = \begin{cases} FS_{i}^{t} + d_g \times G_C \times (FS_{h}^{t} - FS_{i}^{t}) & \text{random location} \\ \text{otherwise} & \end{cases} \quad R \geq P_{dp}$$

$$FS_{i}^{t+1} = \begin{cases} FS_{i}^{t} + d_g \times G_C \times (FS_{a}^{t} - FS_{i}^{t}) & \text{random location} \\ \text{otherwise} & \end{cases} \quad R \geq P_{dp}$$

where $t$ denotes the current iteration, $R$ is a random number in the range of $[0,1]$, and $P_{dp}$ denotes the predator presence probability. If $R \geq P_{dp}$, the squirrels are safe and glide through the forest in search of food. If $R < P_{dp}$, the squirrels are at risk of predation and are forced to use a random walk (described in Section II (D)) to find a nearby hiding place. $G_C$ denotes the gliding constant. In this work, the value of $G_C$ is considered 1.9. The $d_g$ is the random gliding distance, which is computed as:

$$d_g = \frac{h_g}{\tan(\varphi)} \times sf$$

where $h_g$ is the constant value 8; $sf$ is the constant value 18; and $\tan(\varphi)$ represents the gliding angle, which can be calculated by formula (6):

$$\tan(\varphi) = \frac{D}{L}$$

where $D$ is the drag force and $L$ is the lift force, which can be calculated by formulas (7) and (8):

$$D = \frac{1}{2 \rho V^2 S C_D}$$
\[ L = \frac{1}{2\rho V^2 S C_L} \] (8)

where \( \rho (=1.204 \text{ kgm}^{-3}) \) is the air density, \( V (=5.25 \text{ m} s^{-1}) \) is the gliding speed of the squirrel, \( S (=154 \text{ cm}^2) \) is the surface area of the body, \( C_D (=0.6) \) is the frictional drag coefficient, and \( C_L \) represents a random number between 0.675 and 1.5.

**D. SEASONAL TRANSITION JUDGMENT**

The seasonal monitoring condition introduced in SSA helps the algorithm to jump out of the local optimum [29]. At the beginning of each iteration, the SSA requires the entire population to be in the winter state, which means that all individuals are updated as described in Section II (C). When all individuals are updated, the change in seasons is determined according to formulas (9) and (10):

\[
S'_i = \sqrt{\sum_{k=1}^{d} (FS_{h,k} - FS_{aj,k})^2} \quad j = 1, 2, \ldots, N_{fs} \quad (9)
\]

\[
S_{min} = \frac{10^{-6}}{(365)^{2/3}} \quad (10)
\]

where \( T \) is the maximum number of iterations and \( t \) denotes the current number of iterations. When the seasonal constant is less than the minimum \( (S'_i < S_{min}) \), the season changes from winter to summer. When the season changes, all individuals gliding to \( F_h \) stay at the updated positions, and all individuals gliding to \( F_a \) without encountering a predator relocate their positions according to formula (11).

\[
FS_{i}^{new} = FS_L + \text{Lévy} (n) \times (FS_U - FS_L) \quad (11)
\]

The Lévy distribution can be calculated by formula (12):

\[
\text{Lévy} (x) = 0.01 \times \frac{r_a \times \xi}{|r_h|^\beta} \quad (12)
\]

where \( r_a \) and \( r_h \) are normally distributed random numbers in \([0,1] \), \( \beta \) represents the constant value 0.5, and \( \xi \) can be calculated by formula (13):

\[
\xi = \left( \frac{\Gamma (1+\beta) \times \sin (\frac{\pi \beta}{2})}{\Gamma \left( \frac{1+\beta}{2} \right) \times \beta \times 2^{\frac{\beta}{2}}} \right)^{\frac{1}{\beta}} \quad (13)
\]

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

The procedure of the conventional SSA is shown in FIGURE 1.

**A. THE REPRODUCTION MECHANISM OF SQUIRRELS**

In the original SSA, the optimal and suboptimal solutions of the current generation guide the search agent to update the location. However, ignoring the possibility that there are still good individuals in the vicinity of individuals with poor fitness results in poor global search capabilities. Inspired by the IWO algorithm [41], the reproductive behavior of weeds is introduced into the SSA to guide the generation of candidate solutions.

First, according to the reproductive ability of squirrels, it is determined that squirrels will breed and produce offspring in better living areas. For example, the squirrels in hickory and acorn trees have abundant food sources and are apt to survive and reproduce. The number of reproductions is determined according to the fitness value. The number of offspring produced varies linearly with the fitness of the father squirrel. The formula is as follows:

\[
N_i = f(FS_i) - f_{min} \times (S_{max} - n) + n \quad (i = 1, 2, \ldots, n) \quad (14)
\]

where \( f(FS_i) \) is the fitness value of the parent squirrel, \( f_{max} \) and \( f_{min} \) are the maximum and minimum fitness of the population, and \( S_{max} \) is the maximum population.

Then, the numbers of generated offspring in RSSA are normally distributed random numbers with a mean equal to 0 and standard deviation \( FS_{iter}^{\sigma} \). The standard deviation of the random numbers is reduced from \( \sigma_{max} \) to \( \sigma_{min} \) during each iteration. The standard deviation of the population distribution \( FS_{iter}^{\sigma} \) is defined as:

\[
FS_{iter}^{\sigma} = \frac{(Iter_{max} - Iter)^h}{Iter_{max}^{h}} (\sigma_{max} - \sigma_{min}) + \sigma_{min} \quad (15)
\]

where \( \sigma_{max} \) is the maximum standard deviation, \( \sigma_{iter} \) is the minimum standard deviation, \( Iter \) is the maximum number of iterations, and \( h \) is the nonlinear harmonic exponent. In the present work, the value of \( h \) is set to 2.

It is worth noting that the squirrels with poor fitness values still have a chance to generate new candidate solutions, and...
Algorithm 1: Pseudocode of the Squirrel Reproduction Mechanism

1: Determine the initial swarm size $N$ and number of iterations $T$;
2: Randomly generate the initial population of squirrels: $x_i = FS_i$;
3: Initialize $a_0, a_f, N_0$;
4: Evaluate the fitness value of each squirrel;
5: Determine the number of offspring squirrels $N_o$ using Eq.(14);
6: for $i = 1$ to $N$ do
7: for $j = 1$ to $N_o$ do
8: $X(i, j) = x_i + randn(x_i, FS_{iter})$;
9: end for
10: if $X$ is beyond the boundary then
11: Map $X$ to the feasible region;
12: end if
13: end for

one of these candidate solutions is likely to be the global optimum. Therefore, the reproduction mechanism eliminates the limitation of finding food only at the optimal position, which helps improve the exploration ability of the algorithm.

The pseudocode of the proposed reproduction mechanism is shown in Algorithm 1.

B. ADAPTIVE STEP SIZE STRATEGY

In the literature [29], the exploration and exploitation abilities are balanced by adjusting the sliding constant $G_C$. The larger the $G_C$ value, the larger the step size of the squirrel movement, and the stronger the exploration ability. For different optimization problems, however, the value of $G_C$ needs to be continuously adjusted to achieve the balance of search capability. This complex parameter adjustment is not conducive to solving complex optimization problems. In our study, an adaptive step size mechanism was introduced. The search step of the search agent is adaptively adjusted according to the distance between the search agent and other individuals.

First, the distance between the current position of a squirrel and its target position needs to be calculated. The formulas for calculating the distance are as follows:

$$r_{hn} = FS_h - FS_n = \sqrt{\sum_{d=1}^{D} (FS_{h,d} - FS_{n,d})^2}$$  \hspace{1cm} (16)

$$r_{an} = FS_a - FS_n = \sqrt{\sum_{d=1}^{D} (FS_{a,d} - FS_{n,d})^2}$$  \hspace{1cm} (17)

$$r_{na} = FS_h - FS_a = \sqrt{\sum_{d=1}^{D} (FS_{h,d} - FS_{a,d})^2}$$  \hspace{1cm} (18)

where $d = 1, 2, \ldots, D$; $D$ denotes the dimension; and $FS_h$, $FS_a$, and $FS_n$ are the optimal individual, suboptimal individuals, and general candidate solution, respectively.

Then, the adaptive step sizes are generated according to the distance between the current position and the target position.

$$Step_{hn} = r_0 e^{-r_{hn}}, \quad Step_{an} = r_0 e^{-r_{an}}, \quad Step_{na} = r_0 e^{-r_{na}}$$  \hspace{1cm} (19)
Algorithm 2 Pseudocode of RSSA Algorithm

01: Initialize $T$, $N_0$, $N$, $\sigma_0$, $\sigma_f$, $h$, $\text{Iter}_{\text{max}}$ and the population $FS_i$ with $n$ squirrels using Eq.(1)
02: Evaluate the fitness of squirrels and then determine their locations: $FS_h$, $FS_a$, $FS_n$
03: while $\text{Iter} < \text{Iter}_{\text{max}}$
04: for $i_1 = 1$ to $n1$
05: if $R \geq P_{dp}$
06: $FS_n^{t+1} = FS_n^t + d_g G_C (FS_n^t - FS_n^t) \times \text{Step}_{hn}$% adaptive step size factor
07: else
08: $FS_n^{t+1} = \text{rand}(1, D) \times (U - L) + L$
09: end
10: end
11: for $i_2 = 1$ to $N_{S3}$
12: Distribute the position of offspring according to Eq.(15)
13: $FS_i^{\text{new}} = FS_i + \text{randn}(FS_i, FS_{\sigma}^\text{her})$
14: end
15: end
16: for $i_3 = 1$ to $n2$
17: if $R \geq P_{dp}$
18: $FS_a^{t+1} = FS_a^t + d_g G_C (FS_a^t - FS_a^t) \times \text{Step}_{ha}$% adaptive step size factor
19: else
20: $FS_a^{t+1} = \text{rand}(1, D) \times (U - L) + L$
21: end
22: end
23: for $i_4 = 1$ to $N_{S3}$
24: Distribute the position of offspring according to Eq.(15)
25: $FS_i^{\text{new}} = FS_i + \text{randn}(FS_i, FS_{\sigma}^\text{her})$
26: end
27: end
28: for $i_5 = 1$ to $n3$
29: if $R \geq P_{dp}$
30: $FS_a^{t+1} = FS_a^t + d_g G_C (FS_a^t - FS_a^t) \times \text{Step}_{ha}$% adaptive step size factor
31: else
32: $FS_a^{t+1} = \text{rand}(1, D) \times (U - L) + L$
33: end
34: end
35: for $i_6 = 1$ to $N_{S3}$
36: Distribute the position of offspring according to Eq.(15)
37: $FS_i^{\text{new}} = FS_i + \text{randn}(FS_i, FS_{\sigma}^\text{her})$
38: end
39: end
40: if $S_i^t < S_{\text{min}}$% Calculate seasonal constant($S_i$) using Eq.(9)
41: $FS_i^{\text{new}} = FS_L + \text{Levy}(n) \times (FS_U - FS_L)$
42: end
43: $f_i^{\text{new}} = f([FS_h^{\text{new}}, FS_a^{\text{new}}, FS_n^{\text{new}}])$% Evaluate the fitness values of offspring and parent individual
44: Sort fitness values and then determine the locations of squirrels: $FS_h^{\text{new}}$, $FS_a^{\text{new}}$, $FS_n^{\text{new}}$
45: if (The population size exceeds the upper bound $N$)
46: $f_i^{\text{new}} = f_i^{\text{parent}}(1: N)$;
47: end
48: The location of squirrel on hickory nut tree ($FS_h^{\text{new}}$) is the final optimal solution
49: $\text{Iter} = \text{Iter} + 1$
50: End
locations of squirrels on hickory nut trees \((FS_h)\), acorn trees \((FS_a)\), and normal trees \((FS_n)\) (Line 2). Then, the squirrel are randomly selected so that \(n_1\) squirrels in normal trees to move towards the hickory nut tree and \(n_2\) squirrels move towards acorn trees. The remaining \(n_3\) squirrels, which are in acorn trees, will move towards the hickory nut tree. The locations of the squirrels are updated according to Eq. (20), Eq. (21) and Eq. (22) with the proposed adaptive step size strategy (Lines 4-9, 16-21 and 28-33). Then, the offspring of the squirrels are generated according to the proposed reproduction mechanism in Section III (A) and the positions of the offspring are distributed in the surrounding search space (Lines 10-14, Lines 22-26 and Lines 34-38). Then, the algorithm calculates the seasonal constant and the minimum seasonal constant and sets the constraint conditions for seasonal change. If the conditions are met, the positions of squirrels are reset with formula (11) (Lines 40-42). Then it is determined whether the population size exceeds the upper bound. The individuals with better fitness values are selected as the initial population of the next iteration (Line 45-47). Finally, the optimal solution (Line 48) is returned and the above steps are repeated until the stopping criterion is satisfied.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, the effectiveness of the RSSA in solving numerical optimization problems is verified using a series of experimental tests and statistical analysis. Twenty-two benchmark functions are provided in Section IV (A). The parametric settings of the proposed RSSA and other state of the art algorithms are given in Section IV (B). The experimental results are provided in Section IV (C), and the statistical analysis of the results on the benchmark functions is discussed in Section IV (D).

A. BENCHMARK FUNCTION

Twenty-two benchmark functions of different categories are selected to test the performance of the proposed RSSA including unimodal functions [42], multimodal functions [43], fixed-dimensional multimodal functions [44] and the composition functions in CEC 2014 benchmark functions [45]. The unimodal functions contain only one optimal solution and are easy to solve. Therefore, they are often used to test the exploitation ability of algorithms. These unimodal functions are shown in TABLE 2. In addition, multimodal functions are a class of functions that contain multiple local optimal solutions. Moreover, the number of local optimal solutions increases exponentially with increasing dimensionality. These multimodal functions are often used to test the exploration ability of algorithms. The fixed-dimensional multimodal functions are defined on a specific dimensional search space and have multiple local optimal solutions. They are necessary to evaluate the exploration ability of algorithms. The multimodal functions and fixed-dimensional multimodal functions used in this experiment are shown in TABLE 3. In addition, eight CEC 2014 functions are selected to evaluate the comprehensive balance ability of the exploration and exploitation of the algorithm. TABLE 4 shows these CEC 2014 functions.

B. PARAMETER SETTING

The proposed RSSA is compared with five other nature-inspired algorithms, including the improved squirrel search algorithm (ISSA) [37], firefly algorithm (FF) [46], invasive weed optimization (IWO) [41], artificial bee colony (ABC) [47] and the conventional SSA [29]. To more clearly show the parameters used in this study, Table 5 lists the parameter symbols, definitions, and parameter values of all algorithms. In this table, the ‘−’ symbol indicates that the algorithm has no corresponding parameters. The maximum iterations \((T)\) and population size \((N)\) (common parameters of the algorithms) are set to 1000 and 50 in the first two experiments. Considering the complexity of the CEC 2014 functions, we uniformly set the maximum iterations \((T)\) to 5000 in the third experiment to meet the number of function evaluations. The special parameters of the FF, IWO, ABC, SSA, and ISSA algorithms are consistent with the literature [29], [37], [41],
TABLE 3. Description of the multimodal and fixed-dimension multimodal benchmark functions.

| Benchmark function                                                                 | $d$ | Range       | $f_{min}$     |
|-----------------------------------------------------------------------------------|-----|-------------|---------------|
| $F_1(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{|x_i|})$                                | 50  | [-500, 500] | -418.98 x Dim |
| $F_2(x) = \sum_{i=1}^{n} [x_i^2 - 10 \cos(2\pi x_i) + 10]$                       | 50  | [-5.12, 5.12] | 0             |
| $F_6(x) = -20 \exp\left(-0.2 \sum_{i=1}^{n} x_i^2\right) - \exp\left(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + e$ | 50  | [-32, 32]   | 0             |
| $F_{10}(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{n}}) + 1$ | 50  | [-600, 600] | 0             |
| $F_{11}(x) = \frac{\pi}{n} \sum_{i=1}^{n} (10 \sin(y_i) + \sum_{i=1}^{n} (y_i - 1)^2[1 + 10 \sin^2(\pi y_i) + (\sin(y_i + 1))] + (y_n - 1)^2 + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ | 50  | [-50, 50]   | 0             |
| $y_i = 1 + \frac{x_i^4}{4} + \frac{1}{a} x_i, a, k, m = \begin{cases} \begin{align*} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < a \end{align*} \end{cases}$ | 50  | [-50, 50]   | 0             |
| $F_{12}(x) = 0.1 [\sin^2(3\pi x_1) + \sum_{i=1}^{n} (x_i - 1)^2] + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$ | 50  | [-50, 50]   | 0             |
| $F_{13}(x) = \left( \sum_{i=1}^{n} \sum_{j=1}^{n} (x_i - x_j)^2 \right)^{-\frac{1}{2}}$ | 2   | [-65, 65]   | 1             |
| $F_{14}(x) = \sum_{i=1}^{n} (a_i - \frac{x_i (b_i y_i + b_i z_i)^2}{b_i^2 + b_i z_i + x_i})$ | 4   | [-5, 5]     | 0.00030       |

TABLE 4. Description of the CEC 2014 benchmark functions.

| Benchmark function                                                                 | $d$ | Range       | $f_{min}$     |
|-----------------------------------------------------------------------------------|-----|-------------|---------------|
| $F_{15}(x)$ (CEC1: Rotated High Conditioned Elliptic Function)                    | 50  | [-100, 100] | 100           |
| $F_{16}(x)$ (CEC2: Rotated Bent Cigar Function)                                   | 50  | [-100, 100] | 200           |
| $F_{17}(x)$ (CEC6: Shifted and Rotated Weierstrass Function)                     | 50  | [-100, 100] | 600           |
| $F_{18}(x)$ (CEC9: Shifted and Rotated Rastrigin’s Function)                     | 50  | [-100, 100] | 900           |
| $F_{19}(x)$ (CEC10: Shifted Schwefel’s Function)                                  | 50  | [-100, 100] | 1000          |
| $F_{20}(x)$ (CEC17: Hybrid Function 1 (N=3))                                     | 50  | [-100, 100] | 1700          |
| $F_{21}(x)$ (CEC24: Composition Function 2 (N=3))                                | 50  | [-100, 100] | 2400          |
| $F_{22}(x)$ (CEC25: Composition Function 3 (N=3))                                | 50  | [-100, 100] | 2500          |

[46], [47]. The parameters of the proposed RSSA are set to $N_s = 4$, $G_C = 1.9$, $P_{dp} = 0.1$, $N_0 = 0.5$, $\sigma_0 = 10$, and $\sigma_1 = 0.5$. For each benchmark function, all algorithms run 30 times independently from different randomly generated populations. To compare the proposed RSSA fairly with other algorithms, all the algorithms are implemented in MATLAB 2016a. All computations were run on a CPU: Intel Core i5-4200 M, 2.5 GHz, 4 G RAM, and Windows 7 (64-bit) operating system.

C. EXPERIMENTAL RESULTS

1) TEST RESULTS AND ANALYSIS ON UNIMODAL FUNCTIONS

Unimodal functions (TABLE 2) are often used to evaluate the exploitation ability of algorithms. The test results for unimodal functions are shown in TABLE 6, including the mean, standard deviation (Std), and best and worst fitness values of each algorithm after 30 runs. The mean of RSSA is the best in most unimodal functions, such as $F_1$, $F_2$, $F_3$, $F_4$ and $F_5$ functions. However, the performance of the RSSA is not excellent in all unimodal functions. For example, it has a worse performance on $F_6$ in terms of the mean. In addition, RSSA has the lowest standard deviation in most functions, which shows that RSSA has good stability. To reflect the performance of RSSA on the unimodal benchmark functions more clearly, a convergence rate analysis is also carried out. The convergence curves of several representative unimodal functions are shown in FIGURE 3, which reflects that the convergence rate of the RSSA is faster than that of other algorithms on most benchmark functions. Therefore, RSSA has excellent convergence performance for the unimodal functions.

2) TEST RESULTS AND ANALYSIS ON MULTIMODAL FUNCTIONS AND FIXED-DIMENSION MULTIMODAL FUNCTIONS

In this experimental test, the more complex multimodal functions and fixed-dimension multimodal functions (TABLE 3) are used to evaluate the exploration ability of RSSA. In addition, the corresponding statistical results of 30 independent
### TABLE 5. Parameter settings of the algorithms.

| Parameter          | Definition            | FF 1000/5000 | IWO 1000/5000 | ABC 1000/5000 | SSA 1000/5000 | ISSA 1000/5000 | RSSA 1000/5000 |
|--------------------|-----------------------|--------------|---------------|---------------|---------------|---------------|---------------|
| $T$                | Maximum iterations    | 1000         | 1000          | 1000          | 1000          | 1000          | 1000          |
| $N$                | Population size       | 50           | 50            | 50            | 50            | 50            | 50            |
| $\alpha$           | Step factor of disturbance | 0.25        | —             | —             | —             | —             | —             |
| $\beta$            | Attractiveness        | 0.20         | —             | —             | —             | —             | —             |
| $\gamma$           | Light absorption coefficient | 1            | —             | —             | —             | —             | —             |
| $N_{\text{initial}}$ | Initial population size | 10           | —             | —             | —             | —             | —             |
| $\sigma_{\text{initial}}$ | Initial standard deviation | 10           | —             | —             | —             | —             | —             |
| $\sigma_{\text{final}}$ | Final standard deviation | 0.02         | —             | —             | —             | —             | —             |
| $n$                | Nonlinear modulation index | 3            | —             | —             | —             | —             | —             |
| $n_{\text{Onlooker}}$ | Number of Onlookers  | —            | 50            | —             | —             | —             | —             |
| $acc$              | Expansion coefficient | —            | 1             | —             | —             | —             | —             |
| $N_{\text{fs}}$    | Number of acorn trees | —            | —             | 4             | 4             | 4             | —             |
| $G_{\text{C}}$     | Gliding constant      | —            | —             | 1.9           | 1.9           | 1.9           | —             |
| $P_{\text{dp}}$    | Predator presence probability | —     | —             | 0.1           | —             | 0.1           | —             |
| $P_{\text{dmax}}$  | Maximum presence probability | —     | —             | —             | 0.1           | —             | —             |
| $P_{\text{dmin}}$  | Minimum presence probability | —     | —             | —             | 0.001         | —             | —             |
| $N_{\text{o}}$     | Initial number of squirrels | —     | —             | 20            | —             | —             | —             |
| $\sigma_{0}$       | Maximum standard deviation | —     | —             | —             | —             | 10            | —             |
| $\sigma_{f}$       | Minimum standard deviation | —     | —             | —             | —             | 0.5           | —             |

### TABLE 6. Statistical results obtained by RSSA, ISSA, ABC, IWO, FF and SSA through 30 independent runs on functions $F_1$-$F_6$ with unimodal benchmark functions.

| Fun | Best | ISSA | ABC | IWO | FF | SSA |
|-----|------|------|-----|-----|----|-----|
| $F_1$ | Best | 1.66E-112 | 5.92E-30 | 1.39E-30 | 4.34E-76 | 1.07E-03 | 1.49E-90 |
| Worst | 2.36E-107 | 1.12E-28 | 4.96E-30 | 2.15E-70 | 1.78E-03 | 6.21E-86 |
| Mean | 4.71E-107 | 2.18E-28 | 8.52E-30 | 4.29E-70 | 2.50E-03 | 1.24E-85 |
| Std | 3.33E-107 | 1.50E-28 | 5.04E-30 | 3.04E-70 | 1.01E-03 | 8.78E-86 |
| $F_2$ | Best | 7.21E-105 | 8.73E-10 | 3.64E-12 | 3.65E-33 | 2.00E-03 | 1.13E-76 |
| Worst | 2.56E-103 | 3.91E-08 | 6.22E-10 | 4.33E-32 | 2.78E-03 | 6.19E-75 |
| Mean | 5.04E-103 | 7.73E-08 | 1.24E-11 | 8.29E-32 | 3.56E-03 | 1.23E-74 |
| Std | 3.52E-103 | 5.41E-08 | 8.79E-10 | 5.61E-32 | 1.11E-03 | 8.59E-75 |
| $F_3$ | Best | 7.92E-28 | 6.32E-09 | 1.16E-11 | 1.72E-07 | 4.07E-04 | 4.28E-09 |
| Worst | 9.89E-27 | 1.00E-08 | 4.96E-11 | 1.98E-07 | 3.78E-03 | 7.43E-08 |
| Mean | 1.90E-26 | 1.37E-08 | 8.75E-11 | 2.23E-07 | 7.15E-03 | 1.44E-07 |
| Std | 1.29E-26 | 5.22E-09 | 5.37E-11 | 3.64E-07 | 4.77E-03 | 9.91E-08 |
| $F_4$ | Best | 1.87E-01 | 6.18E+00 | 6.88E+00 | 3.95E+00 | 6.31E+00 | 4.42E+00 |
| Worst | 2.09E+00 | 6.66E+00 | 6.98E+00 | 4.39E+00 | 6.39E+00 | 4.56E+00 |
| Mean | 4.00E+00 | 7.15E+00 | 7.08E+00 | 4.84E+00 | 6.47E+00 | 4.69E+00 |
| Std | 2.69E+00 | 6.87E-01 | 1.44E-01 | 6.26E-01 | 1.15E-01 | 1.92E-01 |
| $F_5$ | Best | 0.00E+00 | 5.98E-30 | 7.54E-05 | 1.32E-06 | 1.37E-03 | 1.48E-31 |
| Worst | 0.00E+00 | 1.97E-28 | 9.66E-05 | 1.56E-06 | 1.51E-03 | 7.35E-31 |
| Mean | 0.00E+00 | 3.89E-28 | 1.18E-04 | 1.80E-06 | 1.64E-03 | 1.32E-30 |
| Std | 0.00E+00 | 2.71E-28 | 3.01E-05 | 3.39E-07 | 1.91E-04 | 8.30E-31 |
| $F_6$ | Best | 1.20E-04 | 7.84E-04 | 7.23E-04 | 2.04E-03 | 1.93E-02 | 1.73E-04 |
| Worst | 1.96E-04 | 2.25E-03 | 8.48E-04 | 4.10E-03 | 2.04E-02 | 2.10E-04 |
| Mean | 2.71E-04 | 3.71E-03 | 9.73E-04 | 6.17E-03 | 2.14E-02 | 2.46E-04 |
| Std | 1.07E-04 | 2.07E-03 | 1.77E-04 | 2.92E-03 | 1.51E-03 | 5.13E-05 |
TABLE 7. Statistical results obtained by RSSA, ISSA, ABC, IWO, FF and SSA through 30 independent runs on functions $F_7$-$F_{14}$ with multimodal and fixed-dimension multimodal benchmark functions.

| Fun | RSSA       | ISSA       | ABC        | IWO        | FF         | SSA        |
|-----|------------|------------|------------|------------|------------|------------|
| $F_7$ | Best: -2.67E+03 | -2.82E+03 | -3.28E+03 | -3.40E+03 | -3.08E+03 | -3.01E+03 |
|     | Worst: -2.29E+03 | -2.66E+03 | -3.12E+03 | -3.29E+03 | -3.01E+03 | -2.66E+03 |
|     | Mean: -1.92E+03 | -2.51E+03 | -2.97E+03 | -3.18E+03 | -2.94E+03 | -2.31E+03 |
|     | Std: 5.32E+02  | 2.21E+02  | 2.23E+02  | 1.54E+02  | 9.97E+01  | 4.89E+02  |
| $F_8$ | Best: 0.00E+00  | 4.97E+00  | 9.95E-01  | 4.97E+00  | 1.02E+01  | 0.00E+00  |
|     | Worst: 0.00E+00  | 4.97E+00  | 3.48E+00  | 4.99E+00  | 1.27E+01  | 0.00E+00  |
|     | Mean: 0.00E+00  | 4.97E+00  | 5.97E+00  | 5.00E+00  | 1.52E+01  | 0.00E+00  |
|     | Std: 0.00E+00  | 0.00E+00  | 3.52E+00  | 1.47E+02  | 3.54E+00  | 0.00E+00  |
| $F_9$ | Best: 4.44E-15 | 4.44E-15 | 7.99E-15  | 4.44E-15 | 6.08E-02  | 7.99E-15  |
|     | Worst: 4.44E-15 | 4.44E-15 | 7.99E-15  | 4.44E-15 | 6.47E-02  | 1.15E-14  |
|     | Mean: 4.44E-15 | 4.44E-15 | 7.99E-15  | 4.44E-15 | 6.86E-02  | 1.51E-14  |
|     | Std: 0.00E+00  | 0.00E+00  | 0.00E+00  | 0.00E+00  | 5.51E-03  | 5.02E-15  |
| $F_{10}$ | Best: 0.00E+00 | 3.20E-02 | 5.91E-02 | 3.45E-02 | 1.28E-01 | 0.00E+00 |
|      | Worst: 4.01E-02 | 6.52E-02 | 7.62E-02 | 5.17E-02 | 1.55E-01 | 8.76E-03 |
|      | Mean: 8.02E-02 | 9.84E-02 | 9.34E-02 | 6.89E-02 | 1.82E-01 | 1.75E-02 |
|      | Std: 5.67E-02 | 4.70E-02 | 2.43E-02 | 2.44E-02 | 3.84E-02 | 1.24E-02 |
| $F_{11}$ | Best: 4.71E-32 | 2.62E-07 | 7.23E-05 | 9.84E-32 | 1.94E-04 | 1.06E-31 |
|      | Worst: 4.76E-32 | 3.44E-07 | 1.03E-04 | 5.22E-30 | 2.08E-04 | 6.61E-31 |
|      | Mean: 4.81E-32 | 4.27E-07 | 1.33E-04 | 1.03E-29 | 2.23E-04 | 1.22E-30 |
|      | Std: 6.85E-34 | 1.17E-07 | 4.30E-05 | 7.24E-30 | 2.04E-05 | 7.86E-31 |
| $F_{12}$ | Best: 1.35E-32 | 3.39E-12 | 3.74E-04 | 1.76E-30 | 1.86E-06 | 5.05E-32 |
|      | Worst: 1.35E-32 | 3.79E-12 | 4.08E-04 | 2.71E-30 | 2.62E-06 | 8.01E-31 |
|      | Mean: 1.35E-32 | 4.18E-12 | 4.43E-04 | 3.65E-30 | 3.38E-06 | 1.55E-30 |
|      | Std: 0.00E+00  | 5.60E-13 | 4.85E-05 | 1.33E-30 | 1.07E-06 | 1.06E-30 |
| $F_{13}$ | Best: 9.98E-01 | 2.98E+00 | 9.98E-01 | 9.98E-01 | 1.99E+00 | 5.93E+00 |
|      | Worst: 1.50E+00 | 2.98E+00 | 9.98E-01 | 9.98E-01 | 6.38E+00 | 6.42E+00 |
|      | Mean: 1.99E+00 | 2.98E+00 | 9.98E-01 | 9.98E-01 | 1.08E+01 | 6.90E+00 |
|      | Std: 7.03E-01  | 5.19E-13 | 0.00E+00  | 0.00E+00  | 6.20E+00 | 6.89E-01 |
| $F_{14}$ | Best: 3.38E-04 | 3.07E-04 | 3.07E-04 | 3.08E-04 | 7.87E-04 | 6.08E-04 |
|      | Worst: 3.61E-04 | 1.03E-02 | 1.03E-02 | 7.65E-04 | 1.01E-03 | 7.45E-04 |
|      | Mean: 3.85E-04 | 2.04E-02 | 2.04E-02 | 1.22E-03 | 1.22E-03 | 8.82E-04 |
|      | Std: 3.31E-05 | 1.42E-02 | 1.42E-02 | 6.47E-04 | 3.09E-04 | 1.94E-04 |

**FIGURE 3.** Comparison of the convergence curves of RSSA and the other algorithms on functions $F_1$-$F_6$ with unimodal benchmark functions.
runs of RSSA and its competitors are recorded in TABLE 7. These results show that the proposed RSSA has a better mean and standard deviation on $F_7$, $F_9$, $F_{10}$, $F_{11}$, and $F_{12}$. It can also be observed that the performance of the RSSA is similar to that of the SSA and ABC algorithms on fixed-dimension multimodal functions ($F_{13}$ and $F_{14}$), which is caused by the simplicity of such optimization problems. Moreover, the convergence curves of these six algorithms on some representative benchmark functions are shown in FIGURE 4, from which it can be seen that the RSSA has a better convergence rate and sufficient accuracy for complex multimodal functions and fixed-dimensional multimodal functions. This experiment reflects that RSSA provides better exploration ability than its competitors. This is due to the enhancement of the global search ability by the proposed reproduction mechanism. The reproduction of squirrels in the iterative process increases the diversity of the population. Therefore, the RSSA has greater potential to avoid local optima.

3) TEST RESULTS AND ANALYSIS ON THE CEC 2014 COMPOSITE FUNCTIONS

The purpose of this test is to evaluate the comprehensive performance of the algorithms. The benchmark functions used are eight high-dimensional composite functions (TABLE 4) from the CEC 2014 functions, which are specially designed with complicated features. Therefore, this test is more challenging for each algorithm than the previous two experimental studies. As seen from the test results (TABLE 8), all the algorithms failed to reach the optimal solution. However, RSSA achieved more acceptable results on $F_{15}$, $F_{16}$, $F_{17}$, $F_{18}$, $F_{20}$ and $F_{22}$ than the other algorithms. The mean of ISSA is better than other algorithms on $F_{21}$. In terms of convergence performance, except for $F_{16}$, RSSA has the best convergence performance among all the algorithms. TABLE 9 records the CPU time of each algorithm after completing all iterations. It can be observed that the computational complexity of RSSA is slightly lower than that of SSA, but they are all higher than the other four algorithms. This is because RSSA introduces the reproduction mechanism, which increases the computational complexity. Furthermore, the results of the convergence analysis in FIGURE 5 show that compared with the other five algorithms, RSSA still has the best convergence performance. Through these comparisons, RSSA is still proven to be the best algorithm in solving such complex optimization problems. The reproduction mechanism and adaptive step size strategy of squirrels effectively balance the global search and local search of the RSSA. Therefore, the RSSA shows stability in such complex benchmark functions.
TABLE 9. The CPU time required by each algorithm.

| Fun | RSA | ISSA | ABC | IWO | FF | SSA |
|-----|-----|------|-----|-----|----|-----|
| $F_1$ | 0.236796 | 0.139221 | 0.948231 | 1.007699 | 0.425276 | 0.099441 |
| $F_2$ | 0.376261 | 0.291763 | 1.108325 | 1.13077 | 0.569478 | 0.249325 |
| $F_3$ | 0.210692 | 0.122847 | 0.923629 | 0.927948 | 0.369502 | 0.087914 |
| $F_4$ | 0.291313 | 0.196064 | 0.963699 | 0.986184 | 0.548459 | 0.155867 |
| $F_5$ | 0.233588 | 0.137538 | 0.917391 | 0.923634 | 0.417866 | 0.098528 |
| $F_6$ | 0.3154 | 0.227902 | 1.008702 | 1.203163 | 0.470392 | 0.190645 |
| $F_7$ | 0.25804 | 0.16681 | 0.961123 | 0.971999 | 0.475886 | 0.135536 |
| $F_8$ | 0.229378 | 0.142649 | 0.936736 | 0.947051 | 0.368486 | 0.106156 |
| $F_9$ | 0.254699 | 0.159162 | 0.954595 | 0.957392 | 0.376043 | 0.122372 |
| $F_{10}$ | 0.29681 | 0.202297 | 0.997072 | 1.00586 | 0.37139 | 0.165842 |
| $F_{11}$ | 0.695222 | 0.597155 | 1.481566 | 1.455561 | 0.862758 | 0.548426 |
| $F_{12}$ | 0.684583 | 0.589077 | 1.444744 | 1.429113 | 0.977103 | 0.540708 |
| $F_{13}$ | 1.585516 | 1.587871 | 2.450533 | 2.432886 | 3.039056 | 1.490853 |
| $F_{14}$ | 0.176342 | 0.144318 | 0.876596 | 0.897858 | 0.626944 | 0.086267 |
| $F_{15}$ | 2.26762 | 5.229577 | 12.37886 | 13.75833 | 3.781491 | 2.561535 |
| $F_{16}$ | 1.996387 | 4.458815 | 12.01093 | 12.10593 | 3.682417 | 2.478689 |
| $F_{17}$ | 11.86979 | 7.281775 | 21.13692 | 22.55869 | 13.89739 | 12.94599 |
| $F_{18}$ | 2.446261 | 4.323016 | 10.5707 | 10.65982 | 3.535994 | 2.43576 |
| $F_{19}$ | 2.264934 | 4.394817 | 11.19685 | 11.1103 | 3.716298 | 2.742971 |
| $F_{20}$ | 1.962579 | 3.970957 | 10.57225 | 10.64519 | 3.308914 | 2.364376 |
| $F_{21}$ | 5.705489 | 4.77598 | 14.62687 | 13.77707 | 5.866906 | 4.411275 |
| $F_{22}$ | 5.011003 | 4.168359 | 13.5934 | 13.16849 | 4.542189 | 3.653652 |
| Average | 1.789487 | 1.965356 | 5.548169 | 5.630181 | 2.374101 | 1.712369 |

4) STATISTICAL ANALYSIS

Nonparametric statistical technology should be performed to show the significant performance of the optimization algorithm. Further evidence on this can be found in an earlier monograph by Derrac et al. [48]. In this paper, a famous nonparametric statistical test method, Wilcoxon’s rank-sum test [49], is applied to the statistics of the experimental results to prove the significance performance of RSSA. TABLE 10 records the $P$ values obtained by the Wilcoxon’s test. From the table, almost all $P$ values are under the 95% confidence interval ($\alpha = 0.05$), which proves that RSSA has significant difference from other algorithms. In this table, the ‘+’ and
‘−’ signs indicate that the reference algorithm has better or worse performance than the compared algorithm, respectively. The last line represents the number of ‘+’ and ‘−’ signs, which also corresponds to the number of $P$ values under the 95% confidence interval ($\alpha = 0.05$). From the table, RSSA has more ‘+’ signs than the other algorithms. As a result, the RSSA has a better performance than the other five algorithms.

Finally, to quantitatively analyze the performance of the algorithm, the mean absolute error (MAE) values of all algorithms are calculated and sorted. The MAE is a valid statistical method that shows how far the result differs from the actual value. The MAE formula is as follows:

$$MAE = \frac{\sum_{i=1}^{N} |m_i - k_i|}{N}$$

where $m_i$ is the optimal value of an algorithm, $k_i$ represents the corresponding actual value of the benchmark function, and $N$ is the number of samples. In this work, $N$ is the number of benchmark functions and the MAE values and ranking of all algorithms are listed in TABLE 11. It can be clearly observed that RSSA provides the best performance compared to its competitors. In addition, FIGURE 6 reflects the comparison results of each algorithm run 660 times (30 runs for each benchmark function). The RSSA reached the optimal solution 421 times out of 660 runs.

TABLE 11. Average ranking of the six algorithms using MAE for 22 benchmark functions.

| Algorithm | MAE      | Rank |
|-----------|----------|------|
| RSSA      | 8.2227E+01 | 1    |
| SSA       | 6.3271E+02 | 2    |
| FF        | 3.8974E+03 | 3    |
| ISSA      | 5.5087E+03 | 4    |
| ABC       | 2.6867E+04 | 5    |
| IWO       | 1.0694E+05 | 6    |
V. CONCLUSION

In view of the low convergence accuracy and the ease with which SSA in high-dimensional problems fall into the local optimum, the RSSA with reproduction and adaptive step size strategy is proposed. First, the reproduction mechanism is applied to the search process to increase the diversity of the population, thereby enhancing the exploration ability of the algorithm. Second, the adaptive step size strategy is combined with the position updating formulas to balance the global and local search while avoiding the complex parameter adjustment of the algorithm. The performance of RSSA is evaluated on 22 benchmark functions including unimodal, multimodal, fixed-dimensional multimodal, and CEC 2014 benchmark functions. Finally, a comprehensive evaluation of RSSA is carried out by combining Wilcoxon’s statistics and MAE analysis. The experimental results and statistical analysis show that the accuracy, robustness and local optimal value avoidance of RSSA are superior to those of SSA and other advanced algorithms. In future work, the RSSA can be developed for solving more complex problems such as the multi-objective optimization, constrained optimization, and NP hard problems.

The main improvement of the proposed RSSA is in the global search ability. However, the improvement in the local search needs further research. Other initialization methods can be considered to spread the initial squirrel population throughout the search space to speed up convergence. This is also a new direction for future research.

REFERENCES

[1] H. Zhang, L. Cui, X. Zhang, and Y. Luo, “Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method,” IEEE Trans. Neural Netw., vol. 22, no. 12, pp. 2226–2236, Dec. 2011.
IEEE Access

X. Zhang et al.: Improved SSA With Reproductive Behavior

[2] Narendra and Fukunaga, “A branch and bound algorithm for feature subset selection,” IEEE Trans. Comput., vol. C-26, no. 9, pp. 917–922, Sep. 1977.

[3] A. Slowik and K. Halina, “Nature inspired methods and their industry applications—Swarm intelligence algorithms,” IEEE Trans. Ind. Informat., vol. 14, no. 3, pp. 1004–1015, Dec. 2017.

[4] A. Chakraborty and A. K. Kar, “Swarm intelligence: A review of algorithms,” Nature-Inspired Comput. Optim., vol. 10, pp. 475–494, Mar. 2017.

[5] H. Li, F. He, Y. Liang, and Q. Quan, “A dividing-bound many-objective evolutionary algorithm for large-scale feature selection,” Soft Comput., vol. 24, no. 9, pp. 6851–6870, Sep. 2019.

[6] N. Hou, F. He, Y. Zhou, and Y. Chen, “An efficient GPU-based parallel tabu search algorithm for hardware/software co-design,” Frontiers Comput. Sci., vol. 4, no. 5, pp. 1–18, Mar. 2020.

[7] M. Le Quinio, P. Mandel, and L. Monier, “Optimization of drinking water and sewer hydraulic management: Coupling of a genetic algorithm and two network hydraulic tools,” Procedia Eng., vol. 89, pp. 710–718, Nov. 2014.

[8] Z. Sui, Y. Fan, T. Yuliang, X. Lei, and Z. Yifong, “Optimized algorithm of sensor node deployment for intelligent agricultural monitoring,” Comput. Electron. Agric., vol. 127, pp. 76–86, Sep. 2016.

[9] M. Reihanian, S. R. Asadullahpour, S. Hajarpour, and K. Gheisari, “Application of neural network and genetic algorithm to powder metallurgy of pure iron,” Mater. Design, vol. 32, no. 6, pp. 3183–3188, Jun. 2011.

[10] A. Castiglione, R. de Prisco, A. De Santis, U. Fiore, and F. Palmieri, “A botnet-based command and control approach relying on swarm intelligence,” J. Netw. Comput. Appl., vol. 38, pp. 22–33, Feb. 2014.

[11] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in Proc. Int. Conf. Neural Netw. (ICNN), Perth, WA, Australia, Nov./Dec. 1995, pp. 1942–1948.

[12] M. Dorigo, M. Birattari, and T. Stützle, “Ant colony optimization,” IEEE Comput. Intell. Mag., vol. 1, no. 4, pp. 28–39, Nov. 2006.

[13] D. Karaboga and B. Basturk, “A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm,” J. Global Optim., vol. 39, no. 4, pp. 459–471, Apr. 2007.

[14] X. Yan, F. He, N. Hou, and H. Ai, “An efficient particle swarm optimization for large-scale Hardware/Software co-design system,” Int. J. Cooperat. Inf. Syst., vol. 27, no. 1, Mar. 2018, Art. no. 1741001.

[15] J. Tian, Y. Tan, J. Zeng, C. Sun, and Y. Jin, “Multiojective infill criterion driven Gaussian process-assisted particle swarm optimization of high-dimensional expensive problems,” IEEE Trans. Evol. Comput., vol. 23, no. 3, pp. 459–472, Jun. 2019.

[16] T. Blackwell and J. Kennedy, “Impact of communication topology in particle swarm optimization,” IEEE Trans. Evol. Comput., vol. 23, no. 4, pp. 689–702, Aug. 2019.

[17] J. Derrac, S. García, D. Molina, and F. Herrera, “A practical tutorial on non-parametric tests for analyzing the evolutionary algorithms’ behaviour,” Swarm Evol. Comput., vol. 39, no. 3, pp. 689–702, Aug. 2019.

[18] M. Reihanian, S. R. Asadullahpour, S. Hajarpour, and K. Gheisari, “Application of neural network and genetic algorithm to powder metallurgy of pure iron,” Mater. Design, vol. 32, no. 6, pp. 3183–3188, Jun. 2011.

[19] A. Castiglione, R. de Prisco, A. De Santis, U. Fiore, and F. Palmieri, “A botnet-based command and control approach relying on swarm intelligence,” J. Netw. Comput. Appl., vol. 38, pp. 22–33, Feb. 2014.

[20] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in Proc. Int. Conf. Neural Netw. (ICNN), Perth, WA, Australia, Nov./Dec. 1995, pp. 1942–1948.

[21] M. Dorigo, M. Birattari, and T. Stützle, “Ant colony optimization,” IEEE Comput. Intell. Mag., vol. 1, no. 4, pp. 28–39, Nov. 2006.

[22] D. Karaboga and B. Basturk, “A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm,” J. Global Optim., vol. 39, no. 4, pp. 459–471, Apr. 2007.

[23] X. Yan, F. He, N. Hou, and H. Ai, “An efficient particle swarm optimization for large-scale Hardware/Software co-design system,” Int. J. Cooperat. Inf. Syst., vol. 27, no. 1, Mar. 2018, Art. no. 1741001.

[24] J. Tian, Y. Tan, J. Zeng, C. Sun, and Y. Jin, “Multiojective infill criterion driven Gaussian process-assisted particle swarm optimization of high-dimensional expensive problems,” IEEE Trans. Evol. Comput., vol. 23, no. 3, pp. 459–472, Jun. 2019.

[25] T. Blackwell and J. Kennedy, “Impact of communication topology in particle swarm optimization,” IEEE Trans. Evol. Comput., vol. 23, no. 4, pp. 689–702, Aug. 2019.

[26] J. Derrac, S. García, D. Molina, and F. Herrera, “A practical tutorial on non-parametric tests for analyzing the evolutionary algorithms’ behaviour,” Swarm Evol. Comput., vol. 39, no. 3, pp. 689–702, Aug. 2019.

[27] M. Reihanian, S. R. Asadullahpour, S. Hajarpour, and K. Gheisari, “Application of neural network and genetic algorithm to powder metallurgy of pure iron,” Mater. Design, vol. 32, no. 6, pp. 3183–3188, Jun. 2011.

[28] A. Castiglione, R. de Prisco, A. De Santis, U. Fiore, and F. Palmieri, “A botnet-based command and control approach relying on swarm intelligence,” J. Netw. Comput. Appl., vol. 38, pp. 22–33, Feb. 2014.

[29] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in Proc. Int. Conf. Neural Netw. (ICNN), Perth, WA, Australia, Nov./Dec. 1995, pp. 1942–1948.